Com SIS
Computer Science and Information Systems
Volume 16, Number 3, October 2019

Contents

ISSN: 1820-0214 (Print) 2406-1018 (Online)

Computer Science and Information Systems

Vol 16, No 3, October 2019

Com SIS

Computer Science and Information Systems

Published by ComSIS Consortium

Special Issue on Recent Advances in Information Processing and Security

Volume 16, Number 3
October 2019

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Editorial Office:
University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
Phone: +381 21 458 888; **Fax:** +381 21 6350 458
www.comsis.org; Email: comsis@uns.ac.rs

Volume 16, Number 3, 2019
Novi Sad

Computer Science and Information Systems

Special Issue on Recent Advances in Information Processing and Security

ISSN: 1820-0214 (Print) 2406-1018 (Online)

The ComSIS journal is sponsored by:

Ministry of Education, Science and Technological Development of the Republic of Serbia
http://www.mpn.gov.rs/

ComSIS

Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2018 two-year impact factor 0.620,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Journal Bibliometric Report of the Center for Evaluation in Education and Science (CEON/CEES) in cooperation with the National Library of Serbia, for the Serbian Ministry of Education and Science,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (MS Word or LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from http://www.comsis.org), along with a cover letter containing the corresponding author's details should be sent to official journal e-mail.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 20 (using the appropriate Word or LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

Computer Science and Information Systems

Volume 16, Number 3, Special Issue, October 2019

CONTENTS

GUEST EDITORIAL

RECENT ADVANCES IN INFORMATION SECURITY AND INFORMATION TECHNOLOGY

Recent years have witnessed rapid developments of the information communication technology and the next-generation Internet. We have seen the potential and value for new technologies from a variety of digital networks to various interconnected digital devices, which make it possible to realize the ubiquitous network and society. But, in such environment, copyright infringement behaviors, such as illicit copying, malicious distribution, unauthorized usage, and free sharing of copyright-protected digital contents, will also become a much more common phenomenon. Researchers, content industry engineers, and administrators attempt to resort to the state-of-the-art technologies and ideas to protect valuable digital contents and services assets against attacks and IP piracy in the emerging ubiquitous network.

This special issue aims to bring together related research works in the realm of ubiquitous network and multimedia contents security and further investigates and discusses trusted solutions and secure techniques for ubiquitous network as well as some open issues faced by digital contents, owners/rights, holders/multimedia, and services providers, who dedicate themselves to protect their intellectual property rights and business benefits.

In this section, thirteen papers have been selected for publication. All selected papers followed the same standard (peer-reviewed by at least three independent reviewers) as applied to regular submissions. They have been selected based on their quality and their relation to the scope of the special section.

In the paper entitled “SARSA Based Access Control with Approximation by TileCoding” by Fei Zhu et al., an efficient and feasible TileCoding reinforcement learning algorithm for access control is proposed, which provides a policy for sensor nodes in different complex situations.

The paper entitled “Research on Improved Privacy Publishing Algorithm Based on Set Cover” by Haoze Lv et al. proposes a marginal table cover algorithm based on frequent items by considering the effectiveness of query cover combination, and then obtain a regular marginal table cover set with smaller size but higher data availability.

A novel survivability entropy evaluation method is proposed in the paper entitled “A Novel SMP-based Survivability Evaluation Metric and Approach in Wireless Sensor Network” by Hongsong Chen et al. The method is aimed to precisely calculate the survivability ability under DoS flood attack in wireless sensor network.

In the paper entitled “A Framework for Fog-assisted Healthcare Monitoring” by Jianqiang Hu et al., a framework for fog-assisted healthcare monitoring is proposed. This framework is composite of the body-sensing layer, fog layer and cloud layer. The body sensing layer measures physiological signals and fog-assisted gateway collects these information.

The paper entitled “Design of Intrusion Detection System Based on Improved ABC_elite and BP Neural Networks” by Letian Duan et al. proposes an intrusion detection method based on improved artificial bee colony algorithm with elite-guided search equations (IABC_ elite) and Backprogation (BP) neural networks.

The paper entitled “Hierarchical Authority Based Weighted Attribute Encryption Scheme” Qiuting Tian et al. discusses a weighted attributed encryption scheme based on a hierarchical authority.

The study presented in the paper entitled “Algorithm of Web Page Similarity Comparison Based on Visual Block” by Xingchen Li et al. introduces a new similarity matching algorithm based on visual blocks. The proposed algorithm analyzes the RenderLayer tree of the web page, extracts visual information by making the matching rules, determines the feature sets of the visual blocks corresponding to a web page by processing the visual nodes, and obtains similarity matching in accordance with the optimal free matching principle.

The scheme proposed in the paper entitled “Privacy-preserving Multi-authority Attribute-based Encryption with Dynamic Policy Updating in PHR” by Xixi Yan et al. aims to construct the MA-ABE scheme for PHR scenario which has the requirement for privacy preserving and policy updating.

In order to study on learner’s state and its change, the Hidden Markov Model was applied in the paper entitled “Classification and analysis of MOOC Learner's State: The study of Hidden Markov Model” by Haijian Chen et al. to analyze data about learners, which includes MOOCs learner’s basic information, learning behavior data, curriculum scores and data of participation in learning activities and so on.

Sensor networks will always suffer from load imbalance, which causes bottlenecks to the communication links. In order to address this problem, a multipath routing algorithm based on data-fusion-mechanism (MR-DFM) is proposed in the paper entitled “MR-DFM: A Multi-path Routing Algorithm Based on Data Fusion Mechanism in Sensor Networks” by Zeyu Sun et al.

The paper entitled “Research of MDCOP Mining Based on Time Aggregated Graph for Large Spatio-temproal Data Sets” by Zhanquan Wang et al. proposes the design and implementation of the prototype of PANBED, which builds a small-scale personal testbed for users utilizing devices in their own personal area networks (PANs). The experimental results show that PANBED allows users to set up different network scenes to test applications easily using a home router, PCs, mobile phones and other devices.

The paper entitled “A Novel Self-adaptive Grid-partitioning Noise Optimization Algorithm Based on Differential Privacy” by Zhaobin Liu et al. This article is based on application scenarios for the partition based data distribution algorithm. For the partition-based data distribution method, the authors first uniformly partitions the original spatial data, add a Laplace noise with uniform scale parameters, then select the set of grids to be optimized in a standard way based on the maximum ratio

between the value of overall error reduction and privacy budget increase value, and then operate noise optimized algorithm for the grid.

The paper entitled “Enhanced Artificial Bee Colony with Novel Search Strategy and Dynamic Parameter” by Zhenxin Du et al. describes a novel triangle search strategy to enhance the capability of escaping from local minimum without loss of the exploitation ability of ABC-BB.

We would also like to thank Prof. Mirjana Ivanović, the editor-in chief of ComSIS, for her support during the preparation of this special section in the journal.

Finally, we gratefully acknowledge all the hard work and enthusiasm of authors and reviewers, without whom the special section would not have been possible.

Editors of the Special Issue

Fei Yu, *Peoples' Friendship University of Russia, Moscow, Russia*
Chin-Chen Chang, *Feng Chia University, Taichung, Taiwan*
Muhammad Khurram Khan, *King Saud University, Riyadh, Saudi Arabia*
Jun Wang, *University of Central Florida, Florida, USA*
Miodrag J. Mihaljević, *Mathematical Institute of the Serbian Academy of Sciences and Arts, Serbia*

Computer Science and Information Systems 16(3):689–704 https://doi.org/10.2298/CSIS180830022Z

SARSA Based Access Control with Approximation by TileCoding

Fei Zhu[1,2], Pai Peng[1], Quan Liu[1], Yuchen Fu[3], and Shan Zhong[4]

[1] School of Computer Science and Technology, Soochow University
Shizi Street No.1 158 Box, Suzhou, Jiangsu 215006, China
zhufei@suda.edu.cn, 20185227062@stu.suda.edu.cn,quanliu@suda.edu.cn
[2] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University
Shizi Street No.1 158 Box, Suzhou, Jiangsu 215006, China
[3] School of Computer Science and Engineering, Changshu Institute of Technology
Changshu, Jiangsu, 215500, China
yuchenfu@cslg.edu.cn
[4] School of Computer Science, University of Wisconsin
Eau Claire, Wisconsin, USA
zhongs@uwec.edu

Abstract. Traditional sensor nodes ignore the packet loss rate during information transmission and the access control security problem caused by server utilization when uploading data. To solve the problem, we propose a SARSA based access control method with approximation by TileCoding (SACT), which takes the sensor packet loss rate and the server error rate into account. The network state is estimated by the packet loss rate and variable bit error rate to get a server access control strategy to improve security performance. The eventual strategy complies with the minimum information loss and the maximum server utilization. Results of experiments show that the algorithm is capable of achieving good results in the total amount of information received by the server system. The SACT improves the server utilization rate and the overall security performance of the network.

Keywords: access control, TileCoding, information security, SARSA.

1. Introduction

Access control is a way of granting or restricting the subject's access to the object explicitly featuring a very important information security technology. Access control technology is applied in many areas of information systems such as intrusion detection system [5], information encryption [12], identity authentication [7], security audit [3], security and risk analysis [14]. It combines technology with theory to guarantee safe and reliable transmission. It is also applied to access information system [32], which largely retains the integrality of information effectively and reduces the information leak and omission. Access control system takes some defensive measures to manage the access of system resources so as to ensure the full use of system resources.

Authorization policy [4] is the core issue of access control. With the development of distributed computing and information technology, heterogeneous networks are interconnected and are increasingly communicating. In order to ensure secure networks, we

should consider the tense, environment and other factors when establishing the access control model. However, information error and loss will inevitably occur in the entire information transmission process from the overall and long-term perspective. Consequently, the critical issue is to maximize the safety and effectiveness of transmission of information. Quantitative security of information transmission needs to be focused.

In the wireless sensor network, the sensor nodes will have unique information loss [27] when they upload data to the server, which is called lose bag and will cause the server to receive the unevenly distributed data flow. The number of servers available is rarely taken into account in the traditional access control strategy, which often leads to the inability to receive as much amount of information as possible. On the other hand, the error rate of the port of the wireless sensor network is high enough to the point of resulting the instability of communication between the server and the sensor equipment. At the same time a dynamic change process ought to be reckoned.

Considering different packet loss rates [22] and bit error rates [30] concerning transmission security problems, we propose an access control strategy based on TileCoding reinforcement learning algorithm, referred as SACT, that considers the sensor with packet loss and a limited number of servers, and takes different sensor nodes that upload data at different times as the starting point. Considering the number of available servers and the error rate, it coordinates the rejection and reception of the sensor at certain timestamps, which makes the whole system receive the most information. Access control strategy and the algorithms of SARSA are introduced in the next section.

2. Related work

2.1. Access control

The core of access control is the authorization policy [15], which controls the subject's access to the object. Access control models can be divided into various types including traditional access control models, role-based access control models (RBAC), task-based and role-based access control models (T-RBAC), and task-based and workflow-based access control models (TBAC).

Traditional Access Control Model In general, the traditional access control model can be divided into 2 categories: mandatory access control (MAC) [26] and discretionary access control (DAC) [10]. Mandatory access control takes a coercive measure. It uses up reading/down writing to ensure data integrity and up writing/down reading to ensure data confidentiality. Although the implementation work of the MAC is heavy and the management is inconvenient and not flexible enough, MAC can realize the one-way flow of information to prevent Trojan horses effectively.

The MAC is to search for all users who have access to a particular resource. It can effectively implement authorization management. It is difficult for MAC to deal with situations where an organization modifies its members and where functions are changed.

Role-Based Access Control In role-based access control(RBAC) [29], the concept of roles is introduced, that is, the responsibilities and functions of users within the organization. Each role is under a corresponding function. In this model, permissions are assigned

to roles, and users are assigned to previously assigned roles to obtain permission for roles. Different users are assigned different roles according to their corresponding functions. The system can be simplified by pre-defining the role-permission relationships.

In a workflow environment, the user performing the operation is changing, so are the user's permissions when data flows in a workflow. This is linked to the context of data processing, which traditional access control technologies DAC and MAC cannot implement. The RBAC reference model also includes functional specifications, which are subdivided into management functions [24], support functions [1], review functions and so on.

Task-Based Authorization Control The Task-Based Authorization model(TBAC) [35] is an active security model, which uses dynamic authorization to realize the security model and implement the task-oriented security mechanism . In the model, users combine tasks with access rights. The status of the task determines the permissions that the system grants to the user, based on the task currently executed by the topic. However, TBAC does not make a distinction between tasks and roles and does not realize active access control thereby exposing its own shortcomings. Generally speaking, TBAC is used in combination with RBAC.

The access control of the object is not static, but varies with the context in which the task is performed. The active and dynamic nature of TBAC makes it widely used in workflow, distributed processing, information processing of multi-point access control and decision making of the the transaction management system.

Task-role-based Access Control Task-role-based Access Control(T-RBAC) [19] is an Access Control Model Based on the enterprise environment. Unlike RBAC, the T-RBAC model treats tasks and roles as equals. In this model, the task owner has specific task requirements and authority constraints. The corresponding task has the corresponding permission, which makes the permission change with the task execution. This can truly realize the demand allocation and dynamic allocation of permission. When the task completes, the role's permission is revoked; when the task is not started, the role has no permission; when the task is executing, the role is assigned permission.

Roles in T-REAC are associated with permission, and tasks serve as the bridge for roles and permission to exchange information. It not only achieves easy and convenient operational maintenance of the roles and task management, but also achieves a more secure system.

2.2. Introduce to SARSA

Reinforcement learning(RL) [34] is a classic and effective method in machine learning. As a branch of the machine learning discipline, deep learning constructs a neural network to simulate the human brain to achieve observational learning. It mimics the working principle of human brain to understand external input data, such as images, texts and sounds. So it is a great tool for implementing artificial intelligence. Reinforcement learning is a class of algorithms that are extremely suitable for achieving artificial intelligence. This discipline is based on the study of animal learning and adaptive control theory. In artificial intelligence problems, Agents are generally used to represent an object with behavioral

capabilities, such as robots, unmanned vehicles, animals, and so on. Agents generally have some specific properties: sociality, autonomy, responsiveness, initiative, and so on. The main issue is the interaction between the agent and the external environment. The probability of a certain behavioral choice increases when the agent chooses it and is rewarded by the external environment and decreases when the agent is punished. However, reinforcement learning does not have a clear decidable signal like the common machine learning. It can only evaluate if the action is encouraged according to the symbol and size of the enhanced signal, so the whole learning process is time consuming.

Reinforcement learning includes many famous algorithms such as Q-learning [36], SARSA [2], function approximation [8] and so on. Reinforcement learning has been widely used in practice in many areas [17,11]. It also has achieved numerous contributions in biomedicine [23] and game playing [18]. In reinforcement learning, the agent interacts with Markov Decision Process(MDP) [33] so as to model the reinforcement learning problem. Reinforcement learning can be described as Fig. 1.

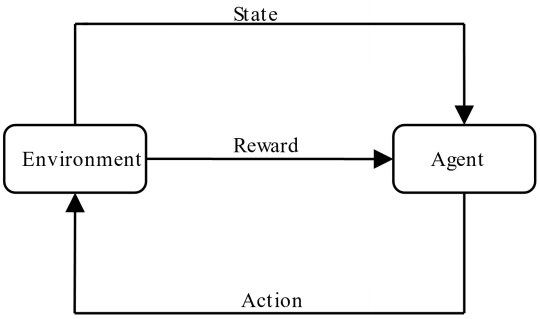


Fig. 1. The process of reinforcement learning

The Markov Decision Process (MDP) is used for modeling. An MDP model is often represented as a tuple $< S, A, P, R >$.

S denotes the state set of agent,$s_t \in S$ represents the state of agent at time t.

A denotes the action set for a state,$a_t \in A$ represents the action that can be taken at time t.

P denotes the transition probability, which is usually expressed as $P(s_{t+1}|a_t, s_t)$, indicating the probability that the agent takes action a' and reach the next state s_{t+1} at the time of t and state s_t.

R denotes the reward function, which is expressed as $r_{t+1} = R(s_t, s_{t+1}, a_t)$, indicating the immediate reward r_t given by the environment after the action a_t was taken from state s_t to state s_{t+1}.

The policy of reinforcement learning π: $S \rightarrow A$ is a mapping from states to actions. In general, reinforcement learning uses state-action function values to evaluate and to improve strategies. The state-action value function is the expected reward when imple-

menting the strategy, which can be generally expressed as:

$$Q(s_t, a_t) = E[R_t|s_t, a_t, \pi] \tag{1}$$

In traditional reinforcement learning, Bellman equation[28] plays an important role, which can be expressed as:

$$Q(s, a) = E[r + \gamma max(Q(s', a')|s, a] \tag{2}$$

where r is the immediate reward,γ is the discount factor, s is the state and a is the action.

SARSA algorithm estimates the action-value function (Q function)[31] rather than the state value function. In other words, we estimate the action value function of all available actions a on any state s under the policy. SARSA algorithm makes full use of markov property, that is, the future state is only related to the current state. SARSA updates the status value at each step using formula as follows:

$$Q(s, a) = Q(s, a) + \alpha(R + \gamma Q(s', a') - Q(s, a)) \tag{3}$$

The complete process of SARSA algorithm is shown in Algorithm 1.

Algorithm 1 SARSA

Input: $Q(s, a)$ arbitrarily
Output: $Q(s, a)$ updated
1: Initialize S
2: **repeat**
3: Choose A from S using policy from Q (e.g., Boltzmann's method)
4: $Q(s, a) \leftarrow Q(s, a) + \alpha[Q(s', a') - Q(s, a)]$
5: $s \leftarrow s', a \leftarrow a'$
6: **until** terminal
7: **return** $Q(s, a)$

The state space and action space are usually large in practical applications. The iterative algorithm used to seek the optimal policy is more computationally intensive and less feasible. Therefore, it is necessary to generalize the large-scale state space. The representing method of function is used to approximate the Q value,as follows:

$$\delta = r_{t+1} + \gamma \max_u Q(s_{t+1}, a_{t+1}; \theta_{t+1}) - Q(s_t, a_t; \theta_t) \tag{4}$$

$$\theta_{t+1} = \theta_t + \alpha \delta e_t \tag{5}$$

$$e_t = \gamma \lambda e_{t-1} + \nabla Q(s_t, a_t; \theta_t) \tag{6}$$

where δ is TD error, e is eligibility trace,and θ is function parameter. Linear functions or nonlinear functions are used to approximate the function. In this paper, function approximation is performed with TileCoding.

In recent years, there are a lot of researches on applying reinforcement learning to the field of access control, routing planning and others. In wireless sensor network structure, the majority of them goes toward two directions. The first is how to reduce the waste of resources, that is, to maximize the use of resources. The second is to improve the efficiency of node and to minimize the cost. Fathi et al.presents q-learning for multiple access control in wireless sensor networks to save energy of sensor node [9]. Chu Yi et al. applies Q-learning to frame based ALOHA as an intelligent slot selection strategy capable of migrating from random access to perfect scheduling [6]. Yun Lin et al. proposes a hybrid spectrum access algorithm that is based on a reinforcement learning model for the power allocation problem of both the control channel and the transmission channel [20]. Many scholars also use the planning method [21] to obtain the optimal solution.

However, these algorithms still suffer from some general issues such as low mobility, security, collision avoidance [25]. Built on the reinforcement learning method, this paper proposes an access control method to maximize the received information data and select the policy with the minimized loss of information.

3. SARSA based Access Control

In the access control of wireless sensor network [16], data transmitted to the server by different sensors may be affected by different levels of error rates, resulting in the loss of important information. The energy consumption of the sensor is also a problem that needs to be considered. In order to address the deficiency of the current sensor network, an optimal access control strategy is got to obtain the most complete information by the server with the minimum energy consumption of the sensor. The algorithm can guarantee a satisfying secure transmission.

In this paper, SARSA based access control with approximation by TileCoding is proposed. The algorithm builds a model based on the idea of reinforcement learning, considering error when uploading, and the package loss of different sensors. We will show the optimal access control policy to achieve the overall minimum information loss and the highest security rate of transmission. The SACT algorithm uses TileCoding to improve the function approximation parameters in the enhanced learning, exploits differential semi-gradient SARSA to constantly update the Q value [13], and utilizes the average reward to update the reward function. After several iterations, the information loss in the transmission of multiple sensors is reduced.

3.1. Model and Algorithm Description

In wireless sensor network, each sensor node may suffer from a series of problems such as information loss, energy consumption and data error when transmitting information to the server. Taking these data security factors into consideration, an algorithm for the access control problem of the sensor node can be obtained. In the algorithm referred in this paper, each server can at the same time receive only one sensor to upload data. We assume the server control system as an agent, which can take steps to accept or reject a sensor. Our goal is to seek the action strategy of the control system for the new sensor node for different states consisting the sensor node and servers.

The optimization of wireless sensor network can be described as a single agent's reinforcement learning problem of seeking out optimal strategies for separate states. In other words, this reinforcement learning problem can be seen as the agent's value function being continuously updated. It is presumed that a wireless sensor network with multiple sensor nodes is sent to a fixed number of servers in a random queue sequence. The system has several features as follows: every server has a certain error rate due to energy loss, which relates to its performance decline over time. Only one sensor node uploads data in a time slice. The server system can decide whether to accept the data uploaded by the sensor node according to the current status. And different sensor nodes have different package loss rates.

Reinforcement learning algorithms require modeling and analysis of state, action, and reward functions. In this wireless sensor network system, n_i servers are initially used to receive data, and there are enough sensor nodes in the queue to transmit data. The sensor nodes are divided into n_j types, and their package loss rate is $e_1, e_2 \cdots e_j$, respectively. In a time slice, only one sensor node can upload data to a server. A busy server has a certain possibility p to complete data transmission and continue to receive the node data of the next sensor. The error rate of the server increases by $w\%$ for every T period. The agent here refers to the server control system, and the state refers to the current number of idle servers, the error rate priority of the server receiving data and the type of sensor node at the current queue head. Action a refers to the option to reject or receive the uploaded data of the current queue head sensor according to the current state. According to the reinforcement learning method, the Q value of state-action team function as well as the state function are evaluated. Some exploration method, such as ϵ-greedy and boltzmann exploration, is used to select the action according to the state function.

3.2. SACT Algorithm

The SACT algorithm is proposed in this paper to obtain the most complete possible information against the increasing server error rate. When considering the reward function, it is needed to integrate the bit error rate of the current server and the package loss rate of sensor nodes. Therefore, the reward function is defined as:

$$r(s_t, a_t) = \alpha(1 - e_t)Inf - \beta w_t \tag{7}$$

where α,β denotes accordingly the weights of the packet loss rate and the bit error rate and Inf denotes the maximum information the sensor node can upload among them.

The agent selects action a_{t+1} with boltzmann probability under the state s_{t+1} with weight w, and updates the average reward and weight w. The average reward and weight are updated as follows:

$$\delta \leftarrow r - \overline{R} + Q(s_{t+1}, a_{t+1}, w) - Q(s_t, a_t, w) \tag{8}$$

$$\overline{R} \leftarrow \overline{R} + m\delta \tag{9}$$

$$w \leftarrow w + n\delta\nabla\overline{Q}(s_t, a_t, w) \tag{10}$$

where r means the immediate reward,m and n are updated steps.

Exploiting the concept of average reward, we define a new type of reward as the sum of the difference values between the rewards and the average rewards, which is called differential reward. The algorithm of differential semi-gradient SARSA is adopted to continuously sample and to learn. The value w and the Q value will converge finally. We apply TileCoding, a kind of CourseCoding to code in the learning stage. A unit of computation is a tiling, and every sensory field characteristic value is associated with a tile. The sensory field of the eigenvalue is divided into the computing unit of the input space. The value function of TileCoding is composed of the weight of each tile, and the expression is:

$$V(s) = \sum_{i=1}^{n} b_i(s) w_i \tag{11}$$

where n denotes the total number of tiles, $b_i(s)$ denotes the ith tile of the corresponding state s and w_i is the corresponding weight.

It is generally not necessary to sum up several tiles for a given state. We update the estimated value of state s according to the following updates expression under the MDP model:

$$\nabla V(s) = \max_a [R(s,a) + \gamma V(T(s,a))] - V(s) \tag{12}$$

Weight update formula is as follows:

$$w_i \leftarrow w_i + \frac{\alpha}{m} b_i(s) \nabla V(s) \tag{13}$$

where m is the number of tilings.

The algorithm of TileCoding is as follows:

Algorithm 2 Updating the parameter by TileCoding

Input: the number of tiles n, the number of tiling m, the initial weights
Output: the weights W updated

for $i = 1$ to n **do**
  Initialize the tiling by using n/m tiles
end for
for $j = 1$ to n/m **do**
  Initialize the tile with weights
end for
repeat
  $s \leftarrow$Random State from S
  $\nabla V(s) = \max_a [R(s,a) + \gamma V(T(s,a))] - V(s)$
  for $k = 1$ to m **do**
    $W \leftarrow W + \alpha/m \nabla V(s)$
  end for
until terminal
return W

Active-tile returns the number of tiles that are activated. By utilizing multiple layers of tilings to generalize in multiple directions, the tradeoff between accuracy and speed is avoided. The tiling of the lattice is used to divide the state space into several tilings in this

paper, which is obtained through the random uniqueness of the discrete grid. Coordinates are obtained in each tiling to obtain the corresponding feature vectors for the number of servers and the status of sensor nodes in the team head. The approximation function of its state space is as follows:

$$V_t(s) = \theta_t^T \phi_s = \sum_{i=1}^{n} \theta_t(i)\phi_s(i) \tag{14}$$

Updating Parameter is expressed as follows:

$$\theta_{t+1} = \theta_t + \alpha[V^\pi(s_t) - V_t(s)]\nabla_{\theta_t} V_t(s_t) \tag{15}$$

TileCoding allocates several large tiles at the beginning of learning. These tiles will be then divided into several sub-tiles all at once. Tiles are split into two parts in this article to simplify calculation. The learning speed and performance indexes were optimized on the basis of compound tiling. Set a global counter u to record the update times. When u reaches the threshold, select a tile to segment. Since the quota policy is based on control or prediction, sub-tiles need to be recorded or screened. So,k status values contain $2k$ tiles. When each tile is generated, the weight of the sub-tile is initialized to 0. As the status is continuously updated, the activated sub tile k is updated accordingly. The following diagram illustrates this process. Fig. 2 shows the sequential segmentation of tiles.

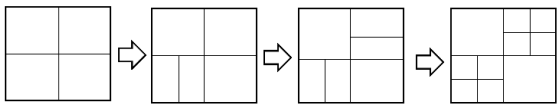


Fig. 2. Sequential segmentation of tiles

In the learning process, we utilized hash coding, which greatly reduced the amount of storage space. We also use pseudo-random algorithm to reduce the number of tiles. Hash correlation is used to form a limited tiling with non-adjacent tiles in the state space randomly. Hash coding can effectively solve the dimension disaster problem. Hashing mapping is showed in Fig. 3.

Main procedures of this algorithm are as follows: we initialize all the servers to be idle and obtain the Q value through the TileCoding function for all the sensor nodes at certain state. The weight parameters are initialized to 0 in tile. All the initial rewards are initialized to 0. The package loss rate and current server bit error rate are used to obtain information from each new sensor nodes. The initial state is chosen by the random policy. The action is chosen by Boltzmann distribution probability according to the free servers and the type of sensor at queue head. The Q value is updated by the SARSA Algorithm. Finally, through multiple sensor nodes in the queue and the bit error rate of the server which increases over time, the evaluation value gradually converges with iteration. The specific algorithm is combined with TileCoding as follows.

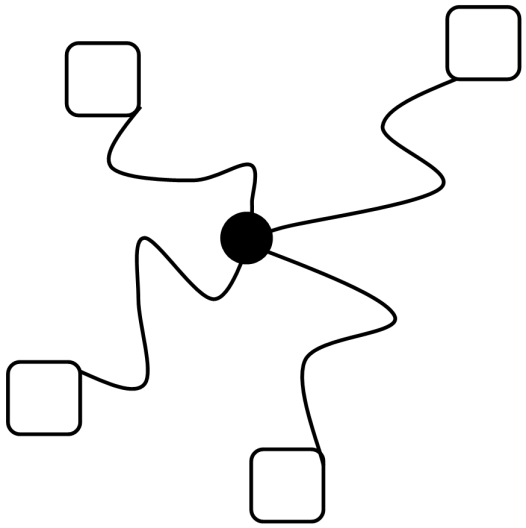


Fig. 3. The hash map of one tile

where m is the number of tilings, e_t is the bit error rate, λ/T is the packet loss rate based on the growth of fixed time period and w is an array of the weights.

The traditional access control problem only considers the condition of the visitor. It ignores the data security processing of some servers. The algorithm in this paper takes into account the package loss rate of data uploaded by sensor nodes and the bit error rate of the server compared to the traditional algorithm, which can be better applied to the actual situation.

4. Experiments and Analysis

We simulate the situation of sensor nodes and servers. The simulation environment is that there are 5 different types of sensor nodes in the queue, and their loss rates of uploaded data are 0.01%, 0.02%, 0.04%, 0.08% and 0.16%, respectively. There are 12 servers that receive data. And bit error rate increases by 3% every fifty thousand time steps. For each busy server, there is an 8% probability that the busy state will change to idle state getting ready to receive data. To facilitate comparison, we set their immediate rewards to 16, 8, 4, 2, and 1 for the sensor nodes with different packet loss rates that are randomly generated in the queue head. Immediate rewards will be multiplied by the corresponding percentages as time goes on. The purpose of this task is to explore a policy to decide whether to receive data in each time step according to the nature of sensor nodes, number of servers and the current state, so as to obtain a secure data transmission policy.

In the learning process, we used the average reward as the reward function, set accept action as 1 and reject action as 0, set step size for learning state-action value as 0.01,

Algorithm 3 SARSA based access control with approximation by TileCoding (SACT)

Input: the style of sensor node in the queue head Y,step sizes $\alpha, \beta > 0$, number of busy servers N

Output: the server system reject or accept the new sensor node

for $i = 0$ to n **do**
  Initialize the weights w of the i tile and $2k$ sub tiles e.g., $w = 0$
end for
Initialize average reward $\overline{R}$ arbitrarily(e.g.,$\overline{R} = 0$)
Initialize state $S = [N, Y]$, actions A
repeat
  Take actions A, observe state S
  Choose A' as the function of $Q(S', w)$(e.g., Boltzmann's method)
  $r(S, A) = (1 - \mu)(1 - e_t)Inf - \mu\lambda/T$
  $\delta \leftarrow r - \overline{R} + Q(s_{t+1}, a_{t+1}, w) - Q(s_t, a_t, w)$
  $\overline{R} \leftarrow \overline{R} + \beta\delta$
  $w \leftarrow w + \frac{\alpha}{m}\delta\nabla Q(s_t, a_t, w)$
  $S \leftarrow S'$
  $A \leftarrow A'$
until terminal
return (S, A)

step size for learning average reward as 0.01, and updated the parameters according to TileCoding. Therefore, in the experiment in this paper, the aim is to get a policy after experiencing 1 million time steps under different numbers of idle servers and the sensor nodes at the queue head required to upload data.

Fig. 4 shows the differential value of optimal actions for sensor nodes that produce different package loss rates in 1 million time steps. It can be observed from the figure that for the sensor nodes with different packet loss rates, the value obtained through TileCoding reinforcement learning is inversely proportional to the packet loss rate, which is consistent with intuition. It can also be seen from the figure that when the number of idle servers reaches 9, there is a huge amount of transitions under different sensor nodes. When the number of busy server is 3, the data is at its most complete and all states are traversed. In addition, the performance appears not satisfactory when the value is lower than 0. When the number of idle servers is in range only from 0 to 4, the data obtained in the simulation experiment cannot cover all cases.

Fig. 5 shows the result of the strategy after 1 million time steps where the dark color means receiving new sensor node data, and light color means refusing to accept new sensor node. On the whole, as the number of idle servers decreases, higher packet loss rate corresponds to greater chance of rejection.

More specifically, when the number of free server is 0, all sensor nodes are rejected which makes the algorithm reliable. The server system can take the corresponding policy according to the figure. In order to maximize security of data transmission, the server system can only choose the packet loss rate of type 4 and type 3 when the free server number is 1. When the number of free servers is 2 or 3, type 0 and type 1 sensor nodes cannot upload data. When the number of free servers is 11 or 12, the system still refuses type 0 or type 1 because the system prefers to keep as much security as possible and get as much reward as possible. By comparing Fig. 4 with Fig. 5, we can find that the two

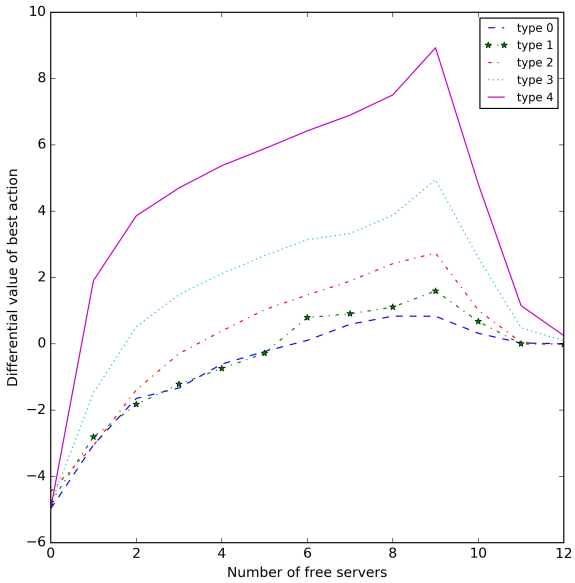


Fig. 4. The differential value of optimal actions of the five different sensor nodes that produce different package loss rates in the 1 million time steps for the different number of free servers. Type 0 to 4 of the sensor nodes denotes the rewards that are getting larger from 1 to 16.

graphs can be related. When the number of idle servers is from 0 to 3, or from 11 to 12, we find that the value is very low so most servers of these numbers are rejected accordingly. In addition, when the values are highest, the server system is in a position to receive all types of sensor nodes to upload data.

5. Conclusion

The defect of traditional access control in wireless sensor networks can lead to a large number of information omissions without adopting a long-term effective strategy. This will make the information transmission insecure. In this paper, an efficient and feasible access control policy based on TileCoding reinforcement learning algorithm is proposed, which provides us with a policy for sensor nodes in different complex situations. After taking full factors into account such as the package loss rate of the sensor node and the bit error rate of the server, the SACT algorithm adopts the method of function approximation by TileCoding and takes corresponding actions through the probability of Boltzmann distribution. We adopt the idea of average reward and simulate the credible strategy. Simulation results show that the SACT algorithm can provide a relatively safe strategy for the access control of sensor nodes.

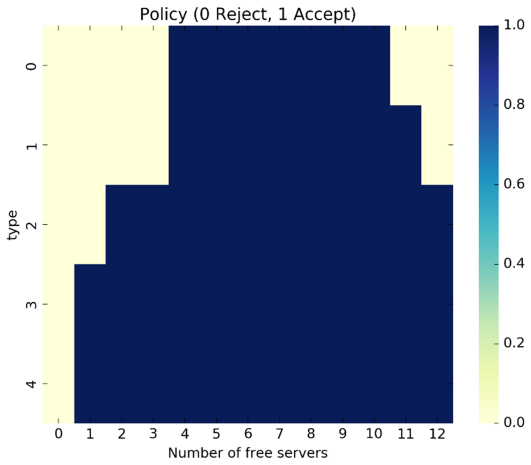


Fig. 5. The result of the strategy for the server system after 1 million time steps. Dark color means receiving new sensor node data, and light color means refusing to accept new sensor node. The actions can be chosen by the state of the number of free servers and type of sensor nodes in the figure

Acknowledgment This work was supported by National Natural Science Foundation of China (61303108, 61373094, 61702055); The Natural Science Foundation of Jiangsu Higher Education Institutions of China (17KJA520004); Suzhou Key Industries Technological Innovation-Prospective Applied Research Project (SYG201804); Program of the Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1524).

References

1. Matthias Althoff and Goran Frehse. Combining zonotopes and support functions for efficient reachability analysis of linear systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7439–7446. IEEE, 2016.
2. J Amaré, S Cebrián, C Cuesta, E García, M Martínez, MA Oliván, Y Ortigoza, A Ortiz de Solórzano, C Pobes, J Puimedón, et al. Status of the anais dark matter project at the canfranc underground laboratory. In *Journal of Physics: Conference Series*, volume 718, page 042052. IOP Publishing, 2016.
3. Abeba Mitiku Asfaw, Mesele Damte Argaw, and Lemessa Bayissa. The impact of training and development on employee performance and effectiveness: A case study of district five administration office, bole sub-city, addis ababa, ethiopia. *Journal of Human Resource and Sustainability Studies*, 03(04):188–202, 2015.
4. Prosunjit Biswas, Ravi Sandhu, and Ram Krishnan. Label-based access control: An abac model with enumerated authorization policy. In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 1–12. ACM, 2016.
5. Loic Bontemps, Van Loi Cao, James Mcdermott, and Nhienan Lekhac. Collective anomaly detection based on long short-term memory recurrent neural networks. *arXiv: Learning*, pages 141–152, 2016.

6. Yi Chu, Selahattin Kosunalp, Paul D Mitchell, David Grace, and Tim Clarke. Application of reinforcement learning to medium access control for wireless sensor networks. *Engineering Applications of Artificial Intelligence*, 46:23–32, 2015.
7. Wang Ding, Haibo Cheng, Debiao He, and Wang Ping. On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices. *IEEE Systems Journal*, PP(99):1–10, 2016.
8. Thanh-Toan Do and Ngai-Man Cheung. Embedding based on function approximation for large scale image search. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):626–638, 2018.
9. Mohammad Fathi. Reinforcement learning for multiple access control in wireless sensor networks: Review, model, and open issues. *Wireless Personal Communications*, 72(1):535–547, 2013.
10. David Ferraiolo, Ramaswamy Chandramouli, Rick Kuhn, and Vincent Hu. Extensible access control markup language (xacml) and next generation access control (ngac). In *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*, pages 13–24. ACM, 2016.
11. Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.
12. Shay Gueron and Vlad Krasnov. Fast prime field elliptic-curve cryptography with 256-bit primes. *Journal of Cryptographic Engineering*, 5(2):141–151, 2015.
13. Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*, 2015.
14. Andreas Jacobsson, Martin Boldt, and Bengt Carlsson. A risk analysis of a smart home automation system. *Future Generation Computer Systems*, 56(C):719–733, 2016.
15. Shellie L Keast, Hyunjee Kim, Richard A Deyo, Luke Middleton, K John McConnell, Kun Zhang, Sharia M Ahmed, Nancy Nesser, and Daniel M Hartung. Effects of a prior authorization policy for extended-release/long-acting opioids on utilization and outcomes in a state medicaid program. *Addiction*, 113(9):1651–1660, 2018.
16. Imran Khan, Fatna Belqasmi, Roch Glitho, Noel Crespi, Monique Morrow, and Paul Polakos. Wireless sensor network virtualization: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):553–576, 2016.
17. Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In *Advances in neural information processing systems*, pages 3675–3683, 2016.
18. Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 4190–4203, 2017.
19. Hongjiao Li, Shan Wang, Xiuxia Tian, Weimin Wei, and Chaochao Sun. A survey of extended role-based access control in cloud computing. In *Proceedings of the 4th International Conference on Computer Engineering and Networks*, pages 821–831. Springer, 2015.
20. Yun Lin, Chao Wang, Jiaxing Wang, and Zheng Dou. A novel dynamic spectrum access framework based on reinforcement learning for cognitive radio sensor networks. *Sensors*, 16(10):1675, 2016.
21. LShuai, Lei Liu, Hong Jiang, and Wei Wei. Classical optimal planning method based on compacted encodings. *Journal of Jilin University*, 40(6):1644–1649, 2010.
22. Renquan Lu, Yong Xu, and Ridong Zhang. A new design of model predictive tracking control for networked control system under random packet loss and uncertainties. *IEEE Transactions on Industrial Electronics*, 63(11):6999–7007, 2016.
23. Polina Mamoshina, Armando Vieira, Evgeny Putin, and Alex Zhavoronkov. Applications of deep learning in biomedicine. *Molecular pharmaceutics*, 13(5):1445–1454, 2016.

24. Rashid Mijumbi, Joan Serrat, Juan-Luis Gorricho, Steven Latré, Marinos Charalambides, and Diego Lopez. Management and orchestration challenges in network functions virtualization. *IEEE Communications Magazine*, 54(1):98–105, 2016.
25. Amir Mukhtar, Likun Xia, and Tong Boon Tang. Vehicle detection techniques for collision avoidance systems: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2318–2338, 2015.
26. Simone Mutti, Enrico Bacis, and Stefano Paraboschi. Sesqlite: Security enhanced sqlite: Mandatory access control for android databases. In *Proceedings of the 31st Annual Computer Security Applications Conference*, pages 411–420. ACM, 2015.
27. M Naghiloo, JJ Alonso, A Romito, E Lutz, and KW Murch. Information gain and loss for a quantum maxwell's demon. *Physical review letters*, 121(3):030604, 2018.
28. Huyên Pham and Xiaoli Wei. Bellman equation and viscosity solutions for mean-field stochastic control problem. *ESAIM: Control, Optimisation and Calculus of Variations*, 24(1):437–461, 2018.
29. Qasim Mahmood Rajpoot, Christian Damsgaard Jensen, and Ram Krishnan. Attributes enhanced role-based access control model. In *International Conference on Trust and Privacy in Digital Business*, pages 3–17. Springer, 2015.
30. Hector Reyes, Sriram Subramaniam, and Naima Kaabouch. A bayesian network model of the bit error rate for cognitive radio networks. In *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, pages 1–4. IEEE, 2015.
31. Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320, 2015.
32. S. Shahrabadi, M. Moreno, J. I. Rodrigues, J. M. Rodrigues, and J. M. Buf. Computer vision and gis for the navigation of blind persons in buildings. *Universal Access in the Information Society*, 14(1):67–80, 2015.
33. Haiying Shen and Liuhua Chen. Distributed autonomous virtual resource management in datacenters using finite-markov decision process. *IEEE/ACM Transactions on Networking*, 25(6):3836–3849, 2017.
34. Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
35. JD Ultra and Susan Pancho-Festin. A simple model of separation of duty for access control models. *Computers & Security*, 68:69–80, 2017.
36. Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Fei Zhu is a member of China Computer Federation. He is a PhD and an associate professor. His main research interests include machine learning, reinforcement learning, and bioinformatics. Email:zhufei@suda.edu.cn.

Pai Peng is a postgraduate student in the Soochow University. His main research interest is reinforcement learning. He programmed the algorithms and implemented the experiments. Email:20185227062@suda.edu.cn.

Quan Liu is a member of China Computer Federation. He is a PhD, post-doctor, professor and PhD supervisor. His main research interests include reinforcement learning, intelligence information processing and automated reasoning. Email:quanliu@suda.edu.cn.

Yuchen Fu (corresponding author) is a member of China Computer Federation. He is a PhD and professor. His research interest covers reinforcement learning, intelligence in-

formation processing, and deep Web. He is the corresponding author of this paper. Email: yuchenfu@cslg.edu.cn.

Shan Zhong got her PhD in computer science and technology. Her main interests include machine learning, artificial intelligence and application of reinforcement learning. Email:zhongs@uwec.edu.

Received: August 30, 2018; Accepted: July 6, 2019.

Computer Science and Information Systems 16(3):705–731 https://doi.org/10.2298/CSIS180915023L

Research on Improved Privacy Publishing Algorithm Based on Set Cover

Haoze Lv[1], Zhaobin Liu[1], Zhonglian Hu[1], Lihai Nie[2], Weijiang Liu[1], and Xinfeng Ye[3]

[1] School of Information Science and Technology, Dalian Maritime University
China
Correspondence: zhbliu@dlmu.edu.cn
[2] Division of Intelligence and Computing, Tianjin University
China
nlh3392@tju.edu.cn
[3] Department of Computer Science, University of Auckland
New Zealand
x.ye@auckland.ac.nz

Abstract. With the invention of big data era, data releasing is becoming a hot topic in database community. Meanwhile, data privacy also raises the attention of users. As far as the privacy protection models that have been proposed, the differential privacy model is widely utilized because of its many advantages over other models. However, for the private releasing of multi-dimensional data sets, the existing algorithms are publishing data usually with low availability. The reason is that the noise in the released data is rapidly grown as the increasing of the dimensions. In view of this issue, we propose algorithms based on regular and irregular marginal tables of frequent item sets to protect privacy and promote availability. The main idea is to reduce the dimension of the data set, and to achieve differential privacy protection with Laplace noise. First, we propose a marginal table cover algorithm based on frequent items by considering the effectiveness of query cover combination, and then obtain a regular marginal table cover set with smaller size but higher data availability. Then, a differential privacy model with irregular marginal table is proposed in the application scenario with low data availability and high cover rate. Next, we obtain the approximate optimal marginal table cover algorithm by our analysis to get the query cover set which satisfies the multi-level query policy constraint. Thus, the balance between privacy protection and data availability is achieved. Finally, extensive experiments have been done on synthetic and real databases, demonstrating that the proposed method preforms better than state-of-the-art methods in most cases.

Keywords: Differential Privacy, Set Cover, Frequent Itemsets, Marginal Table.

1. Introduction

With the rising attention of big data, the amount of published information is rapidly increasing. Online office, cloud storage, data synchronization and other services have brought convenience for us, but also generate a huge user data aggregation. These data contain a wealth of valuable information, therefore, data releasing and sharing has become an important topic both in scientific research and information industry [1]. However, directly releasing original data can lead to information disclosure of user. So, how to publish

the data while protecting the user's personal privacy has become an important issue. In the scenario with privacy protection requirements, the differential privacy model is widely used because it doesn't need attack hypothesis and background knowledge of attackers, but it can quantify and analyze the privacy risk [2] [3].

However, when publishing the multi-dimensional data sets, current privacy models often have low performances in both privacy protection and data availability. The reasons are explained as follows. First, the noise added into each dimension in the released data will undoubtedly increase when the dimension of the data becomes larger. Second, the query results are usually not very valid due to large cumulative noise of data. In view of this issue, we propose a privacy-preserving data releasing algorithm based on regular marginal tables to reduce the noise and improve the data availability.

To figure out the issue of regular marginal table, we propose a publishing algorithm to reduce the dimension of the cover data set in the same dimension marginal table, thus achieving differential privacy protection with Laplace noise [4]. We first introduces how to choose the dimension k of marginal tables in a cover σ. Then we choose a smallest set of k-marginal tables satisfying the cover on the condition of a proper k. Different from the classic algorithms in this step, we use frequent item to measure the importance of the k-marginal tables. The situation that k-marginal tables with more frequent attributes are usually with higher data availability.

Meanwhile, we propose a differential privacy model with irregular marginal table partitioning and find the marginal table query cover set which satisfies the multi-level query policy constraint. This model is proposed to be used in application scenarios with low data privacy protection requirements and high cover requirements. Thus, the balance between privacy protection and data availability is achieved. The main contributions of this paper is summarized as follows.

1. We do many research on bounds of k-marginal table cover and propose to use frequent attributes to increase the usability of the released data.
2. We present a k-marginal tables publishing algorithm based on frequent items and regular marginal benefits, which is with lower time complexity and higher data usability.
3. We introduce improved marginal table differential privacy publishing algorithms based on irregular marginal table in this paper in detail.
4. We conduct extensive experiments in both public databases, demonstrating that the presented algorithm always performs better than the state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 shows the related work of this paper. Then we will present regular marginal table differential privacy publishing algorithm under frequent item set algorithms in Section 3. Another algorithm with some theorems and proofs we give in section 4 is used in the scenario of irregular marginal table. We make performance analysis by experiments in Section 5. Finally, Section 6 will draw some conclusions.

2. Related Work

2.1. Differential Privacy

In literature, variable kinds of privacy-preserving algorithms [5], [6], [7] have been proposed to prevent the sensitive information from being disclosed. Among these algorithms,

perhaps the most well-known algorithm is K-anonymous algorithm proposed by Sweeney in 2002 [8]. K-anonymous algorithm has the characteristic that a record can be protected with at least $k-1$ other records with respect to the quasi-identifier. However, K-anonymous algorithm is only focused on identity protection. To overcome the drawback, Machanavajjhala et al. [9] proposed L-diversity algorithm inspired by K-anonymous algorithm to provide sufficient protection against attribute disclosure. However, the premise of data generalization is to know the background knowledge of the attacker, which is always impossible. Thus the data privacy protection technology that encrypts the sensitive information is proposed.

Recently, the major efforts in this field are paid to reduce noise while protecting privacy at the same time, including histogram [10] and contingency tables [11]. But with the increasing popularization of the big data, the amount of user information exploded. Querying the publishing data is a problem with high sensitivity because of large quantity noise. Therefore the Flat Method [12], Matrix [13], Data Cube [14] do not work with big data.

To solve the problem, Dwork etc. proposed a differential privacy protection model [15] using the Laplace mechanism. It is a common mechanism for achieving differential privacy by adding Laplace noise to numerical data. Therefore, the method is mainly used in the privacy protection release process of statistical data. Laplace differential privacy model solves the problem that the attacker can easily find the user's privacy information through the normalized query strategy. Meanwhile, it can also achieve a high performance in big data.

The basic idea of the differential privacy model is to randomly perturb the published data, so that the attacker cannot obtain the private information of individual from the published data regardless of any background knowledge and any data mining and analysis information. The advantages of this model are that there is no need to make special assumptions about the attacker's background knowledge and the specific attack method. Thus, we use the privacy budget to analyze the risk of data disclosure quantitatively.

2.2. Set Cover Problem

Set Cover Problem (SCP) is a classical problem in combinatorial mathematics, computer science and computer complexity theory. This paper uses the idea of set cover to form mathematically model and improve the middle ware partitioning problem in the differential privacy model. Therefore, this part will introduce the basic concepts of set cover problems. Set cover problem is divided into two categories based on the definition of the problem: the determinant problem [16] and the optimization problem of set cover [17]. The goal of set cover optimization problem is to find cover sets that meet the optimization conditions.

SCP has been proved to be a NPC or NPH problem [18] as early as 1976. As a result, the study of the approximate algorithm of the optimal cover set is one of the important point of this issue which is an approximation optimal set selected from the candidate through the algorithm. The goal of the approximation algorithm is to reduce the time complexity of the algorithm by obtaining the approximate solution [19]. Thus, the approximation algorithm cares about two aspects of the problem: time complexity and approximate degree of solution.

As early 1970, Edmonds has proved that the greedy algorithm for linear functions must be optimal in the matroid structure [20]. However, the problem in practical applications is not linear but a submodule function. The submodule function is a formal description of the "marginal utility decrement" in the set theory. In recent years, in the research of the modulo algorithm of the submodule function, Sagnol G proposed the modulo algorithm of the submodule function in polynomial time [21], and applied it to the maximum cover problem. He proved that unless $P = NP$ is satisfied, the greedy algorithm is the best maximization algorithm in polynomial time for the condition that satisfies the submodule function. In order to solve the problem of finding the minimum submodule cover set, Pengjun Wan et al. proposed MSC/SC (Minimum Submodular Cover with Submodular Cost)[22]. The pseudo code of the algorithm is shown in Algorithm 1.

Algorithm 1: MSC

Input : a collection of E
Output: cover collection X

$X \leftarrow \emptyset$
while $\exists e \in E$ such that $\Delta_e f(X) > 0$ **do**
    select $x \in E$ with maximum $\Delta_e f(X)/c(x)$
    $X \leftarrow X \cup \{x\}$

Nevertheless, there may be a risk that the usability of the data is reduced by mixing too much noise based on contingency table. To overcome this problem, Qardaji et al. proposed a differential privacy released algorithm based on marginal table [23]. Noise is effectively reduced based on marginal table differential privacy model. However, relevance of the real data set of attributes is not taken into account. The marginal table contains a part of invalid query combination, which reduces the data availability. In view of this problem, we propose marginal table differential privacy publishing algorithms based on frequent item sets.

There are many candidate methods for mining frequent items [24], [25], [26], [27], [28] and we choose well-known Apriori [29] to analyze the data set in practical application, considering about the marginal table support and marginal benefit. Then the majority of valid query combinations is covered by generated marginal table and the data availability of query middle ware is improved further.

2.3. Differential Privacy Publishing Algorithm based on Marginal Table

Differential privacy is a model to protect information for data publishing. Therefore, in the scenario of multi-dimensional or high-dimensional data publishing, the research and application of differential privacy publishing algorithm is necessary. Wahbeh Qardaji et al proposed a differential privacy publishing algorithm called PriView [30] in 2014. The algorithm overcomes the problem of large added-noise and low data availability when publishing high-dimensional data sets. Its key idea is to cover the partial query range of high-dimensional data by using multiple marginal table cover sets composed of tables of the same dimension. This section will give a brief introduction to its improvement methods and related technologies.

Middle Ware of Publishing Marginal Table The marginal table is a new type of query middle ware derived from the contingency table proposed by Wahbeh Qardaji. It can be obtained by splitting, extracting, and recombining the attributes included in the contingency table. It's noise has been significantly reduced compared to the contingency table.

In the differential privacy publishing algorithm of the Laplace mechanism, the noise added in the analysis of the query middle ware can be reduced by publishing a low-dimensional marginal table. However, the shortcomings of the marginal table also exist. The marginal table implements dimension reduction by truncating the attributes of the contingency table, which certainly leads to a reduction in the scope of the query. However, in high-dimensional data sets, the noise error made by the contingency table publishing method will be too hard to estimated. What's more, as the data dimension becomes higher, the contingency table may become more and more sparse, which means the result of multi-attribute queries will be 0 or lack of realistic query meaning. Therefore, we sacrifice some of the meaningless high-dimensional queries and use the marginal table instead of the contingency table publishing in the differential privacy publishing algorithm. It has certain practical application value in reducing noise and improving data availability.

Laplace Noise Error Analysis under Marginal Table The marginal differential privacy publishing algorithm publish a cover set of the original d-dimensional data set consisting of n k-dimensional marginal tables. At present, the noise mixing method used for the marginal table cover set is mainly the direct method. Since the middle ware is a combination of multiple tables, in order to facilitate the noise error analysis, Wahbeh Qardaji proposed the ESE (Expected Squared Error) noise estimation method. This method estimates the noise error of the single marginal table by adding the each noise-added result m_{ij} from each item to the variance of the real result n_{ij}. The calculation formula is as shown in Eq. 1.

$$ESE = \sum_{j=1}^{2^k} (m_{ij} - n_{ij})^2 \tag{1}$$

The direct method is proposed by Dwork et al [31]. This method directly adds the noise of $Laplace(n/\varepsilon)$ to each item in the same-dimensional marginal table. It proves that the marginal table cover set proposed by the method satisfies ε-differential privacy, where n is the number of marginal tables. In general $n = C_d^k$, which means all the marginal table cover sets combined by the k-dimensional table are released, in this method $ESE_d = 2^k \times (C_d^k/\varepsilon)$

There is a large amount of redundant information between the marginal tables of the same dimension that is often unnecessary to be published in practice. At the same time, the larger the number of the marginal table, the huger Laplace noise will be mixed. Therefore, in the PriView algorithm, Wahbeh Qardaji et al proposed the marginal table dimension and quantity selection method based on the overlay design to complete the construction of the marginal table cover set. This method further reduce the number of marginal tables n under the cover requirement by reducing the cover relationship between the same dimension tables, which reduces the overall noise. The expected variance of the method is $ESE_d = 2^k \times (n/\varepsilon)$, where $n < C_d^k$.

Issue of Former Method The PriView algorithm proposed by Wahbeh Qardaji et al completed the optimization of the number of marginal tables by the overlay design, but the method still has the following problems.

(1) PriView only gives the approximate range of the optimal marginal table dimension selection. It does not explicitly give the selection method of the marginal table dimension under different dimension data sets as well as the query cover rate the set can provide under this dimension.

(2) The method uses cover design to find the k-dimensional marginal table cover set, and complete the full cover of the query combination of the $k-1$ dimension table. However, the method is under the condition of the assumption that the relationships among the attributes in the data set are completely independent of each other. Actually, there is always a certain correlation between the attributes in the real data set. These correlations determine the validity of the query combination covered by the marginal table. Obviously, PriView's marginal table lookup method has a certain degree of blindness, and will contain redundant invalid query combinations, which will increase the number of published marginal tables and reduce data availability.

To figure out these two problems, we propose a regular marginal table differential privacy publishing algorithm under frequent item sets in section 3. By analyzing the relationship between the dimension change of the marginal table and the cover rate, the algorithm gives the selection method of the table dimension under different query cover requirements. Meanwhile, this method estimates the support of the marginal table combined with the frequent item mining algorithm and establishes the weighted marginal table set cover model with the support degree. By improving the CMC algorithm, a marginal table cover algorithm based on support degree and query is proposed. Finally, it achieves targeting cover of valid query combinations, and reduces the number of marginal tables that further improves data availability.

(3) PriView only uses the same dimension table to publish the marginal table cover set, which can improve the data availability by sacrificing a part of the query scope. However, the cover method has certain limitations in the application scenario where the data privacy protection is not high and the query range cannot be reduced.

In order to overcome this shortcoming, we propose a differential privacy publishing algorithm based on irregular marginal table partitioning, which uses different dimension marginal tables to form a cover set in section IV. We control the amount of noise mixed in the publishing marginal table as much as possible to make a balance between privacy protection and data availability by constraining the multi-level query rules.

3. RD-Privacy Algorithm

In this section, we present our RD-Privacy (Regular marginal table Differential Privacy releasing) algorithm. Firstly, the basic implementation flow of our algorithm is briefly described. Secondly, by analyzing the upper and lower bounds of the cover, we propose the relationship formula between the marginal table dimension and the cover. Finally, the filtering condition of the candidate table of marginal table is analyzed, and the marginal table covering algorithm is presented based on it.

3.1. Algorithm Overview

The main idea of the proposed RD-Privacy method is to find n k-marginal tables, which can effectively cover the query set. To improve the availability of the published data, we want to reduce n so as to add less noise, under the condition that the noise should obey the Laplacian distribution to satisfy ε-differential privacy.

Assuming that there is a data set D with dimension d, then a marginal table with k $(k < d)$ dimension is a view of data set D (only k attributes are shown) and naturally there are $\binom{d}{k}$ kinds of k marginal tables. Suppose U_D is all the query collection of D, we can obviously obtain that $|U_D| = \sum_{i=1}^{d} \binom{d}{i} = 2^d - 1$. We define m_k is the query collection set of all k-marginal tables and $m_k^i (1 \leq i \leq \binom{d}{k})$ is the query collection set of the i-th k-marginal tables. Then $m_k = \{s : s \subset U_D, |s| \leq k\}$, and $\left|m_k^i\right| = 2^k - 1$.

It is easy to see that finding n k-marginal table equals to discovering subsets of U_D. As shown in Eq. 2, the query cover σ is mainly related to the parameter k. We analyze it in Section 3.2, and propose a dimension selection method based on Eq. 2.

$$\sigma = \frac{\left|\bigcup_{i=1}^{n} m_k^i\right|}{|U_D|} < \frac{n(2^k - 1)}{2^d - 1}. \tag{2}$$

To satisfy differential privacy, we need to add $Laplace(n/\varepsilon)$ noise into the marginal tables. The expected squared error of n marginal tables ESE_n is shown in Eq. 3.

$$ESE_n = 2^{k+1} \times (n/\varepsilon)^2. \tag{3}$$

From Eq. 3, we can see that when n is too large, the publishing middle ware which satisfy ε-differential privacy needs to be mixed into the excessive noise, resulting into lower data availability.

To have a clear understanding of notations, we give a summary of notation of in Table 1.

Table 1. Summary of notations

Name	Notion
D	Database
d	dimension of Dataset D
k	dimension of marginal table
U_D	query collection of Dataset D
m_k^i	query collection of the i-th k marginal tables
m_k	query collection of all the k marginal tables
ESE_n	the Expected Squared Error of n marginal tables
M_{opt}	the optimal solution
M_{app}	the approximate solution of M_{opt}
$m(.)$	function of calculating the marginal benefit
$Mben(.)$	represents the rate of $m(.)$

Since ESE_n can be decreased by reducing n, we propose to obtain smaller size k-marginal tables based on frequent attributes. Firstly, we get the frequent item sets and corresponding support of attributes in data set D, and then use their support to weight marginal tables. By this way, we can filter out the ineffective marginal tables with low and little influence on querying. A more detailed description can be found in Section 3.3.

In the following, we give the definition of our weighted k-marginal tables covering problem.

Definition 1. For a data set D, the query collection U_D and the cover σ, our weighted k-marginal tables covering problem is to find n k-marginal tables, so as to satisfy $\left|\bigcup_{i=1}^{n} m_k^i\right| / |U_D| \geq \sigma$, the support of these k-marginal tables $\sum_{i=1}^{n} Sup(m_k^i)$ as large as possible, the expected squared error ESE_n as small as possible.

This kind of problem has been proved to be NP-hard, so it is hard to find the optimal solution M_{opt} in polynomial time. To address this issue, we propose RD-Privacy algorithm, which improves classic CMC algorithm [32]. Firstly, we generate candidate query collection of the k-marginal table from D. Secondly, we weight each query collection m_k^i by their frequency. After that, the corresponding k-marginal tables are weighted by their query collection. Thirdly, A algorithm FMC (Frequent item sets Marginal table Covering algorithm) is presented to obtain a nearly optimal k-marginal tables through their weights. Finally, after the noising and consistency processing, we can get the released marginal table query middle ware.

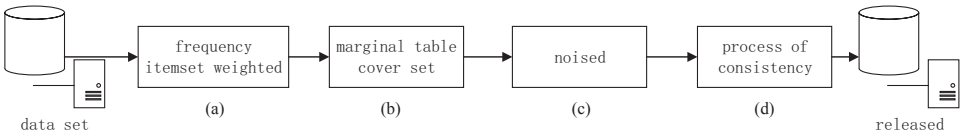


Fig. 1. The flow chart of differential privacy release algorithm based on regular marginal tables

3.2. Selection for the marginal table dimension *k*

The dimension k of marginal table is a main factor to the query cover. In this subsection, we discuss about the selection of k according to σ. The cover of marginal table cover set is σ_k, and the number of count queries that can be performed on n k-dimensional marginal tables. The cover σ_k is shown in Eq. 4.

$$\sigma_k = \frac{\left|\bigcup_{i=1}^{n} \binom{k}{i}\right|}{2^d - 1} \tag{4}$$

Cover Upper Bound It is easy to understand that σ_k can get maximal value when all m_k^i are disjoint as Eq. 5.

$$\sigma_d = \frac{\sum_{i=1}^{k} \binom{d}{i}}{2^d - 1} \tag{5}$$

As it is impossible that all m_k^i are disjoint, then

$$\sum_{i=1}^{k} \binom{d}{i} > (2^d - 1) \times \sigma \tag{6}$$

So, let $f(d,k) = \sum_{i=1}^{k} \binom{d}{i}$, we get Eq. 9 and Eq. 10.

$$f(d,k) = 2^{d-1} \exp^{\frac{(d-2k-2)^2}{4(1+k-d)}}, where\ k \leq \frac{d}{2} \tag{7}$$

$$f(d,k) = 2^{d-1}(2 - \exp^{\frac{(2k-d-2)^2}{4(1-k)}}), where\ \frac{d}{2} < k < d \tag{8}$$

$$\sigma \leq 0.5 * \exp^{\frac{(d-2k-2)^2}{4(1+k-d)}}, where\ k \leq \frac{d}{2} \tag{9}$$

$$\sigma \leq 0.5 * \exp^{2-\frac{(2k-d-2)^2}{4(1-k)}}, where\ \frac{d}{2} < k < d \tag{10}$$

Cover Lower Bound Since the weighted set cover problem is NP-hard, the result m_{app} obtained from FMC algorithm is an approximation of the optimal set m_{opt}, the approximate rate is $1 - 1/e$. The approximate cover is lower than optimal cover. As a result, we need to analyze the lower bound of optimal solution to choose the dimension of marginal table to avoid the loss caused by approximation of cover. The lower bound of $M_{opt}$is a k-marginal table cover set which can cover all k-1 way marginal table query combination, so $\sigma_{opt} > \frac{\sum_{i=1}^{k-1} \binom{d}{i}}{2^d - 1}$. From Eq. 7 and Eq. 8, we can get the lower limit of M_{app} as shown as Eq. 11 and Eq. 12.

$$\sigma_{app} > 0.5(1 - 1/e) * \exp^{\frac{(d-2k)^2}{4(k-d)}}, where\ k \leq \frac{d}{2} \tag{11}$$

$$\sigma_{app} > 0.5 * (1 - 1/e) \exp^{2+\frac{(2k-d-4)^2}{4k}}, where\ \frac{d}{2} < k < d \tag{12}$$

In summary, for a given σ, we can get the marginal table of dimension k according to Eq. 11 and Eq. 12.

We give a verification for the efficiency of k-marginal tables releasing. We use 6 test data whose dimensions are $\{15, 17, 19, 21, 23, 25\}$, and use differential dimensional marginal table to form the cover set to analyze the distribution of cover. The distribution is shown in Fig. 2. We can observe that the cover benefit is getting slower when the dimensions are getting larger. In summary, releasing the marginal table is useful for dimensional reduction.

3.3. Selection of Marginal Table and Error Analysis

After obtaining the dimension of marginal table from σ, we need to filter out the marginal table further. PriView is very useful, but it doesn't take the relationship between attributes

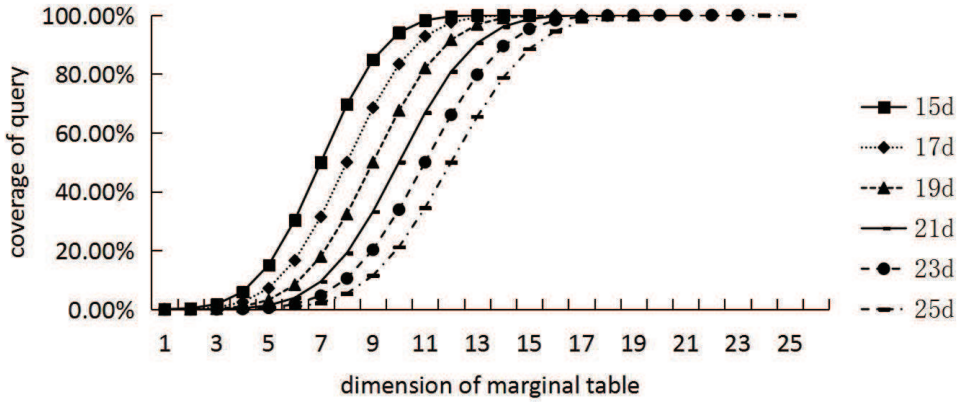


Fig. 2. Cover distribution of different marginal tables under different dimensions

into consideration. To improve this method, we analyze the released data set through frequent item sets. In this method, we can weight the marginal table, and the marginal benefits value of each marginal table are also taken into account. As a result, the marginal table is more reasonable to reduce the redundant marginal table and improve data availability.

Marginal Benefit We use $m(S)$ to represent the marginal benefit of a collection of k-marginal tables S [33]. $Mben(E, S)$ represents the increasing marginal benefit when adding another k-marginal table to S.

$$Mben(E, S) = m(S \cup E) - m(S). \tag{13}$$

It is usually hard to accurately compute $Mben(E, S)$ through Eq. 13. A good news is that $Mben(E, S)$ can be easily computed by the following Eq. 14, if only one k-marginal tables is contained in S.

$$Mben(E, S) = (2^s - 1) + (2^s - 1) \times (2^v - 1) = 2^k - 2^v \tag{14}$$

Where v is the size of the same attributes between E and S. Inspired by Eq. 14, we present a novel approach to approximately compute $Mben(E, S)$ when there are more than one k-marginal tables in S. The approach is shown in Algorithm 4. In order to further verify this approach, we compare the traditional method with the proposed approximate method in Table 4.

FMC. The FMC algorithm contains marginal table generation and a filtering process. In the marginal table preprocessing stage, this paper uses Apriori algorithm to extract the frequent items and attach weight to marginal tables with corresponding support. The algorithm of weighting marginal tables (Weight-MT) is shown in the algorithm 2.

Table 2. Error analysis table of marginal benefit estimation method

Dimension k	6	7	8	9	10
Average error rate	0.167169	0.208876	0.210311	0.207712	0.229283
Accurate Method (ms)	524	899	1691	2524	3107
$Mben_{app}$ (ms)	88	83	98	132	137

Algorithm 2: Weight-MT

Input : original data set D, dimension of marginal table k, min-sup s, N
Output: A_{sup}, M_{sup}

```
Freq = Apriori(D,s)
Mar_k= get k-marginal tables from D
for each mar in Mar_k do
    mar_weight = 0, count = 0
    for each f in Freq do
        if f⊂ any combination of m then
            mar_weight = +f.sup/ |D| ;
        count++;
    mar_weight = mar_weight * (count/n) A_sup.push(mar_weight);
    M_sup.add(mar, m_weight)
```

The dimension of original database D is 17, and we choose k from 6 to 10 to observe the computing time (the first two rows) and the error respectively. From Table 4, it is clear to see that $Mben_{app}$ is computed more faster than traditional method. Meanwhile, the relative error between the estimated value and the actual value is small. Although the relative error increases with the increase of the marginal table dimension, the error rate basically floats around 0.2. In the experiment we find that the trend value of the marginal benefit value obtained by the estimation method is almost the same as the real result, so the effect of the average error rate on the validity of the estimated value can be neglected.

Noise Error Analysis Similar to PriView, we use the expected squared error to evaluate the noise error of RD-privacy, denoted as ESE_{RD} in Eq. 15, where m is the number of the margin tables whose support are smaller than given threshold.

$$ESE_{RD} = 2^{k+1} \times \left(\binom{d}{k} - m/\varepsilon \right)^2 \tag{15}$$

3.4. The Proposed Marginal Table Covering Algorithm

Algorithm Design The weighted k-marginal tables covering problem defined in Definition 1 is similar to a weighted set cover problem. In 2015, Golab proposed a CMC algorithm [30], which is an effective solution to this kind of problem. Inspired by CMC, we proposed the FMC algorithm to solve our problem based on frequent items.

Algorithm 3: FMC

Input : original data set D, dimension of marginal table k,A_{sup},M_{sup},cover σ.
Output: the collection of marginal table Mar_{res}

1 Mar_k= get k-marginal tables from D
2 Mar=$\emptyset$
3 $Mben[mar]$=$\emptyset$
4 **for** each mar in Mar_k **do**
5 $Mben[mar] = Mben(mar, Mar_{res})$
6 **for** i = 1 to $|Mar_k|$ **do**
7 s = get marginal table whose benefit = max($Mben[A_{sup}]$);
8 **if** cover of $Mar_{res} \geq \sigma$ **then**
9 Return;
10 **if** M_{sup}[s] $< \frac{2k-1}{2}$ && $MBEN[s] < sup_t$ **then**
11 Break;
12 delete s from MBen and s in Mar_{res};
13 updateMBen according to algorithm 4

Algorithm 3 is the procedure of FMC. In line 1 to 3, it's the data initialization phase, we get the k marginal table candidate Mar_k from D, and then calculate each marginal benefit of k marginal table to $Mben[mar]$. Line 7 to 13 is the detail process of filtering candidate marginal table. In the beginning , Mar_{res} is empty, then all marginal benefit is the same. So we choose the max support weight marginal table. With the change of Mar_{res}, we update the $Mben[A_{sup}]$ in line 13 according to algorithm 4.

Algorithm 4: MBen

Input : marginal table mar_k, the collection of marginal table Mar_k, the length of marginal table k
Output: the marginal benefits Mben(mar_k,Mar_k)

1 $min = \infty$
2 sum = 0
3 i = 0
4 **for** s in Mar_k **do**
5 v = the number of same attribute in marginal mar_k compared with s
6 $Mben = 2^k - 2^v$
7 **if** $Mben < min$ **then**
8 $min = Mben$
9 $sum+ = 2^v - 1$
10 $i++$
11 return $min - (sum/i)$

After obtain the marginal table cover set Mar_{res}, we can then calculate the size of Mar_{res}. Add noise that follow $Laplace(|Mar_{res}|)$ into marginal tables to make the released middle ware satisfy the ε-differential privacy.

3.5. Consistency Analysis of k-marginal Tables

As the random noise is mixed, the direct publishing method of the marginal table covers the existence of inconsistent query results. Therefore, this section introduces consistency processing algorithms and the corresponding analysis.

We use the same way in PriView to consistency the query middle ware. However, the middle ware obtained by PriView is the query combination cover of the partial dimension contained in the marginal table. The marginal table cover set proposed by the algorithm based on the frequent item sets is completely used for all the query combinations in the marginal table, involving more query combinations. Therefore, for the marginal table cover set handled by the method in the same way, the difference between the privacy properties and the noise error of the published marginal table is different. Therefore, in this section, we analyze the consistency of the process, the marginal of this article to cover the set of differential privacy model of the impact of the query results and noise error.

Differential Privacy Analysis After determining the harmonization process duplicates of each target, each target in accordance with an marginal table is normalized. This process is carried out after the mixing process with Laplace noise, so we need to analyze whether privacy protection has been changed.

In order to determine whether the marginal table middle ware still retains the nature of differential privacy protection, the post-processing inefficiencies of the differential privacy model are introduced here, as shown in Theorem 1. Obviously, since the differential privacy model has post-processing inefficiencies, the subsequent consistency handling operation for the overlay marginal table satisfying the difference privacy does not affect the differential privacy protection feature of the overlay marginal table set itself.

$Theorem1$ (post-processing inefficiencies): If the algorithm S satisfies ε-differential privacy, then for any function, the new processing mechanism $M = \varphi(S(D))$ still satisfies ε-differential privacy.

Noise Error Analysis From previous literature [34] we can get following equation.

$$Q(M_c) = Q(M_c) + \frac{U(A) - Q(M)}{2^{|M|-|A|}} \tag{16}$$

We can know from consistency processing, the first step is to calculate the target value of the same query paradigm A from different marginal tables. Set query results in original data set is $U(A)$ and Laplace noise δ. After noise added process, the result from a k-marginal tables M_i is shown in Eq. 17. If there are j k-marginal tables that satisfy paradigm A, the consistent target is shown in Eq. 17.

$$Q(A) = \mu + \sum_{i=1}^{|M_i|-|A|} \delta_i \left(|A| \leq |M_i|\right) \tag{17}$$

$$U(A) = \mu + \frac{1}{j} \sum_{j=1}^{j} \sum_{i=1}^{|M_i|-|A|} \delta_i \left(|A| \leq |M_i|\right) \tag{18}$$

According to the Eq. 16, Eq. 17 and Eq. 18 we can obtain Eq. 19. Let $\eta = \sum_{i=1}^{|M_i|-|A|} \delta_i$. As the added noise follows Laplace distribution, then $E(\eta) \to 0$. Therefore, from the Eq. 19 we can see that, the whole of noise is reduced and valid of released data is improved while published data also satisfies ε-differential privacy.

$$Q(M_c) = \mu + \sum_{i=1}^{|M_i|-|A|} \delta_i + \frac{\sum_{i=1}^{|M_i|-|A|} \delta_i - \frac{1}{j} \sum_{j=1}^{j} \sum_{i=1}^{|M_i|-|A|} \delta_i}{2^{|M_i|-|A|}} \tag{19}$$

4. IM-Privacy Algorithm

In the previous description, the marginal table differential privacy publishing algorithm under frequent item sets uses many k-dimensional tables of the same dimension to ensure the data availability of the published data set at the expense of a certain cover rate. However, this method has certain limitations in the application scenario where the data privacy protection requirement is not high and the query range has high requirements. To solve this problem, we propose a differential privacy publishing algorithm partitioned by irregular marginal table to improve the PriView model from another perspective in this section. The algorithm composes a cover set by leveraging different marginal tables. By constraining the multi-level query rules, the amount of noise mixed in the publishing marginal table is controlled as much as possible to ensure the balance between privacy protection and data availability without sacrificing the scope of the query.

4.1. Global Sensitivity Analysis

The main idea of the proposed IM-Privacy method (Irregular partition Marginal table differential Privacy release algorithm) is to find n marginal tables that has 1 to d dimensions, which can completely cover all query combinations of D. The method can achieve differential privacy protection through the Laplace mechanism. The premise of the implementation of the Laplace mechanism is that the middle ware F has a clear global sensitivity. The calculation method is as shown in Eq. 20.

$$S(F) = \max_{D_1 D_2} (\sum_{f \in F} |f(D_1) - f(D_2)|) \tag{20}$$

Where D_1 and D_2 are adjacent data sets, and f is arbitrary query on F. If the publishing middle ware M_d composed of multidimensional marginal tables cannot determine the query strategy, the global sensitivity under the middle ware cannot be determined, and the differential privacy protection based on the Laplace mechanism cannot be performed. If the global sensitivity is too high, the noise mixed into the middle ware will be large, which may directly reduce the data availability. To solve this problem, this paper proposes the following constraint on the query strategy of the irregularly divided marginal table cover set. We can use the same-dimensional marginal tables to perform the combined query of the same size. If the query result cannot be obtained with the same-dimensional marginal table, then the result of last-dimension marginal table is used to get the query result. Based on this constraint, the formula for querying the query sensitivity $S(M_d)$ of the middle ware M_d is shown in Eq. 21. Thus, in the worst case, the global sensitivity for the query

middle ware under the constraint is $d+1$, which means the query operation with a $d-1$ attribute combination needs to use the d-dimensional marginal table, and the global sensitivity is D at this time. Therefore, under this constraint, $(S(M_d)/\lambda)$-differential privacy can be satisfied when the Laplace noise distribution is $Laplace(S(M_d)/\lambda)$.

$$S(M_d) = \max(1, C_k^{k-1} + 1) \tag{21}$$

In order to overcome this constraint, we need to further obtain the cover algorithm that satisfies the query policy constraint and minimizes the marginal table cover set ESE to get the irregular marginal table cover set M_d. Therefore, this paper obtains the maximum marginal benefit cover strategy for greedy approximation through the further analysis of ESE, and proposes an approximate optimal marginal table cover algorithm combining this strategy, which can get the approximate solution ρ compared with the optimal cover set M_{d-opt} under the premise of meeting the cover requirement.

4.2. Greedy Approximation Maximum Marginal Benefit Cover Strategy

In order to further improve the availability of the data, we need to find the optimal marginal table cover set that satisfies the query constraint, so that the ESE of the released middle ware is as small as possible. Obviously, the problem is a set cover problem that minimizes the objective function min (ESE). The definition of this minimization problem is shown in Eq. 22, where E is an approximate solution and S is an optimal solution. Since the table of different dimensions is used to form the release middle ware, the overall noise intensity needs to be analyzed. The calculation of ESE_m in this case is as shown in the formula Eq. 23, where $\mid m_i \mid$ represents the number of i-dimensional marginal tables.

$$\min\{ESE(S) : S \in E\} \tag{22}$$

$$ESE_m = 2^d(\sum_{i=1}^{d} \mid m_i \mid)Laplace(S(M_d)/\varepsilon) \tag{23}$$

It is difficult to directly estimate the lower bound of the optimal solution S because variables added are randomly distributed to the marginal table, so we cannot the estimate the approximate ratio of the approximation algorithm. However, it can be seen from Eq. 22 that there are two values affecting the change of ESE_m, the number of covered concentrated marginal tables $\sum_{i=1}^{d} \mid m_i \mid$ and the query sensitivity $S(M_d)$. Therefore, we analyze the marginal benefit value that directly affects the number of marginal tables and query sensitivity, and then transform the ESE minimization problem into the optimization problem of marginal benefit.

In the set cover problem of the marginal table, the k-dimensional marginal table benefit value relative to the cover set M is equal to the number of query combinations covered by the k-dimensional table and not covered by M. For marginal benefit and global sensitivity $S(M_d)$, the description of global sensitivity is a combination of queries covered by a marginal table. Therefore, the marginal benefit is directly proportional to the query sensitivity. In order to guarantee the query scope be completely covered, the cover set

M_d whose global sensitivity is $S(M_d) = d + 1$ needs to be added to the d-dimensional marginal table. Therefor we can further derive Eq. 24 according to the Eq. 23.

$$ESE_m = 2^d(\sum_{i=1}^{d} \mid m_i \mid)Laplace((d+1)/\varepsilon) \tag{24}$$

It can be seen from the ESE estimation formula of the marginal table that, while the published data set dimension d is determined, the overall ESE_m is only related to the number of tables. Thus the smaller the number of covered concentrated marginal tables, the smaller the overall ESE, the higher its data availability. The greater the marginal benefit value, the higher the query cover rate that a single marginal table can provide, the smaller the number of marginal tables when conditions to reach the cover rate are satisfied. As a result, the marginal benefit is inversely proportional to the number of marginal tables.

Therefore, the maximum marginal benefit can be obtained by the approximate solution of the optimal marginal table cover set. The process is as follows. Firstly, we select the marginal table with the largest margin benefit value when filtering the k-dimensional table for the first time. Secondly, the remaining k-dimensional marginal table is iteratively searched until only one table with the marginal benefit value of 1 is left in the dimension. The table with the largest margin benefit value is selected in each iteration. Then enter the $k+1$ dimension search process. This is because the table with the residual margin benefit value of 1 can only complete the cover of itself. Based on the query strategy proposed later of this paper, it can be covered by the $k + 1$ dimension marginal table.

In the choice of the overall idea of the algorithm, since the marginal benefit value of the marginal table changes to satisfy the submodule function, the greedy algorithm is selected to search the marginal table, and the approximate solution of the optimal marginal benefit value is obtained. In summary, this section transforms the ECE minimization problem into the marginal benefit maximization problem, and the optimal cover set S has the marginal benefit value $MBen(S) = 2^d - 1$, which means the optimal set covers all the query combination of d-dimensional data under the constraint query condition. Assuming that the marginal benefit of the approximate solution E obtained by the algorithm is $\sum_{e \in E} MBen(e)$, the approximation rate ρ is as shown in Eq. 25.

$$\rho = \min_{S:min-cov} \frac{\sum_{e \in E} MBen(e)}{MBen(S)} \tag{25}$$

4.3. Approximate Optimal Marginal Table Covering Algorithm

In this section we propose an approximate optimal marginal table covering algorithm to solve the marginal table cover set selection problem of the irregular partitioned differential privacy model. The key of the differential partitioning model with irregular division is to use the marginal table of different dimensions to form the cover set of the query range of the original data set. For a data set whose dimension is d, if we iteratively find the optimal marginal table set on all the 1-to-k-dimensional marginal tables, one iteration needs to be traversed $2^d - 1$ times in the worst case. As the dimension of the data set continues to increase, the number of iterations and traversals per iteration becomes larger, the time complexity we traverse the marginal tables of all the 1 to d dimensions directly

and update the marginal benefit values of the remaining tables will be difficult to estimate. Assuming that the number of iterations is n and the original data set dimension is d, the time complexity of the no-grouped algorithm that is $O(n(2^d-1)^2)$ in the worst case.

In order to reduce the time spent in marginal table traversal, we partition the tables according to their dimensions during algorithm implementation. We find the marginal table with the most marginal benefit in each group, and then iterate the remaining table until the marginal benefit value of the table is 1 or there is no remaining table. The pseudo code of the algorithm is shown in Algorithm 5 whose input is the original data set D, the query cover rate is $\sigma = 1$, and the output is the marginal table cover set that satisfies the cover requirement.

Algorithm 5: IM-Privacy

Input : a data set D, $\mid D \mid = d$, query cover fraction σ
Output: a collection of marginal table M

```
1  M = ∅
2  M_u n = ∅
3  for each i = 1 to d do
4      Margin_i = get i-way marginal tables
5      if i=1 then
6          choose ⌈|D|/2⌉ 1-way marginal tables
7      else
8          for each margin in Margin_i do
9              m = get marginal table with max marginal benefit
10             M.push(m)
11             delete m from Margin_i
12             repeat
13                 update marginal benefit of rest marginal table in Margin_i
14                 get 1 marginal table with max marginal benefit
15             until MBen(m) = 1 or |Margin_i| = 0 ;
16             put uncover marginal tables into M_un
17 for each m in M_un do
18     if MBen(m) ≠ 0 then
19         M.push(m)
20 return M
```

Algorithm 5 is the procedure of IM-Privacy. Line 1 to 5 are the data initialization and preparation phases, where M represents the marginal table cover set and $level$ represents the layer to be divided. We partition the group based on the dimensions of the marginal table. Line 7 is the processing of the 1-dimensional marginal table, and the first $\lceil d/2 \rceil$ marginal tables are directly selected from the 1-dimensional marginal table and added to the cover set M. Lines 11 to 20 are the iterative search process of the marginal table. First, we search the marginal table whose benefit is the highest in the i-th dimension table. Then we iterate the remaining marginal table in the i-th dimension, as shown in lines 13 to 16. Each marginal table is recalculated to figure out the marginal benefit of the remaining

table until the set is empty or only the table remains a marginal benefit value of 1. Lines 17 to 20 check the marginal table with a marginal benefit of 1. If there is a table with a marginal benefit of the relative cover set M greater than 0, it can be directly inserted into M to ensure the final published marginal table. The cover of the query set under the multi-level query strategy is 100%.

4.4. Time Complexity Analysis

The approximate optimal marginal table covering algorithm is the key of the differential privacy publishing algorithm based on the irregular marginal table. We analyze the time complexity of the algorithm. The algorithm consists of two parts, the search process of the 1-to-d-dimensional marginal table and the inspection process of the table with the residual marginal benefit of 1. The marginal table iterative search process is a traversal lookup in the marginal table of groups 1 to d, where the number of tables in the group is $n = C_d^k$. In the lookup process, the time consumption of updating the marginal benefit value of the table is related to the solution set size s of the marginal table cover set, and the time complexity of the process is $O(d \times n \times s)$. For the marginal table set $M_{un}(|M_k| = m)$ with the remaining marginal benefit value of 1, the time complexity of the traversal check is $O(m \times s)$. We can figure out that the overall time complexity of the marginal table algorithm is $O(s \times (d \times n + m))$, where $d \ll n$. So the time complexity is $O(s \times (n + m))$

5. Experiment

In this section, we will compare and analyze the two marginal table-based differential privacy publishing algorithms proposed in this paper. The experiment mainly analyzes the feasibility of differential privacy publishing algorithm based on improved cover set from two aspects: algorithm efficiency and data availability. By comparing with the representative differential privacy model, the advantages of the proposed method in improving data availability are verified. The detail of the hardware used in experiment is shown in table 3. We use public data set MSNBC and Kosarak to design experiments and make performance analysis.

Table 3. Experimental configuration

CPU	i7-6560U CPU @ 2.20GHz 2.21GHz
Memory	DDR4 8GB
Disk	256GB SSD
System image	Windows 10 64-bit operating system

MSNBC [35]: This data set is clicked record of web sites collected from msnbc.com and msn.com which contains 989,818 item sets. Each sequence in the data set corresponds to the page category that a user navigates in 24 hours.

Kosarak [36]: This data set records the click flow information of a Hungarian news web site, with 912,627 items. Each record in the data set represents a combination of news identifier that the user clicks.

Now, we describe the parameters used in the experiments, the privacy default budget $\lambda = 1$ and the dimension of marginal table k is chosen from$\{6, 8\}$ for the default cases.

In the rest of the section, we compare the proposed algorithm with the existing differential privacy algorithm in data availability and efficiency. Firstly, we compare the number of marginal tables generated by RD-privacy with the state-of-the-art method. We analyze the average support of the obtained marginal tables between these two methods, and make the error analysis in the last subsection.

5.1. Results on RD-Privacy

Analysis for Quantity of Marginal Table In this subsection, we select the SNBC to conduct experiment, which has 17-dimensional attributes. We use the Apriori algorithm to mine frequent itemsets (the Apriori algorithm sets the minsup as 0.005).

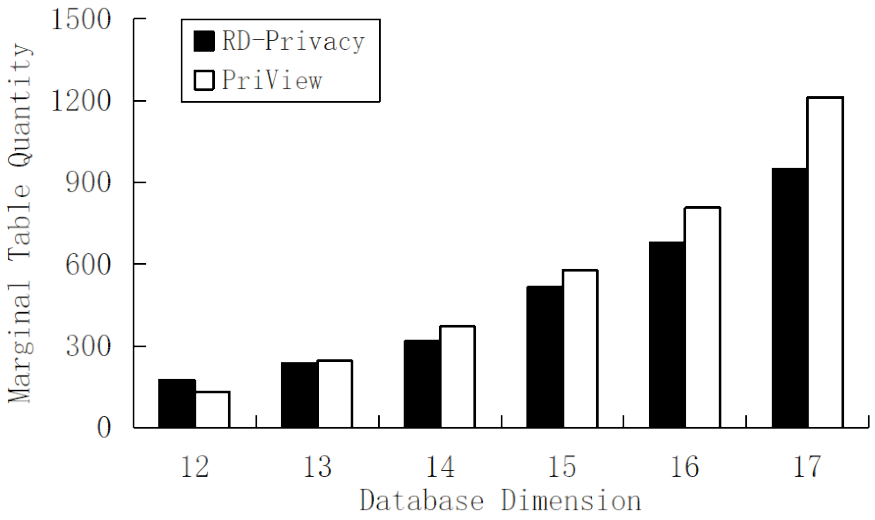


Fig. 3. Marginal table quantity distribution $(k = 6)$

As shown in Fig. 3 and Fig. 4, as the dimension increases, RD-Privacy finds fewer marginal tables than PriView does. The reason is clearly that RD-Privacy make use of Apriori, which can analyze the validation of query combination in MSNBC and delete the combination which have a low support. It is noticeable that when $d = 12$, RD-Privacy obtains more marginal tables because frequent itemsets cannot work better when dimension is low.

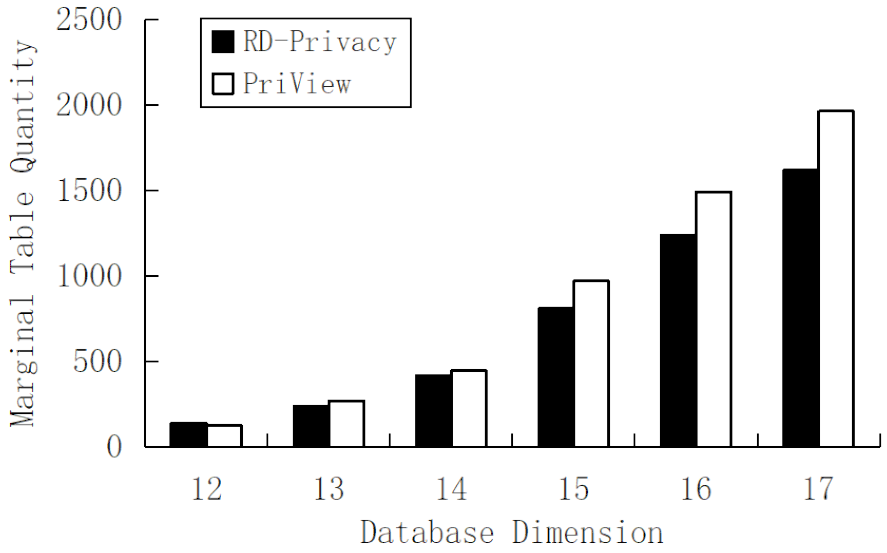


Fig. 4. Marginal table quantity distribution ($k = 8$)

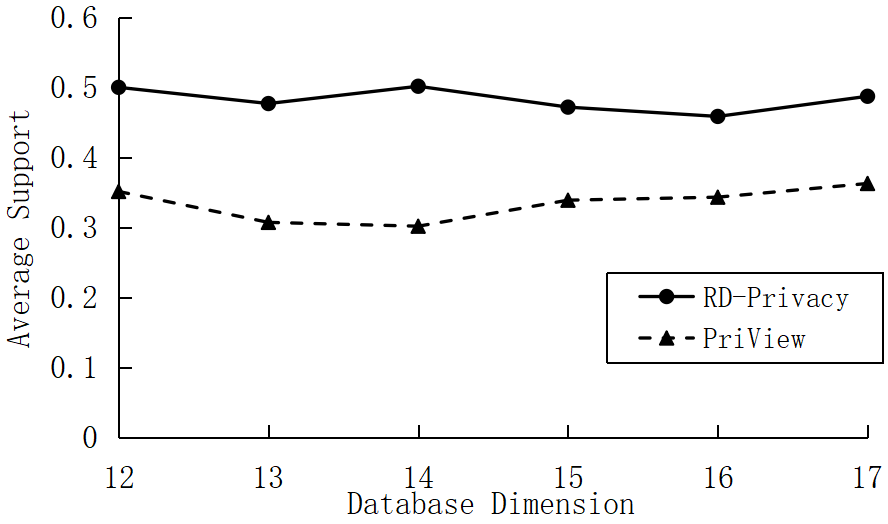


Fig. 5. Average support distribution ($k = 6$)

Analysis of Average Support We can see from Fig. 5 and Fig. 6, the proposed RD-Privacy can get marginal table with lower support which means the obtained marginal tables are more meaningful. When $k = 8$ in Fig. 6, the length of query combination become larger. Then average of support maturely become larger and the Priview still obtain lower support than proposed algorithm.

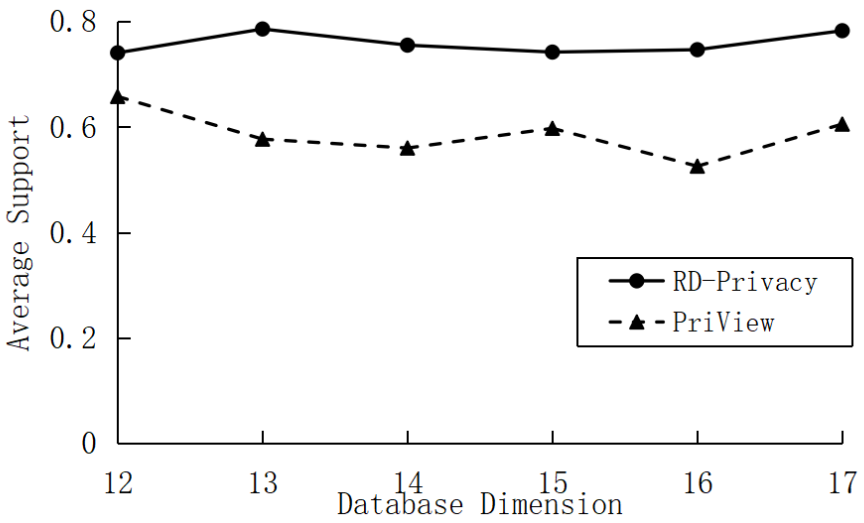


Fig. 6. Average support distribution ($k = 8$)

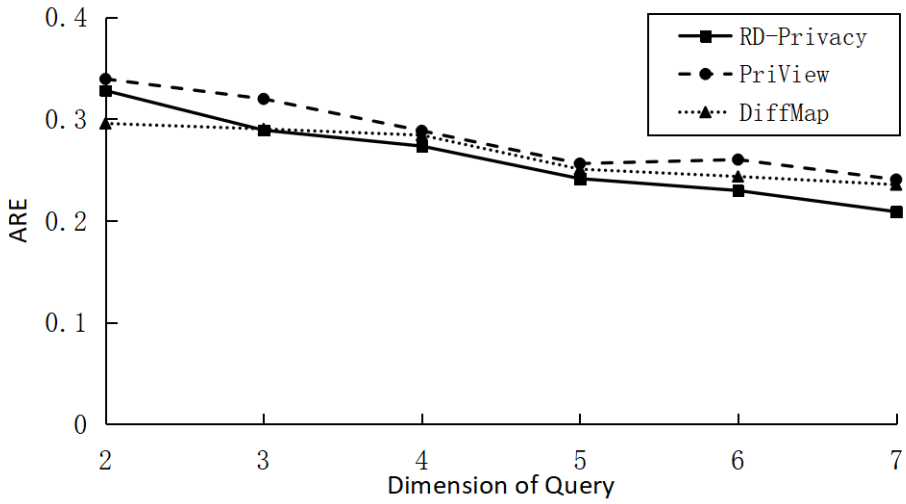


Fig. 7. ARE distribution ($\lambda = 1$)

Analysis of ARE We definite ARE as following Eq. 26, where A is the true value and B is the obtained value.

$$ARE = \frac{|A - B|}{A} \tag{26}$$

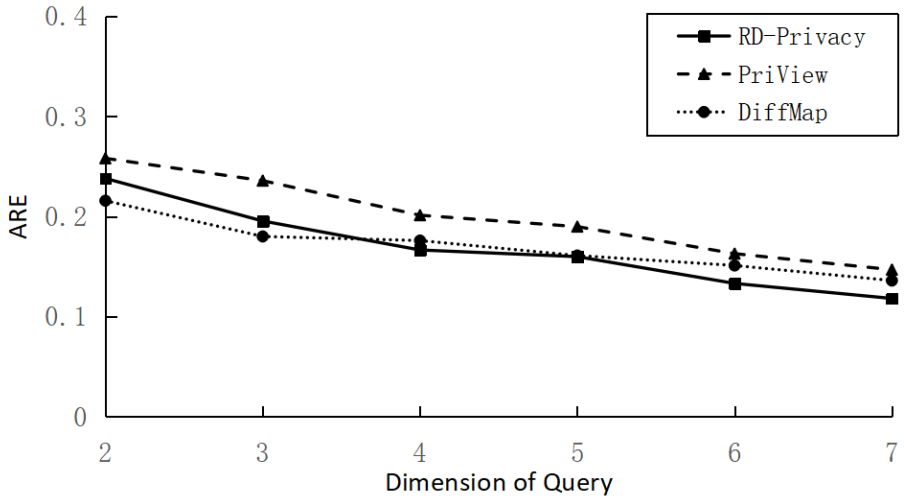


Fig. 8. ARE distribution ($\lambda = 2$)

We can infer from Fig. 7 and Fig. 8 that, in the most of case, the presented algorithm can obtain marginal table with less errors. It is noticeable that when the dimension is small the ARE of RD-Privacy is larger than others. The reason is that the RD-Privacy can obtain more marginal tables based on frequent item sets and then more noises are added.

5.2. Results on IM-Privacy

Experimental Analysis on IM-Privacy This part of the experiment is based on the comparison of IM-Privacy with PINQ and Dwork differential privacy models whose models are differential privacy models with a query cover of 100%. The comparative experiment uses these three algorithms to generate the query middle ware on the Kosarak data set, and then compares the average relative error between the query result and the true value.

The privacy budget in the experiment $\lambda = 1$, the data set used is a user click list of the first 21 categories of news from Kosarak. First, use the IM-Privacy, Dwork, and PINQ for the 21-dimensional Kosarrak data set to generate query middle ware that meets the same differential privacy strength. The IM-Privacy algorithm mainly publishes the middle ware of the marginal table query for the application scenario with high query cover requirements, and the algorithm uses the marginal table of different dimensions to construct the middle ware. There is no limitation of the query dimension by the RD-Privacy algorithm. In the comparative experiment of the algorithm, the setting of the query dimension is different from the former subsection. Due to the marginal table cover set obtained by the IM-Privacy algorithm, the query cover is wide. Therefore, this subsection needs to further investigate the data availability of the query middle ware under the higher query dimension.

In the comparison experiment, in order to verify the data availability under different query dimensions, this paper carries out 5 to 10 random attribute combinations, and each combination distribution performs 100 random count queries. The query paradigm is as

shown in Eq. 27. We record the results of each query with the actual values and then compare the average relative errors of the three different middle ware.

$$SELECT\ COUNT(^*)\ FROM\ D\ WHERE\ A_1 \in S_1\ AND\ A_2 \in S_2 \cdots\ A_m \in S_m \tag{27}$$

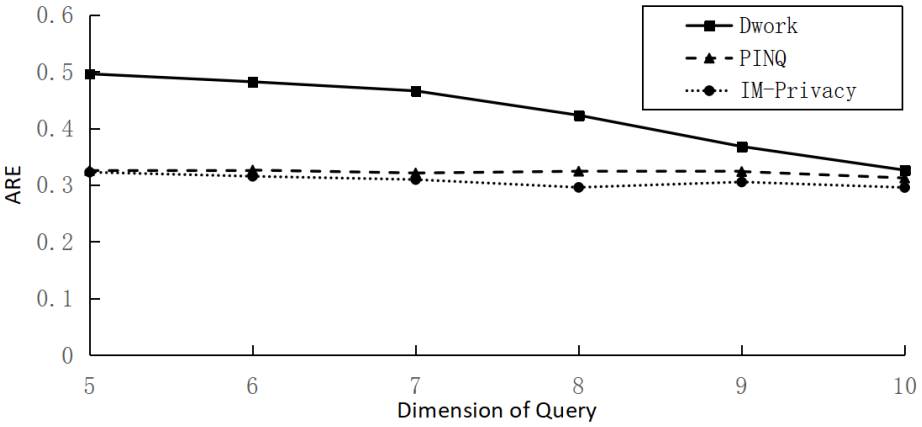


Fig. 9. Comparison chart of ARE for IM-Privacy, PINQ, Dwork

As can be seen from Fig. 9, the query brings more noise when performing a combined count query with a smaller number of attributes since Dwork is a Laplace noise directly added to the contingency table. Therefore, when the query dimensions are 5 to 7, the average relative error of the query results is much higher than the PINQ and the IM-Privacy release algorithm proposed in this paper. PINQ is a way to add noise directly to the query results. IM-Privacy is a rule-based query method (in the k-dimensional marginal table or $(k+1)$-dimensional marginal table with the same size of the combined query). PINQ will only bring the noise in the single query result while IM-Privacy brings noise of two results in the middleware at most, so the average relative error distribution is stable and lower than Dwork's traditional differential privacy model. However, in order to achieve the same level of privacy protection, the global sensitivity of IM-Privacy calculation is lower than half of PINQ, which means the Laplace noise added by IM-Privacy is less than PINQ although the noise of the two results may be brought at most. As a result, the overall average relative error of IM-Privacy is lower than the PINQ. With the increasing of query dimension, the number of items in the Dwork cascade query is gradually reduced. At this time, it is similar to the PINQ method which directly adds noise to the query result. Therefore, when the query dimension is 10, the average relative error of PINQ and Dwork is similar, but the Dwork approach introduces more noise and the query results are more unstable which leads to low data availability. Thus, the proposed method in this paper performs better in improving data availability.

Approximate Rate on IM-Privacy The approximate optimal marginal table covering algorithm in this paper is proposed for the optimal marginal table cover set. Its approximation rate is affected by the size of the data set. Therefore, it is necessary to consider the approximate ratio of the algorithm. We should calculate the ratio between actual irregular marginal table cover set and optimal one in theory. Meanwhile, in order to verify the availability of proposed algorithm, we will also figure out the effect of the change on data set size to approximate ratio stability.

The data set used in the experiment is MSNBC. We select 13 to 17 attributes randomly to form MSNBC of different dimensions and use the IM-Privacy algorithm to find the cover set composed of irregular marginal tables. Finally, we calculate the sum of the marginal benefit values of the marginal table and compare with the theoretical optimal marginal benefit value to analyze the approximation rate.

It can be seen from Table 4 that the overall marginal benefit value obtained by the algorithm is always between 0.8 and 0.9 as the increasing of data dimension, what's more, the ratio is always higher than the theoretical optimal marginal benefit value. It is proved that the approximation algorithm has certain advantages in the stability of its approximation degree and the availabilities of the algorithm.

Table 4. Change of approximation rate for IM-Privacy algorithm

Dimension k	13	14	15	16	17
Optimal marginal benefit	8191	16383	32767	65535	131071
Actual marginal benefit	7127	14376	28881	58169	116467
Approximate rate	0.87010133	0.877494964	0.881405072	0.887602	0.88857947

6. Conclusion and Overlook

The development of the information industry has brought convenience to the office and life of each of us, and it has also created hidden dangers of user data leakage. However, The data publishing process becomes less secure and may result in user privacy leaks. The traditional privacy protection method can not meet the privacy protection requirements in the current environment. Therefore, new differential privacy model is widely used in the data release process with privacy protection requirements. Traditional middle ware for statistical data-based differential privacy algorithms contain more noise and lower data availability due to the differential privacy model that uses the noise to protect data. At present, although the marginal table based differential privacy model effectively reduces the noise, it does not consider the relevance of the attributes in the real data set. Meanwhile, the table cover set contains some invalid query combinations, which reduces the data availability. In the mean time, only the table of the same dimension is selected, which can not meet the needs of practical applications.

To settle these issues, this paper proposes a differential privacy publishing algorithm for regular marginal table differential (RD-Privacy) and irregular marginal table (IM-Privacy) under frequent item sets.

The differential privacy releasing algorithm based on regular marginal tables under frequent item sets uses Apriori algorithm to analyze the actual application data set, comprehensively considers the marginal table support degree and marginal benefit, and provides targeted cover for effective query combination, further improving the data availability of middle ware of the query.

Considering on the low requirements of data privacy protection but high requirements of the cover, a differential privacy publishing algorithm based on irregular marginal table partitioning is proposed. Using the approximate optimal marginal table covering algorithm proposed in this paper, we find that the multi-level edge is satisfied. The table query cover set of the table query policy constraint achieves a balance between privacy protection and data availability to a certain extent.

The paper only studies the differential privacy model under the count query, we can further expand the algorithm to the field of subgraph area. The paper focuses on the research of differential privacy protection for numerical statistical data. We can further study the statistical data of multi-category and multi-valued attributes, and obtain a more applicable method of constructing the marginal table differential privacy model.

Acknowledgment Supported by the Double-class Construction Innovation Project 01431-90518. Supported by the National Nature Science Foundation of China 61370198, 6137-0199, 61672379 and 61300187. Supported by the Liaoning Provincial Natural Science Foundation of China NO. 2019-MS-028.

References

1. Lyu, M., Su, D., Li, N.: Understanding the Sparse Vector Technique for Differential Privacy, PVLDB, 637-648.(2017)
2. Kong, W., Lei, Y., Ma, J.: Data security and privacy information challenges in cloud computing, International Journal of Computational Science and Engineering, 16(3), 212-215.(2018)
3. Shynu, P. G., Singh, K. J.: Privacy preserving secret key extraction protocol for multi-authority attribute-based encryption techniques in cloud computing, International Journal of Embedded Systems, 10(4), 287-300.(2018)
4. Dwork, C.: Differential Privacy, ICALP, 63(6), 1-12.(2006)
5. Atzori, M.: Weak k-anonymity: A low-distortion model for protecting privacy, International Conference on Information Security, 60-71.(2006)
6. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression, (1998)
7. Omran, E., Bokma, A., Abu-Almaati, S.: A k-anonymity based semantic model for protecting personal information and privacy, Advance Computing Conference, 2009. IACC 2009. IEEE International, 1443-1447.(2009)
8. Sweeney, L.: K-anonymity: A model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst, 10(5), 557-570.(2014)
9. Han, J., Yu, H., Yu, J.: An improved l-diversity model for numerical sensitive attributes, 2008 Third International Conference on Communications and Networking in China, 938-943.(2008)
10. Kroc, L., Sabharwal, A., Selman, B.: Leveraging belief propagation, backtrack search, and statistics for model counting, International Conference on Integration of Ai and Or Techniques in Constraint Programming for Combinatorial Optimization Problems, 127-141.(2008)
11. Ohkubo, J.: Basics of counting statistics, Ieice Transactions on Communications, E96.B(112014), 2733-2740.(2013)

12. Wang, H., Chen, Y., Li, L., Pan, H., Gu, X., Liang, Z.: A novel colon wall flattening model for computed tomographic colonography: Method and validation, Comput Methods Biomech Biomed Eng Imaging Vis, 13(4), 1-14.(2014)
13. Li, C., Hay, M., Rastogi, V., Miklau, G., Mcgregor, A.: Optimizing linear counting queries under differential privacy, Twenty-Ninth ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems, PODS 2010, June 6-11, 2010, Indianapolis, Indiana, Usa, 123-134.(2010)
14. Edoh-Alove, E., Bimonte, S., Pinet, F., Bdard, Y.: New design approach to handle spatial vagueness in spatial olap datacubes, International Journal of Agricultural and Environmental Information Systems, 6(3), 29-149.(2015)
15. Dwork, C.: Differential privacy: a survey of results, International Conference on Theory and Applications of MODELS of Computation, 1-19.(2008)
16. Lu, S., Mandava, G., Yan, G., et al.: An exact algorithm for finding cancer driver somatic genome alterations: the weighted mutually exclusive maximum set cover problem, Algorithms for Molecular Biology, 11(1), 1.(2016)
17. Al-Shihabi, S., Arafeh, M., Barghash, M.: An improved hybrid algorithm for the set covering problem, Computers & Industrial Engineering, 85, 328-334.(2015)
18. Cheng, T. C. E., Shafransky, Y., Ng, C. T.: An alternative approach for proving the NP-hardness of optimization problems, European Journal of Operational Research, 248(1), 52-58.(2016)
19. Das, Aparna, Claire M.: A quasipolynomial time approximation scheme for Euclidean capacitated vehicle routing, Algorithmica, 73(1), 115-142.(2015)
20. Feldman, M., Naor, J., Schwartz, R.: A unified continuous greedy algorithm for submodular maximization[C], Foundations of Computer Science (FOCS). IEEE, Palm Springs, CA, USA, 570-579.(2011)
21. Sagnol, G.: Approximation of a maximum-submodular-coverage problem involving spectral functions, with application to experimental designs[J], Discrete Applied Mathematics, 161(1), 258-276.(2013)
22. Wan, P. J., Du, D. Z., Pardalos, P., et al.: Greedy approximations for minimum submodular cover with submodular cost, Computational Optimization and Applications, 45(2), 463-474.(2010)
23. Qardaji, W., Yang, W., Li, N.: Priview:practical differentially private release of marginal contingency tables, 1435-1446.(2014)
24. Zhang, Q., Li, F., Yi, K.: Finding frequent items in probabilistic data, ACM International Conference on Management of Data (SIGMOD), 63(6), 819-832.(2008)
25. Bernecker, T., Kriegel, H. P., Renz, M.,Verhein, F., Zfle, A.: Probabilistic frequent pattern growth for itemset mining in uncertain databases, International Conference on Scientific and Statistical Database Management (SSDBM), 38-55.(2010)
26. Wang, L., Cheung, W. L., Cheng, R., Lee, S. D., Yang, X. S.: Efficient mining of frequent item sets on large uncertain databases, IEEE Transactions on Knowledge and Data Engineering, 2170-2183.(2012)
27. Tong, Y., Chen, L., Cheng, Y., Yu, P. S.: Mining frequent itemsets over uncertain databases, Proceedings of the Very Large Database (VLDB), 1650-1661.(2012)
28. Tang, P., Peterson, E. A.: Mining probabilistic frequent closed itemsets in uncertain databases, Southeast Regional Conference, 86-91.(2011)
29. Agrawal, R., Imieliski, T., Swami, A.: Mining association rules between sets of items in large databases, ACM SIGMOD Record, 22(2), 207-216.(1993)
30. Qardaji, W., Yang, W., Li, N.: PriView: practical differentially private release of marginal contingency tables, Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, Snowbird, UT, USA, 1435-1446.(2014)
31. Fienberg, S. E., Rinaldo, A., Yang, X.: Differential privacy and the risk-utility tradeoff for multi-dimensional contingency tables, International Conference on Privacy in Statistical Databases. Springer Berlin Heidelberg, 187-199.(2010)

32. Golab, L., Korn, F., Li, F., Saha, B., Srivastava, D.: Size-constrained weighted set cover, IEEE, 879-890.(2015)
33. Emek, Y., Rosn, A.: Semi-streaming set cover, International Colloquium on Automata, Languages, and Programming, 453-464.(2014)
34. Hay, M., Rastogi, V., Miklau, G., Dan, S.: Boosting the accuracy of differentially private histograms through consistency, Proceedings of the Vldb Endowment, 3(1-2), 1021-1032.(2009)
35. http://archive.ics.uci.edu/ml/datasets/MSNBC.com+Anonymous+Web+Data
36. Frequent itemset mining data set repository, http://fimi.ua.ac.be/data/.

Haoze Lv is currently in school with the Department of Computer Science in Dalian Maritime University. His research mainly focuses on Big data and privacy protection.

Zhaobin Liu (corresponding author) received the Ph.D degree in computer science from Huazhong University of Science and Technology, China, in 2004. He is currently a Professor in the school of information science and technology, Dalian Maritime University, China. He has been a Senior Visiting Scientist at The University of Auckland, New Zealand, in 2017, and a visiting scholar at University of Central Florida, USA, in 2014 and University of Otago, New Zealand in 2008 respectively. His research interests include big data, cloud computing and data privacy.

Zhonglian Hu is currently pursuing the master's degree with the Department of Computer Science in Dalian Maritime University. His research mainly focuses on Big data and privacy protection.

Lihai Nie is currently an doctoral candidate in the division of Intelligence and Computing, Tianjin University, China. His research interests include data mining, artificial intelligence, network and clouding computing. He has published more than several papers in international journals and conferences.

Weijiang Liu received the Ph.D. degree in computational mathematics from Jilin University in 1998. From June 2004 to June 2006, he was a postdoctoral fellow of Post Doctoral Station for Computer Science and Technology, Southeast University, China. He is currently a professor in School of Information Technology, Dalian Maritime University, China. He has published more than 80 papers and his current research interests include network measurement, network security, and mobile computing.

Xinfeng Ye gained BSc in Computer Science from Hua Qiao University, China, in 1987, and MSc and PhD in Computer Science from The University of Manchester, England, in 1988 and 1991 respectively. He is currently a senior lecturer in the Department of Computer Science at The University of Auckland, New Zealand. His current research interests include cloud computing and system security.

Received: September 15, 2018; Accepted: July 23, 2019.

Computer Science and Information Systems 16(3):733–751 https://doi.org/10.2298/CSIS180918024C

A Novel SMP-based Survivability Evaluation Metric and Approach in Wireless Sensor Network

Hongsong Chen[1], Haiyan Zhuang[2], Zhiguang Shan[3], Chao-Hsien Lee[4], and Zhongchuan Fu[5]

[1] School of Computer & Communication Engineering, University of Science and Technology Beijing
Beijing 100083,China
chenhs@ustb.edu.cn
[2] Railway Police College
Zhengzhou, Henan, 450053, China
zhuanghaiyan@rpc.edu.cn
[3] State Information Center of China
Beijing 100045, China
shanzg@sic.gov.cn
[4] Department of Electronic Engineering, National Taipei University of Technology
Taipei City, China
chlee@ntut.edu.tw
[5] Department of computer science, Harbin Institute of Technology
Harbin 150001, China
15124585561@163.com

Abstract. In Industrial Internet of Things (IIoT) device and network, wireless sensor network (WSN) is an important component.Routing protocol is the critical component of WSN. As the WSN may be attacked by all kinds of intruders, the survivability of WSN is important to IIoT security. To precisely evaluate the systematic survivability ability under external attack and internal security mechanism,a novel survivability entropy-based quantitative evaluation metric is proposed to calculate the systematic survivability ability of WSN routing protocol. Numerical analysis and simulation experiments are combined to precisely calculate the survivability entropy metric.To validate the evaluation approach, NS2 (Network simulator) is used to simulate the DoS attack and security mechanism in WSN. Experimental results show that the novel survivability evaluation metric and method can precisely evaluate the systematic survivability ability of WSN.

Keywords: survivability entropy, quantitative evaluation, systematic survivability, wireless sensor network.

1. Introduction

With the rapid development of Industrial Internet of Things (IIoT), many smart devices are connected to Internet. With the convenience brought by the IoT and services, potential security problems and threats exist in the IIoT applications. In IIoT device and network, wireless sensor network is an important part of IIoT. Wireless Sensor Network (WSN) is applied in many aspects, such as industry automatic control, environment monitoring,

smart home system. However, the security problems are hindering the wide application of WSN.

Routing protocol is the critical component of wireless sensor network (WSN). If the routing protocol is attacked, the survivability of WSN will be affected. As the attack behaviours are random, the consequences of attack are uncertain, it is difficult to quantitatively evaluate the change of survivability when the routing of WSN is attacked. There are some research progresses on the survivability of WSN, however, how to precisely evaluate the survivability of WSN routing protocol is a great challenge in current WSN security research area.

The generally definition of information system survivability was introduced by Ellison et al. [1]: Survivability is the ability of a network computing system to provide essential services in the presence of attacks and failures, and recover full services in a timely manner. In the Federal Standard 1037C. [2], survivability is defined as the property of a system, subsystem,equipment, process, or procedure that provides a defined degree of assurance that the named entity will continue to function during and after a natural or man-made disturbance. Although the definitions provide a good description of the concept of survivability, they do not provide mathematical precision to quantitative description of survivability. It is difficult to compare the survivability quantitatively by experimental method [3].

The remainder of this paper is organized as follows: in Section 2, related works and problem statement are analyzed. Survivability model for wireless sensor network routing protocol is described and derived in section 3. SMP-based Survivability evaluation method and novel metric are proposed in section 4. Simulation experiment and survivability precise evaluation are calculated in Section 5. The conclusions are made in Section 6.

2. Related Works and Problem Statement

2.1. Related Works

As the complexities of network systems, it is difficult to model and analyze the attacks for systematically analyzing network survivability ability. Xing Fei and Wang Wenye use semi-Markov process model to characterize the evolution procedure of node behaviors. The Semi-Markov-based node behavior model can be used as a bridge among some dynamic factors, such as node mobility or attack behaviour and network survivability [4]. They establish a survivability index system based on the Analytic Network Process (ANP) [5]. The ANP-based model of survivability index system was proposed to assess the survivability of Wireless Sensor Network in emergency communications.However, they do not provide experiment validation to the survivability model.

KIM Dong Seong etc have presented a survivability model framework for Wireless Sensor Network(WSN). The approach uses software rejuvenation to rejuvenate the sensor nodes under attack or/and compromised in a wireless sensor network [6]. They analyzed their model in mathematical manner and showed that software rejuvenation mechanism based on SMP and Discrete Time Markov Chain (DTMC) can decrease the failure probability while increases the probability of the system stayed in healthy state. Their model analysis is based on numerical analysis, the results can not reflect the real network scenario efficiently. Survivability model for cluster-based WSN was proposed, in which the

state of each cluster is regarded as a stochastic process based on a Semi-Markov Process (SMP) and Discrete Time Markov Chain (DTMC) [7]. The isolation problem between clusters is researched and discussed. Quantitative survivability is calculated based on k connectivity metric. Numerical results show the model is effective. Experiment analysis should be considered to make the research results more effective.

Denial-of-Service and Black hole attacks are the two main problems in the security of ad hoc network. There are not satisfied solutions to solve the problems [8]. A novel multi-agent-based dynamic lifetime intrusion detection and response scheme are proposed to counter against the two types of attacks. Systematic impact and survivability metric should be considered in wireless network applications.The security of WSN and big data are the foundation of smart city [9]. Data security should be combined in the network security and survivability in smart city.

Attack tree based approach and stochastic model based approach are two main threat modeling method [10]. In tree modeling approach, the root node representing the attack goal and leaf nodes representing the ways of achieving the attack goal. Stochastic model based threat modeling approaches transform system state models to Markov chains and analyze them using finite state transition matrix or game theory. While experiment data should be filled into the model to make the method more effective.

A quantitative protection effectiveness evaluation method based on entropy theory is introduced [11]. They propose the protection intensity model, which can be used to compute the protection intensity of a stationary or moving object provided by a secure network. They presents a method for anomaly detection and classification based on Shannon, Rényi and Tsallis entropies of selected features, the construction of regions from entropy data using Mahalanobis distance [12]. They use One Class Support Vector Machine (OC-SVM) with different kernels (Radial Basis Function and Mahalanobis Kernel) for normal and abnormal traffic detection. Entropy is used to measure and analyze network traffic. Public transit network is a typical complex network with scale-free and small-world characteristics[13]. In order to analyze the survivability of public transit network, Fu Bai-Bai et al define new network structure entropy based on betweenness importance, the ”inflexion zone” is discovered which can be taken as the momentous indicator to determine the public transit network failure. The research object is about public transit network, the network survivability is dependent on the network structure parameters.

Dagdeviren O, Akram V K provides two localized distributed algorithms for determining the states of nodes. The first proposed algorithm identifies most of the critical and noncritical dominator nodes from two-hop local subgraph and connected dominating set information [17].

Xiue Gao and Keqiu Li propose a new method of evaluating the survivability of military heterogeneous networks, based on network structure entropy. A model of survivability is proposed based on the network irreversibility which considers not only the nodes but also the edges [18].

Security model in wireless sensor network and cloud computing are proposed and researched [19,20,21] They can be references to the security model in WSN.

2.2. Problem Statement

SMP and DTMC stochastic models can be used to build survivability evaluation model in wireless sensor network. Steady state probability and Mean Time To Security Fail-

ure (MTTST) are evaluation metrics of survivability model in stochastic process theory. Mean sojourn time and transition probabilities are the necessary parameters when calculating the steady state probability and MTTSF. In current research,as the attack behaviours are random and unpredictable, mean sojourn time and transition probabilities are computed by numerical analysis method. If we want to evaluate the survivability metrics precisely,experimental method should be used to obtain the mean sojourn time and transition probabilities in the survivability model.

However, the mean sojourn time and transition probabilities are difficult to be obtained by experimental method, thus steady state probability and MTTSF are difficult to be calculated precisely by traditional method. At the same time, traditional survivability metrics can not reflect the systematic changes caused by network attacks and security techniques. The changes of WSN routing protocol states will cause the change of network survivability ability. When the attackers intrude the routing protocol of wireless sensor network, how to precisely evaluate the attack effects to survivability ability of the WSN routing protocol is a great challenge. The research object is to propose a novel survivability evaluation experimental approach and metric in WSN routing protocol, the novel survivability evaluation approach and metric can reflect the dynamic characteristic under network attack and security mechanism.At the same time , the survivability metric can be measured by experimental method and calculated by mathematical formula.

3. Building the Survivability Model for Wireless Sensor Network Routing Protocol

3.1. Survivability Models in Wireless Sensor Network

Routing protocol is the key component of wireless sensor network, if the routing protocol is attacked, the wireless sensor network can not work normally, the survivability of the network will be affected greatly. The Ad hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multi-hop routing network. It offers quick adaptation to dynamic link conditions, low processing and memory overhead and determines unicast routes to destinations. Ad hoc On-Demand Multi-path Distance Vector (AOMDV) is the extended work of AODV routing [16]. AOMDV provide multipath to reach the destination, AOMDV is designed to solve the connectivity problem due to highly dynamic network topology. It provides multipath for data packets delivery from the source to the destination to mask the attack to route path. The routing protocols can be used to build routing paths of wireless sensor network. The survivability of AOMDV is higher than that of AODV because of the multi-paths characteristic of AOMDV routing protocol.

Survivability model should be built quantitatively and precisely to describe the change of survivability ability. Three types of survivability models are used to describe the survivability of wireless sensor network routing protocol. They are finite state machine model, DTMC model and SMP model. They are introduced in the following sections respectively. Dynamic transition behaviors of WSN routing protocol are described in finite state machine model. State transition probabilities are described in DTMC model. Steady-state probabilities and mean sojourn times are described in SMP model.

3.2. Finite State Machine(FSM)-based Survivability Model for Security AOMDV Routing Protocol

The routing protocol of wireless sensor network may suffer from all kinds of attacks, such as flooding attack, DoS (Denial of service) attack, Sybil attack, impersonation attack. The attacks will affect the survivability of routing protocol, multi-path routing and intrusion detection algorithm will be used to be against the attacks. So security AOMDV protocol will be in different states under different attack types and security techniques. Finite state machine model is used to describe the state transition process under different attacks and security schemes. The model is shown in figure 1.

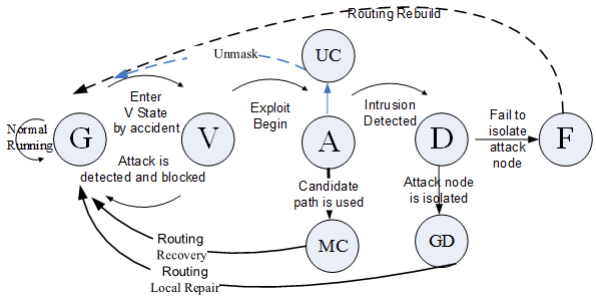


Fig. 1. FSM-based survivability model for security AOMDV routing protocol

As shown in figure 1, there are eight states in FSM-based survivability model of security AOMDV routing protocol. The state starts from good state to failed state, all the eight states come into being the whole life cycle of AOMDV routing protocol. They are listed in the table 1.

Table 1. Finite states and description

The name of state	Description of state
G	Good state
V	Vulnerable state
A	Attack state
MC	Masked compromised state
UC	Undetected compromised state
D	Detection state
GD	Graceful degradation state
F	Failed state

When AOMDV routing protocol runs normally, it is in Good state. When the attacker attempts to probe the network service and find the vulnerability, the system will enter the vulnerable state. If the probing behavior is detected and blocked, the system will be

back to Good state. If the vulnerability is exploited by the attacker, the system will enter the attack state. As multi-paths can be provided by AOMDV routing protocol, if current routing path is attacked and the intrusion is in local scope, candidate path will be used to mask the attack. If the strength of attack is strong, the intrusion can be detected timely, the system will enter the detection state. When the intrusion can not be detected and masked efficiently, the system will enter the undetected compromised state. The system need to be reconfigured to recovery to good state. When the system enters into detection state, if the attack node can be isolated and controlled successfully, routing protocol can be back to good state; otherwise, the system will enter the failed state and signal an alarm, it needs rebuild routing to recovery to the good state. So the eight states reflect all possible conditions of AOMDV routing protocol under different attack and security techniques.

3.3. DTMC-based Survivability Model for Security AOMDV Routing Protocol

As the attack behaviors are random and the network topology is complex in WSN, the future state of system only depends on the current state, the state transition process has no relation to the past state, state transition process of the system meets the character of Markov process. As the state transition process of the system can be mapped to discrete time sequence, the transition of states can be represented by a serial of probabilities. The survivability model of system can be described by Discrete Time Markov Chain (DTMC). The DTMC-based survivability model for security AOMDC routing protocol is shown in figure 2,the system was given that the routing protocol was vulnerable, the DTMC model was consisted by a set of states and probabilities, only real lines are assigned to state transition probabilities because of multi possible states.

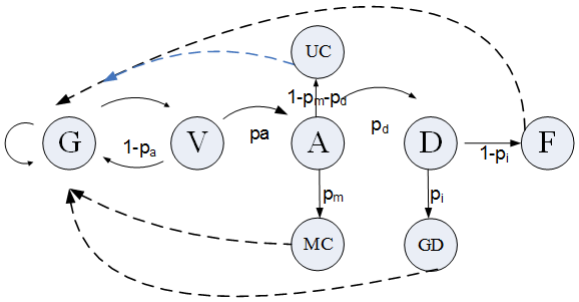


Fig. 2. DTMC-based survivability model for security AOMDV routing protocol

The state transition probability of DTMC model of security AOMDV routing protocol is shown in table 2.

The steady-state probabilities vector $\overline{v}$ of DTMC model can be computed as:

$$\overline{v} = \overline{v} \cdot P \tag{1}$$

Where $\overline{v}_{=} [v_G, v_V, v_A, v_{MC}, v_{UC}, v_D, v_{GD}, v_F]$ and P is the DTMC state transition probability matrix, $\overline{v}$ stands for a eight dimensions row vector. The state transition probability

Table 2. Finite states and description

p_a	State transition probability from vulnerability state to attack state
p_m	State transition probability from attack state to Masked compromised state
p_d	State transition probability from attack state to detection state
$1-p_m-p_d$	State transition probability from attack state to Undetected compromised state
p_i	State transition probability from detection state to grace degradation state
$1-p_i$	State transition probability from detection state to failed state
p_a	State transition probability from vulnerability state to attack state
$1-p_a$	State transition probability from vulnerability state to good state

matrix P describes the DTMC state transition probabilities between DTMC states which is shown in figure 2. The matrix P can be written as the formula2:

$$P = \begin{array}{c} \\ G \\ V \\ A \\ MC \\ UC \\ D \\ GD \\ F \end{array} \begin{array}{c} \begin{array}{cccccccc} G & V & A & MC & UC & D & GD & F \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \widetilde{p}_a & 0 & p_a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_m & \widetilde{p}_{md} & p_d & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_i & \widetilde{p}_i \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}, \begin{cases} \widetilde{p}_a = 1 - p_a \\ \widetilde{p}_{md} = 1 - p_m - p_d \\ \widetilde{p}_i = 1 - p_i \end{cases} \tag{2}$$

Steady-state probability vector $\overline{v}$ should be satisfy with the constrain condition(3)

$$\sum_i v_i = 1, i \in \{G, V, A, MC, UC, D, GD, F\} \tag{3}$$

The DTMC steady-state probabilities can be theoretically deduced by equation 1,they are shown in formula 4.

$$\begin{aligned} v_G &= v_V(1 - P_A) + v_{MC} + v_{UC} + v_{GD} + v_F \\ v_V &= v_G \\ v_A &= v_V P_a \\ v_{MC} &= v_A P_m \\ v_{UC} &= v_A(1 - P_m - P_d) \\ v_D &= v_A P_d \\ v_{GD} &= v_D P_i \\ v_F &= v_D(1 - P_i) \end{aligned} \tag{4}$$

Conjunction with the equation 3 and equation 4,we can get the mathematical relationship between steady state probability and state transition probability, so steady-state probability v can be solved as formula 5:

$$
\begin{aligned}
v_G &= \frac{1}{2+2P_a+P_aP_d} \\
v_V &= \frac{1}{2+2P_a+P_aP_d} \\
v_A &= \frac{P_a}{2+2P_a+P_aP_d} \\
v_{MC} &= \frac{P_m}{2+2P_a+P_aP_d} \\
v_{UC} &= \frac{P_a(1-P_m-P_d)}{2+2P_a+P_aP_d} \\
v_D &= \frac{P_aP_d}{2+2P_a+P_aP_d} \\
v_{GD} &= \frac{P_aP_dP_i}{2+2P_a+P_aP_d} \\
v_F &= v_D(1-P_i) = \frac{P_aP_d(1-P_i)}{2+2P_a+P_aP_d}
\end{aligned}
\tag{5}
$$

All steady-state probabilities of DTMC model can be solved by transition probabilities. So the steady state probability of DTMC can be calculated by the transition probability of system. If the value of v_G is greater, the value of v_F is less, the AOMDV routing protocol mostly runs in Good state, the survivability ability of AOMDV routing protocol is more powerful. Numerical analysis method is used to illustrate the relationship between steady-state probabilities and transition probabilities.To be numerical analysis method, the probability of detection and grace degradation can be set to 0.6. To solve the value of steady-state probabilities of DTMC model, transition probability values are set by expert experiences and related references[1,6,12]. They are given as the following:p_a=0.4;p_m= 0.3;p_d = 0.6;p_i = 0.6. By solving equations (4) and (5), steady-state probability values of DTMC model can be computed as: v_G=0.3289, v_V=0.3289, v_A=0.1316, v_{UC}=0.0132, v_{MC}=0.0395, v_D=0.0789, v_{GD}=0.0474, v_F=0.0316.

3.4. SMP-based Survivability Method for Security AOMDV Routing Protocol

A stochastic process is called a Semi-Markov Process if the embedded jump chain is a Markov chain, and the holding times (time between jumps) are random variables with any distribution. From the security researchers' viewpoint, the attacker's behavior and duration time are random, security response is diversified, vulnerability risk is uncertain, all these cause the sojourn time's distribution functions may be non-exponential. According to the definition of SMP(Semi-Markov Process), the survivability stochastic model needs to be formulated by SMP model. There are two types of parameters in SMP survivability model: mean sojourn time and steady-state probability in each state [14]. For computing the survivability measure, the steady-state probabilities $\{\pi_i, i \in X_s\}$of the SMP states should be computed firstly. Therefore π_i can be computed in terms of the embedded DTMC steady-state probabilities v_i and the mean sojourn times h_i [14]:

$$
\begin{aligned}
&\pi_i = \frac{v_ih_i}{\sum_j v_jh_j} \qquad (i,j \in X_s) \\
&\textstyle\sum_i \pi_i = 1
\end{aligned}
\tag{6}
$$

Seen from the formula 6, π_i can be calculated by v_i and h_i , the sum of all steady-state probability π_i is 1, that is $\sum \pi_i = 1$. To calculate the the value of every π_i , $\sum_j v_jh_j$

should be calculated firstly, we use variable H to substitute the long mathematical express.From conjunction with the formula (5), we can get the mathematical express of $\sum_j v_j h_j$, which is shown as formula 7.

$$\begin{aligned}&\sum_j v_j h_j \\ &= \frac{h_G+h_V+p_\alpha h_A+p_{mc}h_{MC}+p_a(1-p_m-p_d)h_{UC}+p_a p_d h_D+p_a p_d p_i h_{GC}+p_a p_d(1-p_i)h_F}{2+2p_a+p_a p_d} \\ &= \frac{H}{2+2p_a+p_a p_d} = v_G H\end{aligned} \tag{7}$$

Every steady π_i state probabilities can be calculated by equation 6 and 7, they are show in formula 8:

$$\begin{aligned}\pi_G &= \frac{v_G h_G}{\sum_j v_j h_j} = \frac{v_G h_G}{v_G H} = \frac{h_G}{H} \\ \pi_V &= \frac{v_V h_V}{\sum_j v_j h_j} = \frac{v_G h_V}{v_G H} = \frac{h_V}{H} \\ \pi_A &= \frac{v_A h_A}{\sum_j v_j h_j} = \frac{v_G p_a h_A}{v_G H} = \frac{p_a h_A}{H} \\ \pi_{MC} &= \frac{v_{MC} h_{MC}}{\sum_j v_j h_j} = \frac{p_m v_G h_{MC}}{v_G H} = \frac{p_m h_{MC}}{H} \\ \pi_{UC} &= \frac{v_{UC} h_{UC}}{\sum_j v_j h_j} = \frac{p_a(1-p_m-p_d)v_G h_{UC}}{v_G H} = \frac{p_a(1-p_m-p_d)h_{UC}}{H} \\ \pi_D &= \frac{v_D h_D}{\sum_j v_j h_j} = \frac{p_a p_d v_G h_D}{v_G H} = \frac{p_a p_d h_D}{H} \\ \pi_{GD} &= \frac{v_{GD} h_{GD}}{\sum_j v_j h_j} = \frac{p_a p_d p_i v_G h_{GD}}{v_G H} = \frac{p_a p_d p_i h_{GD}}{H} \\ \pi_F &= \frac{v_F h_F}{\sum_j v_j h_j} = \frac{p_a p_d(1-p_i)v_G h_F}{v_G H} = \frac{p_a p_d(1-p_i)h_F}{H}\end{aligned} \tag{8}$$

To obtain the sojourn time of every state, performance metric analysis method is used to analyze and acquire the sojourn time of every steady state.

4. SMP-based Survivability Evaluation Approach and Metric

In traditional survivability evaluation method [15], the steady-state availability of this system and Mean Time To Security Failure (MTTSF) are used to evaluate the survivability ability of system. While the evaluation metric can not reflect the whole system states' change and balance relation among different states. When attack occurs, the balance point among different states may be moved. How to measure the systematic change and balance point moving is a great scientific problem.

In information theory, entropy is a measure of the uncertainty associated with a random variable. The probability distribution of the events, coupled with the information amount of every event, forms a random variable whose expected value is the average amount of information. Entropy is calculated by the probability distribution. It is one of the evaluation functions for quantifying the diversity, uncertainty or randomness of a system. In wireless sensor network, the attack is random, the response to attack may be diversified, the routing protocol may stay in different state and keep different time. The systematic survivability of routing protocol in wireless sensor network is uncertain.We

propose to use the information entropy method to quantitatively describe and explain the systematic change caused by external attack and internal security techniques. To describe the systematic survivability of wireless sensor network routing protocol, survivability entropy is proposed to quantitatively evaluate the systematic change and balance point moving condition.In information entropy theory, discrete information sources refer to that of discrete distribution on time and amplitude. The finite states in DTMC model and SMP model can be treated as discrete information sources. The steady state probabilities distribution of DTMC and SMP model can be treated as probabilities distribution of discrete information source.

As we know, the sum of discrete information source probability in information theory should be 1. As the sum of steady-state probabilities in DTMC and SMP model is 1, every state in SMP model can be treated as every discrete information source in Shannon information theory. According to the definition of information entropy, the steady-state probability distribution of SMP model can be treated as the probability distribution of distrete information source in information theory. According the definition of entropy, the mathematical formula of survivability entropy is shown as:

$$H(X) = E\left(\log \frac{1}{p\left(a_i\right)}\right) = -\sum_{1}^{n} p\left(a_i\right) \log p\left(a_i\right) \tag{9}$$

In the formula 9 , the number of states in SMP model is n, steady state probability of state i is $p\left(a_i\right)$. Seen from the formula 6, we know that $\sum_i \pi_i = 1$, they meet the condition of information source probability distribution in Shannon information theory. In SMP model, the survivability entropy can be expressAed as:

$$H(X) = E\left(\log \frac{1}{p\left(a_i\right)}\right) = -\sum_{1}^{n} p\left(a_i\right) \log p\left(a_i\right) = -\sum_{1}^{n} \pi_i \log \pi_i, i \in X_s \tag{10}$$

In formula 10, π_i stands for steady state probability of state i, X_s stands for the set of all possible states. $H(X)$ stands for system survivability entropy in SMP model. $H(X)$ reflects the average uncertain degree of system survivability in SMP model. Under attack or security conditions, attack or security conditions related state probabilities will change, at the same time, the sojourn time in related states will change, so the $H(X)$ will change with different attack or security conditions. We use condition entropy to express the change under condition y, it is shown in equation 11.

$$H(X|y) = E\left(\log \frac{1}{p\left(a_i|y\right)}\right) = -\sum_{1}^{n} p\left(a_i|y\right) \log p\left(a_i|y\right) \tag{11}$$

To describe the systematic change caused by the attack and security techniques, we use entropy difference to express the change quantitatively, it is shown in formula 12:

$$\Delta H(X) = H(X) - H(X|y) \tag{12}$$

Entropy difference $DeltaH(X)$ is be used to quantitatively describe the systematic change caused by attack and security techniques. $DeltaH(X)$ can be used to express the

systematic change by external attack and internal security technique, the entropy difference reflects the systematic changes and tradeoff between attack and security. Traditional performance metric can not describe the systematic change quantitatively.

5. Simulation Experiment and Survivability Evaluation

5.1. Simulation Experiment and Network Scenario

To precisely evaluate the survivability in wireless sensor network, we use simulation experiments to analyze and calculate the survivability entropy metric in SMP model. NS2 (network simulator 2) is a common software tool to do network simulation research. It is used to simulate the attack and record network simulation data. In SMP model, the survivability is related to mean sojourn time and steady-state probabilities in DTMC model. As the steady-state probabilities in DTMC model are calculated by the transition probabilities in DTMC, they are computed by numerical analysis method. The mean sojourn time can be obtained by simulation experiments in this paper. There are eight states in the SMP-based survivability model of wireless sensor network routing protocol. To recognize the eight states and compute the sojourn time, performance metrics analysis method are adopted.

If the attack is serious, the performance metrics will be changed significantly, the states can be recognized by the performance metrics, multi-metrics should be used to distinguish different states. Network performance metrics are used to detect the routing-level DoS attack and recognize the different states of SMP survivability model in wireless sensor network. Network Simulator 2 (NS2) is adopted to simulate the low-rate DoS attack behaviour, all the packets in network are recorded in the Trace file of NS2. Network performance metric and routing packets statistic can be obtained in the simulation trace file. Network scenario and attack model are described in the following, the network topology is shown in figure 3.

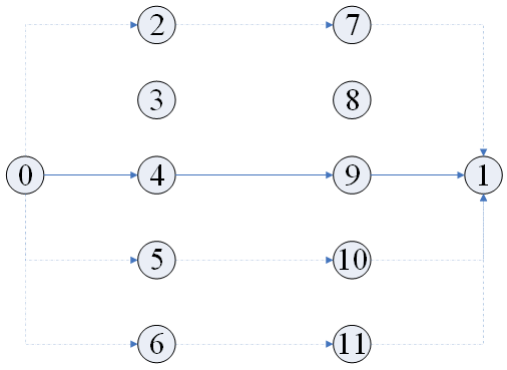


Fig. 3. Network simulation topology

The simulation configuration is shown in table 3,

Table 3. Network Simulation Configuration

Parameter type	value
Moving area	1000m * 1000m
Simulation time	200s
Number of nodes	12
Source node ID	0
Destination node ID	1
Attack node ID	5
Transmission protocol	TCP
Routing protocol	AOMDV
Application protocol	FTP
MAC layer protocol	IEEE 802.11
Attack type	Random RREQ flood attack
Attack duration	30-40s 60-100s
DoS Attack interval	0.03-0.08 s

To reflect the character of AOMDV multi-paths routing protocol, three different route paths are simulated in the simulation experiments; to embody the character of multi-hops routing protocol, there are three hops in the route path in the simulation scenario. So there are twelve nodes in wireless network simulation scenario in the paper,which is shown is figure 3. Node 0 is source node, node 1 is destination node, there are four routing paths between node 0 and node 1,they are 0-2-7-1,0-4-9-1,0-5-10-1 and 0-6-11-1.Node 5 is attack node. It sends a amount of Routing Request (RREQ) messages to other nodes to flood and attack the network connection. To simulate different flood attack severity, the number and interval of RREQ flooding packet are changeable in our experiments.

(1) Detection method

As the limited computing and storage resource of wireless sensor node, all network traffic is recorded in network trace file,performance metrics-based analysis method is used to detect the RREQ message flood attack. Reasonable performance metric thresholds are set to detect and block the attack. So routing protocol will enter different states under different attack strength and security mechanism.

(2) State recognize and separation

There are eight states in our survivability model. To recognize the different states in the model,three kind of performance metrics are used to separate the different states.They are packet loss ratio, network throughput, network delay. We use the three performance metrics to recognize the different states and calculate the sojourn time in every state.

(3) Mean Sojourn time calculation

Numerical analysis method can be used to analyze and calculate survivability parameters in traditional SMP model. However, numerical analysis method can not reflect the real attack and security status, so experimental method is used to calculate mean sojourn time of every state, numerical analysis method is used to compute the steady-state probabilities in SMP model. After all state in SMP model are recognized and separated in network

simulation trace file, the sojourn time in every state can be easily calculated. Simulation experiment method is more accurate than traditional numerical analysis method.

5.2. State Separation by Performance Metric

NS2 can be used to simulate many network protocols over wired and wireless networks.It is used to simulate the AOMDV routing protocol and DoS attack in this paper. The following assumes that each state is recognized and separated by performance metrics and attack indicators, the metrics and indicators are deduced from simulation experiments and grade analysis. The goal of state separation by performance metrics is obtain the mean sojourn time, it is necessary parameter to the SMP model; the steady-state probabilities of SMP model are given by expert experiences and related references; so we use a hybrid approach to calculate the survivability ability of WSN routing protocol.

(1) State V

State V stands for vulnerability state,in which attacker try to do some probing and scanning,so network performances have been affected partially. Before DoS flood attack, the attack node try to inject some redundant route messages to network to probe the vulnerability of network routing protocol, so network performance can be affected locally. If the packet loss ratio has an increases less than 5%, the network throughput has a decreases less than 10%, and network delay has a increase less than 10%,the current state is considered to enter into state V.

(2) State A

State A stands for attack state, in which attacker has started a DoS attack to the routing protocol of wireless sensor network, so network performances have been affected significantly. If the packet loss ratio has an increases more than 5% and the network throughput has a decreases more than 10%,and network delay has an increase more than 10%, the current state is considered to enter into state A.

(3) State MC

State MC stands for mask compromised state, in which AOMDV routing protocol can mask some DoS attack by switching the attacked route path to a backup route path. By analyzing the network simulation trace file, the packet loss ratio increased more than 5%,at this time, routing path was switched to a backup path, that results in the packet loss ratio decreased near to normal value, the current state is considered to enter into state MC.

(4) State D

State D stands for detection state,in which the DoS attack can be detected by the intrusion detection algorithm. By analyzing the network simulation trace file, after the packet loss ratio increased more than 5%, routing path was not switched to a backup path, however,the indicator of intrusion detection is set to true, the attack nodes were detected,then the current state is considered to enter into state D.

(5) State UC

State UC stands for undetected compromised state, in which the DoS attack can not be masked and detected. By analyzing the network simulation trace file, after the packet

loss ratio increased more than 5%, routing path was not switched to a backup path, the indicator of intrusion detection was set to false, the network throughput has a decreases more than 15%,and network delay has an increase more than 15%, the current state is considered to enter into state UC.

(6) State GD

State GD stands for graceful degradation state, in which the network system only maintains essential services, some wireless nodes are isolated to not join in the network. By analyzing the network simulation trace file, after the packet loss ratio has an increase more than 5%, routing path was not switched to a backup path, the indicator of intrusion detection is set to true, the attack nodes were detected and isolated to not to join in the wireless sensor network, then the packet loss ratio decreased to near normal value, the current state is considered to enter into state GD.

(7) State F

State F stands for failed state, in which the network system can not provide essential services,the DoS attack can not be controlled efficiently. If the packet loss ratio has an increases more than 10% and the network throughput has a decreases more than 20%,and network delay has an increase more than 20%, at the same time, no backup path can be used, the indicator of intrusion detection is set to true, however,the attack node can not be identified and isolated efficiently, the current state is considered to enter into state failed.The routing protocol need be recovered to normal state manually.

(8) State G

State G stands for good state, in which the network system works well,there are not any attack or probing behaviour in the network. All the network performance metrics and attack indicators show well.

So each state can be recognized by the performance indicators and related parameters, mean sojourn time can be calculated by the performance metrics and security indicators analysis from NS2 network simulation trace files.

5.3. Steady-state Probability and Survivability Precise Calculation

Numerical analysis method is used to show the relationship between steady-state probabilities and transition probabilities in DTMC model and SMP model. DoS (Denial of Service) attack is a kind of representative attack mode in wireless sensor network. We use DoS attack and intrusion detection technique to illustrate how to calculate the steady-state probability and survivability entropy in SMP model. As the DoS attack is a kind of representative attacks, according to the references [4,6,15] and our research experiences, the probability of detection P_d is set to 0.5,the probability of grace degradation P_i is set to 0.6, the probability from vulnerability to attack P_a is set to 0.4, the probability of masking attack is set to 0.3. According to the formula 5, steady-state probability in DTMC model can be calculated. They are calculated as the following: v_G=0.3333, v_V=0.3333, v_A=0.1333, v_{UC}=0.0267, v_{MC}=0.04, v_D=0.0667, v_{GD}=0.04, v_F=0.0267.

According to the formula 678, to solve the value of steady-state probabilities of SMP model, mean sojourn time should be calculated by the definition of every state.In our simulation experiment, the whole simulation time is 200s, the mean sojourn in Good state

is set to 15s. Mean sojourn time can be acquired by simulation experiment and trace file analysis, then steady-state probability of SMP model can be calculated according to formula 6. As the intensity and interval of DoS flooding attack are changeable in some range, only some states appear in the experiments. We do three times of simulation experiments by NS2 and calculate the mean sojourn time of every state. The mean sojourn time is calculated,mean sojourn time h under different Attack Time Interval(ATI) are shown in table 4.

Table 4. Mean sojourn time h under different attack time interval

Mean sojourn time (s)	ATI=0.08	ATI=0.06	ATI=0.04	ATI=0.03
h_v	4.5	7.5	9	10.5
h_a	7.5	10	11	11.5
h_d	15	14	10.5	9.5
h_{gd}	20	14.5	15	10
h_g	1	15	15	15

Seen from the table 4, with the increase of DoS attack time interval, the mean sojourn time in vulnerability state to attack state are increased. Based on the mean sojourn time from simulation experiments, steady state probability of SMP model can be calculated by formula 6., they are listed in table 5.

Table 5. Steady state probability under different attack time interval

Mean sojourn time (s)	ATI=0.08	ATI=0.06	ATI=0.04	ATI=0.03
h_v	0.1613	0.2416	0.2786	0.3163
h_a	0.1075	0.1289	0.1362	0.1386
h_d	0.1075	0.0902	0.0650	0.0572
h_{gd}	0.0861	0.0561	0.0557	0.0361
h_g	0.5376	0.4832	0.4644	0.4518

At the same time, survivability Entropy (SE) under different attack time interval can be calculated by formula 10, they are listed in table 6.

Seen from the table 5, with the decrease of DoS attack interval,the strength of DoS flood attack increases. The steady state probability π_G, π_{GD} and π_D are decreased, while the steady state probability π_V and π_A are increased; the survivability entropy is increased firstly, then decreased. The experiment data shows that DoS attack can influence the network performance because of the different attack strength,thus the mean sojourn time of every state is affected, the WSN survivability will be changed as the change of every state. When the attack strength reaches some value, intrusion detection and multi-pathes masking will be triggered to improve the network performance and survivability, the attack is effectively suppressed by security mechanism,such as intrusion detection and muti-pathes

Table 6. Survivability Entropy under different attack time interval

SE	ATI
1.9021	0.08
1.9293	0.06
1.9077	0.04
1.8474	0.03

masking. Survivability entropy metric based on SMP model can reflect the survivability ability of the routing protocol in wireless sensor network, at the same time,the survivability entropy can reflect the dynamic change and trade-off between attack and security mechanism.However,traditional survivability evaluation metric can not reflect the systemic change

6. Comparison with Existing Methods

To explain the advantages of our approach, different items are used to compare our approach to existing methods,the comparison is shown in table7.

Table 7. Comparison our approach with existing methods

Comparison Items	Numerical calculus	Experimental calculus	Systemic survivability evaluation metric	computing complex
B. Madan[15]	support	none	None	medium
KIM et cl[6]	support	none	None	medium
Fei Xing et cl[4]	support	support	Network topology connectivity	high
Our approach	support	support	Network Survivability Entropy	medium

As shown in table 7, reference [6] and reference[15] only support the numerical parameter calculation, Mean Sojourn Time h dose not comes from experimental calculation,so the survivability calculus depends on the expert knowledge. Although the systemic survivability evaluation metric is supported in reference [4], the network topology connectivity information is difficult to be obtained in WSN environment, and the computing complex of survivability calculus depends on the connectivity probability problem of a geometric random graph, so the the computing complex in reference [4] is high.

However, Numerical Parameter calculus,Experimental Parameter calculus, systematic survivability evaluation metric are all supported in our approach, in addition to the analysis methods, a novel survivability evaluation metric-network survivability entropy is firstly proposed to describe the systematic survivability change of WSN routing protocol, at the same time, the computing complex of survivability calculus is medium, because our survivability calculus in formula 10 depends on the steady state probability of every state. Traditional surivivability evaluation approach is based on network topology structure, our approach is based on system-level states and changes. As the network topology structure

is difficult to be obtained, our approach is more suitable to the WSN application environment. So our approach performs better than other existing methods.

7. Conclusion

Survivability evaluation is an important challenge in wireless sensor network security research.Because of the special characteristic of wireless sensor network,existing method is difficult to precisely calculate the survivability ability of routing protocol in wireless sensor network. A novel survivability entropy evaluation method is proposed to precisely calculate the survivability ability under DoS flood attack in wireless sensor network.

There are two main contributions in our research work:

1. Systematic survivability ability is firstly proposed and described by survivability entropy metric, which can describe the systematic survivability change of WSN routing protocol, it is validated by NS2 network simulation experiments.
2. Numerical analysis and simulation experiment methods are firstly combined to precisely calculate the survivability entropy metric,especially the sojourn time of every state was calculated by experiments. Experimental results show that the novel survivability entropy evaluation method is scientific and effective to precisely calculate the survivability ability of routing protocol in wireless sensor network.

Acknowledgments This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0803403, Grant 2018YFB0803400, and Grant 2018YFB0505302, in part by the National Natural Science Foundation of China under Grant 61832012, in part by the National Social Science Fund of China under Grant 18BGJ071, and in part by the Fundamental Research Funds for the Central Universities under Grant TW201706.

References

1. R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, and N. Mead. Survival Network Systems: An Emerging Discipline. Technical Report CMU/SEI-97-TR-013, SEI, CMU, http://www.cert.org/research/97tr013.pdf, 1997
2. Telecom Glossary 2000. Technical Report ANS T1.523-2001, Institute for Telecomm. Services, NTIA, DOC, http://www.atis.org/tg2k/, Feb. 2001,3
3. D. Chen, S. Garg, and K.S. Trivedi. Network Survivability Performance Evaluation: A Quantitative Approach with Applications in Wireless Ad-hoc Networks. Proc. ACM Int'l Workshop Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '02), 2002: 61-68
4. Xing Fei, Wang Wenye. On the Survivability of Wireless Ad Hoc Networks with Node Misbehaviors and Failures. IEEE Transactions on Dependable & Secure Computing, 2010, 7(3):284-299
5. WANG Haitao, et al. Survivability evaluation index systems and evaluation models for Wireless Sensor Networks. IEEE 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD):2203-2207
6. KIM, Dong Seong, SHAZZAD, Khaja Mohammad, PARK, Jong Sou. A framework of survivability model for wireless sensor network. IEEE 2006 The First International Conference on Availability, Reliability and Security:514-522

7. C Chang, C Zhu, H Wang. Survivability Evaluation of Cluster-Based Wireless Sensor Network under DoS Attack. Internet of Things. 2012:126-132
8. Hongsong Chen, et al. Design and performance evaluation of a multi-agent-based dynamic lifetime security scheme for AODV routing protocol. Journal of Network and Computer Applications 30.1 (2007): 145-166
9. CHEN H S, HAN Z, DENG SN. Analysis and Research on Big Data Security in Smart City. Netinfo Security. 2015, (7):1-6
10. Goncalo Martins, Sajal Bhatia, Xenofon Koutsoukos, Keith Stouffer, Cheeyee Tang, Richard Candell. Towards a systematic threat modeling approach for cyber-physical systems. Resilience Week (RWS) 2015, 1-6
11. LV, Haitao, et al. Protection Intensity Evaluation for a Security System Based on Entropy Theory. Entropy, 2013, 15.7: 2766-2787
12. Azni A H, Ahmad R, Noh Z A M, et al. Systematic Review for Network Survivability Analysis in MANETS. Procedia - Social and Behavioral Sciences, 2015, 195: 1872-1881
13. Fu, Bai-Bai, et al. Survivability of public transit network based on network structure entropy. International Journal of Modern Physics C 26.09 (2015): 1550104
14. K. S. Trivedi. Probability and Statistics with Reliability, Queuing, and Computer Science Applications (2nd ed.). John Wiley & Sons, 2001
15. Bharat B. Madan, Katerina Goševa-Popstojanova, Kalyanaraman Vaidyanathan, Kishor S. Trivedi, A method for modeling and quantifying the security attributes of intrusion tolerant systems, Performance Evaluation. 2004,(56):167-186
16. Marina M K, Das S R. Ad hoc On-demand Multipath Distance Vector Routing. Network Protocols Ninth International Conference on ICNP. IEEE Xplore, 2001:14-23
17. Dagdeviren O, Akram V K, Tavli B. Design and Evaluation of Algorithms for Energy Efficient and Complete Determination of Critical Nodes for Wireless Sensor Network Reliability. IEEE Transactions on Reliability, 2018.
18. X. Gao, K. Li and B. Chen. Invulnerability Measure of a Military Heterogeneous Network Based on Network Structure Entropy. IEEE Access, 2018 (6):6700-6708
19. Aaron Zimba, Hongsong Chen, Zhaoshun Wang. Bayesian network based weighted APT attack paths modeling in cloud computing. Future Generation Computer Systems. 2019, 96(7):525-537
20. Hongsong Chen, Caixia Meng, Zhiguang Shan,Zhongchuan Fu and B. K. Bhargava. A novel Low-rate Denial of Service attack detection approach in ZigBee wireless sensor network by combining Hilbert-Huang Transformation and Trust Evaluation. IEEE Access. 2019 vol. 7, pp. 32853-32866
21. Hongsong Chen, Ming Liu, Zhongchuan Fu. Using Improved Hilbert-Huang Transformation Method to Detect Routing-Layer Reduce of Quality Attack in Wireless Sensor Network. Wireless Personal Communications. 2019, 104(2): 595-615

Hongsong Chen (corresponding author) received the Ph.D. degree in Department of Computer Science from Harbin Institute of Technology, China, in 2006. He was a visiting scholar in Purdue University from 2013 to 2014. He is current a professor in Department of Computer Science,University of Science and Technology Beijing, China. His current research interests include wireless network security, attack and detection model, cloud computing security.

Haiyan Zhuang was born in May 1976, Female, Associate Professor of Railway Police College, Zhengzhou, China. Her research interested include information security, data analysis and mining, trust evaluation and computing.

Zhiguang Shan was born in 1974, PhD, professor and the director of Informatization and Industry Development Department, State Information Center of China. He also serves as the director of China Smarter City Development and Research Center. He received the B.A. degree in automation engineering, and the Ph.D. degree in computer science, both from the University of Science and Technology Beijing, Beijing, China, in 1997 and 2002, respectively. His main research interests include computer networks, performance evaluation, strategic planning and top design of smarter city, macro planning and developing policies of informatization. He has co-authored more than 70 papers in research journals and conference proceedings and 9 books in these areas.

Chao-Hsien Lee is Associate Professor,Department of Electronic Engineering, National Taipei University of Technology, Taipei City, Taiwan. His research area are mobile and information security, trust computing and Internet of Things.

Zhongchuan Fu received the Ph.D. degree in Department of Computer Science from Harbin Institute of Technology, China, in 2006. He is current an associate professor in Department of Computer Science,Harbin Institute of Technology. His research interested include computer system security, Fault tolerant computing, trust computing.

Received: September 18, 2018; Accepted: June 30, 2019.

Computer Science and Information Systems 16(3):753–772 https://doi.org/10.2298/CSIS180930025H

A Framework for Fog-assisted Healthcare Monitoring

Jianqiang Hu[1,2], Wei Liang[2], Zhiyong Zeng[3], Yong Xie[1,2], and Jianxun Eileen Yang[4]

[1] School of Computer and Information Engineering,
Xiamen University of Technology,
361024 Xiamen, P.R. China
hujianqiang@tsinghua.org.cn
[2] Key Laboratory of Internet-of-Things Applications of Fujian Province,
Xiamen University of Technology,
361024 Xiamen, P.R. China
{wliang,yongxie}@xmut.edu.cn
[3] College of Mathematics and Informatics,
Fujian Normal University,
350007 Fuzhou, P.R. China
zzyong@fjnu.edu.cn
[4] Shenzhen Research Institute of Sun Yat-Sen University,
Shenzhen 518057, P.R. China
eileenyjx@163.com

Abstract. In order to tackle some challenges in ubiquitous healthcare monitoring such as mobility, scalability, and network latency, a framework for Fog-assisted healthcare monitoring is proposed in this paper. This framework is composite of body-sensing layer, Fog layer (Fog-assisted gateway), Cloud layer (health Cloud). And then, this paper makes an intensive study in some key technologies of the proposed framework such as an IPv6-based network architecture, intelligent warning model based on subband energy feature, security framework of HL7 RIM-based data exchange, health risk assessment based on fusion of grey model and Markov model. Finally, results of experiment depict that the proposed intelligent warning model can make immediate distinction abnormal signals. Moreover, the proposed health assessment model confirms its effectiveness with respect to 245 patients in Xiamen District of Jimei.

Keywords: Healthcare monitoring, Fog computing, IPv6, HL7 RIM, Health risk assessment.

1. Introduction

Population aging is happening faster than ever before. A report issued by the World Health Organization (WHO) stated that there was a shortage of about 7.2 million healthcare workers in 2013, and this is estimated to reach 12.9 million by 2035. The increase in population and chronic diseases as a result there is increasing pressure on quality and quantity of healthcare. According to the Chinese Cardiovascular Health Index (2017), the mortality of cardiovascular and cerebrovascular diseases is on the rise. The number of patients with vascular disease was 290 million, including 13 million of stroke, 11 million coronary heart disease, 5 million pulmonary primary heart disease, 2 million 500 thousand rheumatic heart disease and 2 million congenital heart disease. The prevention

and monitoring of cardiovascular disease (CVD) is very important, especially for sudden heart disease. If it is possible to monitor the subtle signs and take effective measures in advance, the patients of 70%-80% can avoid death. Healthcare monitoring considered as an important means to reduce the cost of medical treatment, alleviate the shortage of medical resources and improve the overall level of medical treatment.

Recent technological trends such as WBAN (Wireless Body Sensor Network), Cloud computing and Bigdata provide an infrastructure for ubiquitous healthcare monitoring that can prevent cardiovascular disease (CVD) and respond to the occurrence of disease. These systems not only allow the elderly to live independently for longer but they also have potential to make e-health services more sustainable by reducing the pressure placed on the overall health clinics by patients. The general architecture of ubiquitous healthcare monitoring includes three main components: (i) WBAN, (ii) Internet-connected gateways, and (iii) Cloud and Bigdata support. Sensors and devices of WBAN provide real-time physiological information related to the health condition of the monitored subject. Interconnected gateways act as a hub between WBAN and Cloud services, which are responsible for the coordination between heterogeneous sensors and their connection to WAN. Cloud is responsible for Bigdata processing and analytics of physiological data and further personalized healthcare and treatments. However, the architecture faces following challenges:

(i) Sensors and devices of WBAN generate a huge amount of data and it is difficult for Cloud system to process it in real-time due to communication overhead. Cloud computing is not able to provide low latency, location awareness and high quality of service for real time applications. In pratice, healthcare applications often require expeditious analysis of health data and immediate decision. The delay of data transfer and processing over Cloud is unacceptable.

(ii) The gateway is proficient to maintain reliable and secure network connectivity between Sensors and Cloud. However, some advantageous services that can be potentially offered by a smart gateway will be limited if the gateway is deployed in a standalone and independent fashion. Smart gateway should provide more intelligence at the edge of the network and facilitates the interplay between WBAN and Cloud system.

In Healthcare a little delay can cost a patient's life therefore, to enhance services and applications, Fog Computing has came in to picture. Fog computing enables the architecture to support low-latency response, efficient scalability, location awareness, and developing applications offered by gateways. By utilizing geographically distributing Internet-connected gateways, a Fog-assisted intermediary layer between WBAN and Cloud can be formed to provide efficient healthcare services. The objectives of our paper include: (i) proposing a framework for Fog-assisted healthcare monitoring; (ii) enabling underlying network to provide mobility of patients with different protocols; (iii) giving intelligent warning model based on subband energy feature; (iv) proposing security framework of HL7 RIM-based data exchange; (v) providing health risk assessment on fusion of grey model and Markov model.

The remainder of the paper is organized as follows: Section 2 gives a brief overview of related works. Section 3 presents a framework for Fog-assisted health monitoring. Section 4 discusses some key technologies of proposed framework. The experiments for showing effectiveness of proposed frameworks are setup in Section 5. Section 6 concludes the paper.

2. Related Works

2.1. Cloud Computing in Healthcare System

In Cloud Compting domain, in 2014, Hua-Pei Chiang et al. [5] proposed a green cloud-assisted healthcare service on WBAN, and considered the sensing frequency of the physiological signals of various body parts, as well as the data transmission among the sensor nodes of WBAN. Transmission in WBSN is coordinated according to the number of sensor nodes worn by each user and the detection frequency of the various sensor nodes; personal physiological signals are regularly and efficiently transmitted to the cloud network for processing. PhysioDroid [3] used a wearable chest belt with sensors for ECG, heart and respiration rates, skin temperature, and body motion. It shows the severity of health vital signs using different colors and generates an emergency call according to them. In 2015, Shu-Lin Wang et al. [23] proposes a framework which integrates Cloud Computing Wireless Communication, and Wireless Sensor Networks technology, and applies a Collaborative Filtering (CF) technique to develop a Mobile Health Information Recommendation service to help users to obtain their preferred health information more efficiently. In 2016, Rasha Talal Hameed et al. [10] developed health monitoring system based on wearable sensors and cloud platform. The sensors measure various parameters, such as a glucometer, airflow and patient position which are transmitted via microcontroller by a gateway to a cloud storage platform. In 2018, Prabal Verma el al. [22] proposed cloud-centric IoT based disease diagnosis healthcare framework consists of three phases. In phase1, users' health data is acquired from medical devices and sensors. The acquired data is relayed to cloud subsystem using a gateway or local processing unit (LPU). In phase 2, the medical measurements are utilized by medical diagnosis system to make a cognitive decision related to personal health. In phase 3, an alert is generated to the parents or caretakers in context of person's health.

2.2. Fog Computing in Healthcare System

In 2016, C. S. Nandyala et al. [17] proposed architecture for IoT based u-healthcare monitoring with the motivation and advantages of Cloud to Fog(C2F) computing which interacts more by serving closer to the edge (end points) at smart Homes and Hospitals. M. Ahmad et al. [1] proposed a framework of Health Fog where Fog computing is used as an intermediary layer between the cloud and end users. The design feature of Health Fog successfully reduces the extra communication cost that is usually found high in similar systems. In 2018, B. Negash et al. [18] focuses on a smart e-health gateway implementation for use in the Fog computing layer, connecting a network of such gateways, both in home and in hospital use. Home-based and in hospital patients can be continuously monitored with wearable and implantable sensors and actuators. A. M. Rahmani et. al [19] proposed to exploit the concept of Fog Computing in Healthcare IoT and the strategic position of such gateways at the edge of the network to offer several higher-level services such as local storage, real-time local data processing, embedded data mining, etc., presenting thus a Smart e-Health Gateway. Sandeep K. Sood [21] designed a Fog assisted cloud-based healthcare system to diagnose and prevent the outbreak of chikungunya virus. The state of chikungunya virus outbreak is determined by temporal network analysis at cloud layer using proximity data.

In summary, Fog computing is a paradigm extending Cloud computing and services to the edge of the network, and thus to reduce the latency of decision making. Local decision making not only reduces the latency but also network traffic resulting in a much more energy-efficient system than cloud-assisted solutions. Many researchers have proposed Fog-assisted healthcare monitoring systems but have not laid emphasis on location awareness, intelligent warning for real time applications (e.g., heart disease, cerebral infarction), health risk assessment for forecasting health conditions (e.g., heart attacks) before they occur.

3. Proposed Framework

In this paper, we proposed a framework for Fog-assisted healthcare monitoring, which is composite of body-sensing layer, Fog layer(Fog-assisted gateway) and Cloud layer (Health Cloud). In body-sensing layer layer, wearable and implantable physiological sensors of body-sensing layer generate physiological data. Fog-assisted gateway provides protocol conversion, data preprocessing and local analytics and services, located at the network edge. Health cloud implements data warehouse, bigdata analysis and provides health services. A Framework of Fog-assisted health monitoring is shown as Fig. 1.

3.1. Body-sensing Layer

Body-sensing layer is composed of a series of intelligent physiological sensors, including fingertip oxygen sensors, blood-glucose sensors, ECG sensors, implantable sensors of blood pressure, to measure some basic physical vital information of the patients, like temperature, blood pressure, blood sugar, pulse rate, heart condition, respiration etc. Each sensor is equipped with the physiological signal conditioning circuits, a microcontroller, and a short distance protocol interface. According to demands of a patient's disease, these sensors can be selectively configured to monitor the respective physiciological signal. Consequently, they provide a continuous flow of physiological information related to real-time health conditions of the monitored subject.The physiological information initially processed by these sensor nodes is then transmitted to the gateway via wireless or wired communication protocols such as Bluetooth, Wi-Fi, ZigBee or 6LoWPAN.

3.2. Fog-assisted Gateway

Physiological sensors select the nearest Fog-assisted gateway to send physiological data. Multiple geographically distributed Fog-assisted gateway forms the fog. Each Fog-assisted gateway supports different communication protocols, acts as a dynamic touching point between WBAN and the local switch/Internet. It receives data from different subnetworks, performs protocol conversion, and provides other higher-level services such as data preprocessing, local analysis and services, including intelligent alarm, on-line real-time monitoring, and notification service. And then, selected data is sent to Health Cloud for further analysis based on security framework of HL7 RIM-based data exchange. According to framework of Fog-assisted health monitoring, Fog-assisted gateway requires to continuously handle a large amount of sensory data in a short time and response appropriately with respect to various conditions. Consequently, Fog-assisted gateway also enhances location awareness and high quality of service for real time applications.

3.3. Health Cloud

Health Cloud holds health monitoring archives and electronic medical records in Cloud data center. Combined with patient's physiological monitoring data, Health Cloud provides intelligent classification and risk assessment of chronic disease, which helps the individuals to have a comprehensive understanding of health conditions. Intelligent classification based on cascaded deep learning helps individuals to predict the potential disease with its level of severity [12]. Health risk assessment model based on fusion of grey model and Markov model is applied to predict relative risk and absolute risk of individual's health status. Health Cloud provides some healthcare services and health intervention with doctors, nutritionists and other medical team.

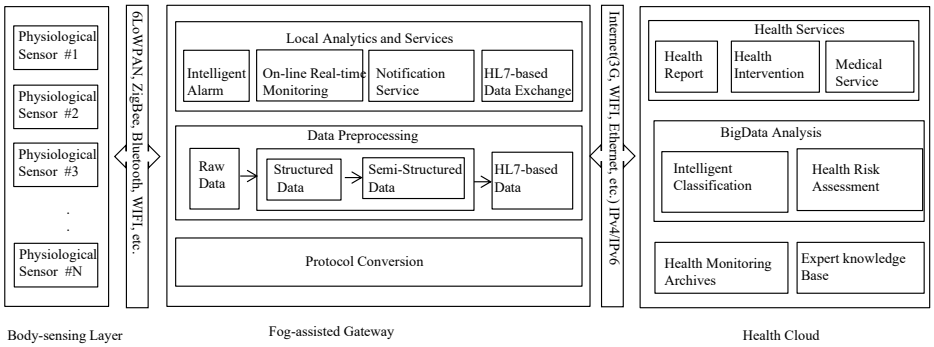


Fig. 1. A framework for Fog-assisted health monitoring

4. Some Key Technologies of Proposed Framework

4.1. Protocol Conversion for Enhancing Sensor Node Mobility

As shown in Fig. 1, intelligent physiological sensors are connected to Fog-assisted gateway using different standards (e.g., ZigBee, 6LoWPAN, Bluetooth, Wi-Fi). Thus, Fog-assisted gateway plays very important role in providing interoperability for the various sensors connected via distinct network interfaces, enabling them to exchange information wand work seamlessly. For example, ECG sensor integrates ECG sensor module, data conversion module, power module, storage unit, wireless transceiver and other major functional modules. ECG sensors are responsible for collecting ECG parameters. The wireless transceiver is responsible for the communication between sensor node and Fog-assisted gateways. IPv6 (Internet Protocol Version 6) is one of the most important connectivity of the IoT, as it is not possible to add billions of devices to the IPv4 (Internet Protocol Version 4) Internet. IPv6 is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks. In this approach each device on the network has a unique address globally reachable directly from any other location on the Internet. Therefore, protocol conversion is responsibile for

uniformed mapping from different standards (e.g., ZigBee, 6LoWPAN, Bluetooth, Wi-Fi) to IPv6.

Protocol conversion method for sensor network access Fog-assisted gateway comprises the following step: (i) initializing a gateway: respectively establishing a mapping relationship between a sensor address and the gateway. For example, ECG sensors have unique IPv6 addresses, which are configured automatically by global routing prefix, subnet ID and interface ID. Each sensor has EUI-48 bit Bluetooth device address, so the interface ID can be obtained by the IEEE EUI-64 translation mechanism. To convert an EUI-48 Bluetooth device address into an EUI-64, the interface appends the two octets FF-FE and then copy the organization-specified extension identifier. The interface ID plus the routing prefix FE80::/64 and automatically configures the 128 bit local link address. (ii) simultaneously starting two processes: receiving and analyzing periodic data packets of the sensor network by using the sensor network monitoring process, and saving the periodic data packets in a local memory according to the mapping relationship; updating a corresponding memory by using the received new sensing data. The method finishes communication protocol conversion at the gateway.

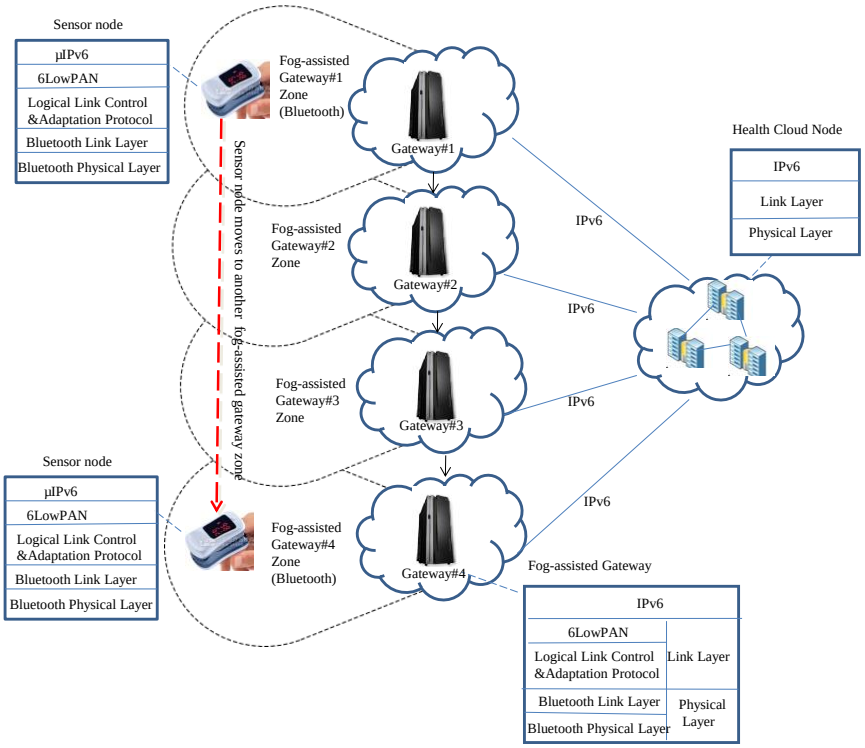


Fig. 2. An IPv6-based network architecture

Based on protocol conversion method, we proposed an IPv6-based network architecture (see Fig. 2). Some characteristics of this architecture are as follows:

(i) The protocol stack of sensor node from bottom to top includes: Bluetooth Physical Layer, Bluetooth Link Layer, Logical Linking Control and Adaptation Protocol, 6LowPAN, μIPv6.

(ii) The protocol stack of Fog-assisted gateway from bottom to top includes: Bluetooth Physical Layer, Bluetooth Link Layer, Logical Link Control and Adaptation Protocol, 6LowPAN and IPv6. It also supports Physical layer, Link layer, and IPv6 in dual protocol stack.

(iii) The protocol stack of Health Cloud from bottom to top includes: physical layer, link layer, and IPv6.

Based on protocol conversion and IPv6-based network architecture, Fog-assisted gateways are used to support sensor nodes during mobility from one geographic location to another domain. Mobile IPv6 [13] allows sensor nodes to move within the Internet topology while maintaining reachability and ongoing connections between mobile and correspondent nodes. In practice, each gateway utilizes discovery and mobility support module to provide uninterrupted service for sensor node. For example, as sensor node moves from Fog-assisted gateway#1 zone to Fog-assisted gateway#4 zone, it receives a broadcast message from the Fog-assisted gateways regarding its identity. When sensor node receives broadcast message, it replies with a discovery request to the respective gateway, which is processed by the device discovery and mobility support module in the gateway. Each sensor is always identified by its home address, regardless of it is situated away from its home geographic location.

4.2. On-line Real-time Monitoring and Processing

As a key feature of Fog computing, one-line real-time monitoring and local data processing are implemented to provide intelligence at the Fog-assisted gateway, which requires to continuously handle a large amount of sensory data in a short time and response appropriately with respect to various conditions. Physiological signal analysis plays a significant role in the on-line real-time monitoring and processing. Takes ECG monitoring for example, ECG signal is the recording of the electrical activity of the heart which provides the clinical information about the condition of heart. The signal is characterized by electrical activity during a cardiac cycle named as QRS complexes, P and T waves. Detection of QRS complex and R-peak is one of the most important parts of the ECG signal analysis. Till now, differentiation methods and digital filters, including neural networks (NNs)[24], Hilbert transform [14] [20] , are used for detection of QRS complex or the R-point in the ECG signal processing. However, robustness and high detection accuracy still remain open problems. In this paper, we exploit wavelet transforms and average-absolute difference threshold to detect more accurately the morphology of QRS complex.

Definition 1 (Wavelet Transform) The wavelet transform decomposes non-stationary signal into a number of scales having different frequency component and analyses each scale with a certain resolution for getting accurate features of the signal.

$$H(a,b) = \int_{-\infty}^{+\infty} x(t)\varphi_{a,b}(t)\, dt \tag{1}$$

$$\varphi_{a,b}(t) = \frac{1}{\sqrt{a}}\varphi^*(\frac{t-b}{a}) \tag{2}$$

Where, $x(t)$ is the original signal, * denotes the complex conjugation, $\varphi_{a,b}(t)$ is the window function of the mother wavelet and $\varphi^*(\frac{t-b}{a})$ is its shifted and scaled version.

Definition 2 (Hilbert Transform) A real valued time function is $y(t)$, and the Hilbert transform of the given signal is

$$z(t) = H[y(t)] = \frac{1}{\pi}\int_{-\infty}^{+\infty} y(\tau)\frac{1}{t-\tau}\, d(\tau) \tag{3}$$

Hilbert Transform exhibits the property of time dependency because the independent variable is not changed in accordance with the transformation.

Definition 3 (Absolute Difference Threshold) Discrete wavelet transform decomposes a signal at decomposition level n, the time axis is recursively divided into halves at the ideal cut-off time $\frac{1}{2^{n+1}}t_s$, where is sampling time. Let absolute difference be $\Delta y_i = |y_i - y_{i-1}|$, where amplitude (mV) y_i, y_{i-1} respectively is the at the time (second) of t_i, t_{i-1} .

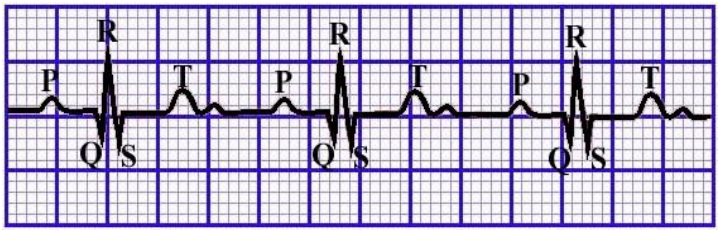


Fig. 3. ECG signals extracted by Fog-assisted gateway

QRS complex comprises of **Q** wave, **R** wave and **S** wave generated due to ventricular depolarization ((see Fig. 3). Detection of QRS complex is the entry point of almost all ECG signals analysis technique. In most of the ECG signals **R** wave appears as a sharp peak in between **Q** and **S** waves, which are of lesser amplitude and duration with respect to **R** wave. QRS complex is the region between **Q** wave onset and **S** wave offset. The detection of the R-peak algorithm is summarized as:

ECG sensors record the signals of physiological in real-time mode, and Fog computing node extracts original ECG signal. ECG signal is de-noised by wavelet transform. The enhanced signal undergoes to the differentiation to maximize the R-peaks with zero-crossing on x-axis. The differentiated signal is processed by Hilbert transform to provide a region of finding real QRS complex. Absolute difference threshold $|\Delta y_i|$ is calculated in real QRS complex and set its average as set the threshold ξ . Let difference sequence be $\Delta y_i = |y_i - y_{i-1}|$,i=1,2,. . . ,n, if $|\Delta y_{i_0}| > \xi$, then y_{i_0} the maximum point of y_{i_0} region is determined as the R-peak.

4.3. Intelligent Warning Model Based on Subband Energy Feature

Fog-assisted gateway collects physiological data from physiological sensor nodes. One of goals in healthcare monitoring is detecting health changes and asses health states. In particular, drastic health changes result from abrupt changes in process dynamics. Fog-assisted gateway collects physiological monitoring data in real-time from physiological sensor network. Furthermore, it needs to effective to distinguish and find the cause of the abrupt changes, as far as possible to remind the deviation type and avoid false alarm. In general, the process of intelligent warning is as follows:

(i) Fog-assisted gateway collects the signal under normal condition and establishes sample library of feature parameters.

(ii) Based sample library, Fog-assisted gateway establishes and holds knowledge base and inference rule base.

(iii) The physiological sensor keeps continuous monitoring patient, and Fog-assisted gateway receives signals and compares with the standard sample library of feature parameters to determine the current health state.

(iv) If the current state is "Red" state, Fog-assisted gateway starts alarm and finds out the reasoning from knowledge base.

Takes ECG monitoring for example, abrupt signal in ECG waveform is a sharp ridge. The feature of subband energy distribution is extracted by discrete wavelet transform. ECG waveform can be decomposed into high-frequency subbands and low-frequency subbands. So it needs to design perfect reconstruction filters to extract the vector of numerator coefficients of subbands energy feature. Subband analysis algorithm is summarized as:

(i) By using Orthogonal wavelet packet decomposition of ECG signal sequence, the coefficients of high frequency and low frequency are

$$c_{j,k} =< f(t), 2^{j/2}\phi(2^j t - k)|j, k \in Z > \tag{4}$$

$$a_{j,k} =< f(x), 2^{j/2}\phi(2^j x - k)|j, k \in Z > \tag{5}$$

where $\phi(t)$, $\phi(x)$ are orthogonal scaling functions,j is a scale parameter, k is a time-locationzation parameter, and wavelet function $\phi(t), \phi(x) \in L^2(R)$.

(ii) Let $f_{i,j}(t_j)$ be ECG signal in node (i, j) of wavelet packet decomposition tree (the j^{th} node of level i) which is reconstructed by wavelet packet decomposition. Sequences of subband energy areobtained by

$$E_{i,j} = \sum_{k=1}^{m} |x_{j,k}|^2 \tag{6}$$

where $x_{j,k}$ (j=0,1,2,...,2^j-1;k=1,2,...,m) is the ampltude of discrete point of $f_{i,j}(t_j)$, m is the number of sampling point in node (i, j).

(iii) The feature vector $[\lambda_0, \lambda_1, ..., \lambda_{(2^i-1)}]$ of i level is constructed as

$$\lambda_j = \frac{E_{i,j}}{\sum_{j=1}^{2^i-1} E_{i,j}} \qquad j = 1, 2, ..., 2^i - 1. \tag{7}$$

(iv) Fog-assisted gateway calculates deviation between ECG feature vector and sample library of feature parameters under normal condition, and determines abrupt signal

caused by sensors fault(sensor periodic fault, sensor blockage fault, sensor bias fault) or heart disease.

4.4. Security Framework of HL7 RIM-based Data Exchange

Fog-assisted gateway should realize the data exchange with Health Cloud which generally adopts SaaS mode. Because the monitoring data models are different, and only by adopting uniform standards, barrier-free transmission can be solved and interpreted unambiguously by different receivers. HL7 (Health Level 7) [6] standard is an international standard for data transmission of medical and health institutions and medical instruments and equipment authorized by the National Standards Agency (ANSI) of the United States. HL7 International specifies a number of flexible standards, guidelines, and methodologies by which various healthcare systems can communicate with each other. Such guidelines or data standards are a set of rules that allow information to be shared and processed in a uniform and consistent manner. The HL7 RIM data model [11] can clearly express the timing, hierarchy and logic. The purpose is to solve the inconsistency of information standards developed and formulated by different developers and provide a reference model at the highest level for standard developers and formulators. For example, following HL7 RIM, HL7 aECG (the HL7 Annotated Electrocardiogram) [4] is a standard medical record data format for storing and retrieving electrocardiogram data for a patient.

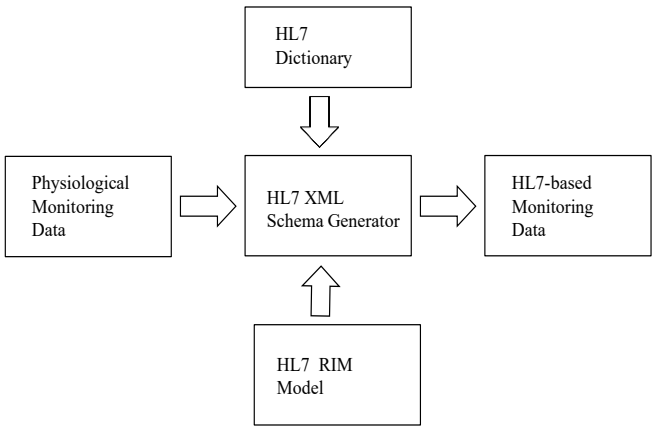


Fig. 4. HL7 RIM-based data transformation framework

Fig. 4 shows a unified data transformation framework based on HL7 RIM, including HL7 dictionary, HL7 RIM model and HL7 XML schema generator. Each data element is either composed of simple attributes into composite data, or a list of data elements, i.e. each is a composite data type attribute. HL7 RIM uses any element, data type, and vocabulary in the HL7 dictionary deriving from RIM specifications to ensure consistency. HL7 XML schema generator transforms physiological monitoring data into unified data

format following HL7 standards. Consequently, physiological monitoring data is to easily share clinical information.

HL7 RIM-based data transformation framework is conducive to realize data exchange between Fog-assisted gateways and health cloud, further support personal health record management and health risk assessment. Theoretically, this ability to exchange information should help to minimize the tendency for medical care to be geographically isolated and highly variable. HL7 RIM-based data is encapsulated as XML request, and sends to receiver through Simple Object Transfer Protocol (SOAP).

Aiming at the characteristics of security requirements of HL7 RIM-based data exchange, combing the existing security models (e.g., Web Services Security specification [2]), a security framework of data exchange is presented (see Fig. 5). The XML SOAP request processing is composed of a series of message handlers as follows: security attribution handler is responsible for adding security attributions including time stamp, period of validity and sender into original SOAP message; digital signature handler puts the digital signature to the message based on XML-Signature specification[7][15]; encryption handler is in charge of encrypting SOAP message where cipher key is distributed according to XKMS specification[8]. Each handler provides the functionality for parsing XML SOAP requests and dispatching the calls to the appropriate methods[16].

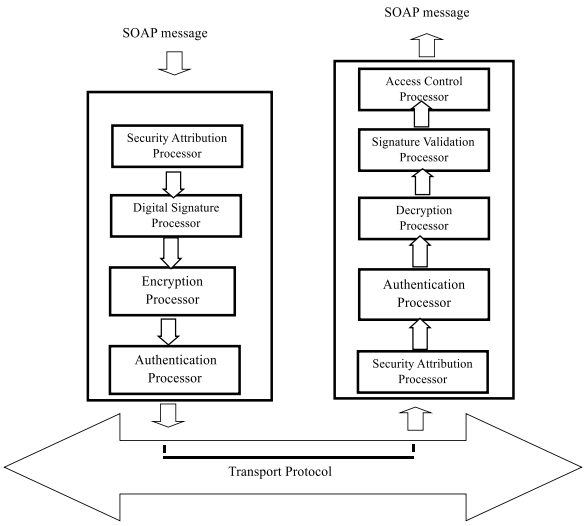


Fig. 5. Security framework of HL7 RIM-based data exchange

4.5. Health Risk Assessment Based on Fusion of Grey model and Markov Model

Health Risk Assessment (HRA) is used to provide individuals with an evaluation of their health risks and quality of life. The Framingham risk assessment (FRS) model [9] is a classic health risk assessment model, which is used to predict risk of individual cardiovascular disease in the next 10 years. Because of different countries and regions, people's

living habits are different, FRS model is not universal. For example, RAWG (Risk Assessment Work Group) desired to build upon experience with prior Framingham 10-year CHD risk prediction equations and the more recent Framingham 10-year general CVD risk prediction equations. RAWG elected to capitalize on the extensive data from several large NHLBI-sponsored cohort studies to derive a more geographically and racially diverse database. Aiming at the limitations of Framingham model, we choose Chinese medicine recognized factors, including age, body weight, blood pressure, blood glucose and BMI, etc. Health Cloud holds health monitoring archives and electronic medical records in Cloud data center. In order to help patients to have a comprehensive understanding of health conditions, health risk assessment model based on fusion of grey model and Markov model is applied to predict "relative risk" and "absolute risk" of individual's health status. "Relative risk" refers to the possibility of certain chronic diseases compared with the same age group and the average level of other people. "Absolute risk" is the possibility of individuals suffering from certain chronic diseases in the next few years.

Definition 4 (Baseline Incidence) Let RR_i be the relative risk of a certain level of factors, P_i be the proportion of individuals who are exposed to a level of the total population, then the baseline incidence rate is $RR_i = \frac{1}{\sum_{j=0}^{n}(RR_i \times P_i)}$.

Definition 5 (Relative Risk) Risk Score=(Baseline Incidence) $\times$ (Relative Risk). Let P be the combination of m factors $P = (P_1 - 1) + (P_2 - 1) + ... + (P_n - 1) + Q_1 \times Q_2 \times ... \times Q_m$, where P_i be risk factors (greater than or equal to 1), Q_i be risk factors (less than 1).

There are many models which can be used in chronic disease forecasting in "Absolute Risk", such as Markov chain models, Grey Models, General Regression Models, Autoregressive Integrated Moving Average Class models (ARIMA) and Neural network. However, these models typically require large numbers of observations and complicated input factors to make sensible predictions. Physiological monitoring data has the characteristics of random fluctuation. For better forecasting performance, hybrid models which combined two or more single models for communicable disease forecasting have also been explored, and previous findings indicate that hybrid models outperformed single models. A hybrid approach combining Grey model GM(1,1) and Markov Model to forecast the prevalence of physiological monitoring data.

(i) GM(1,1) Model

Step1. Let original data sequence be $X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), ..., x^{(0)}(n)\}$.

Step2. $X^{(1)}$ is obtained by 1-AGO (Accumulated Generating Operation)

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), ..., x^{(1)}(n)\} \tag{8}$$

where $x^{(1)}(t) = \sum_{i=1}^{t} x^{(0)}(i)$, *t*=1,2,3…,$n$.

Step3. The grey differential equation of GM(1,1) of $x^{(1)}(t)$ is as follows

$$\frac{\mathrm{d}x^{(1)}}{\mathrm{d}t} + ax^{(1)} = u \tag{9}$$

where a, u are obtained respectively by using least square method $x^{(1)}(t) = [x^{(0)}(1) - \frac{u}{a}]e^{(-at)} + \frac{u}{a}$.

Step4. Applying the Inverse Accumulated Generating Operation (IAGO), and then we have

$$\hat{x}^{(0)}(t+1) = (1 - e^{a})[\hat{x}^{(0)}(1) - \frac{b}{a}]e^{(-at)} \tag{10}$$

(ii) Residual Error Correction of GM(1,1) Model
Step 1. Let residual error sequence be

$$\varepsilon^{(0)}(t) = |x^{(0)}(t) - \hat{x}^{(0)}(t)| \qquad i = 1, 2, 3, ..., n \tag{11}$$

Step 2. $\hat{x}^{(0)}(t+1)$ is obtained by similar method as follows

$$\hat{x}^{(0)}(t+1) = (1-e^{a})[x^{(0)}(1) - \frac{u}{a}]e^{(at)} + sgn(t+1)(1-e^{a_1})[\varepsilon^{(0)}(1) - \frac{u_1}{a_1}]e^{(-a_1 t)} \tag{12}$$

where symbol function $syn(t)$ is obtained by the original residual errors.
(iii) Markov Model
Markov chain is a forecasting method which can be used to predict the future data by the occurred events. We can get the simulation sequence by Equation 13 as follows

$$\hat{y}^{(0)} = \{\hat{y}^{(0)}(1), \hat{y}^{(0)}(2), ..., \hat{y}^{(0)}(n)\} \tag{13}$$

We can divide $\hat{y}^{(0)}$ into n states. According to the relative error, any state can be denoted as $\Theta_j = [\Theta_{j-1}, \Theta_{j+1}]$, where $\Theta_{j-1} = \hat{y}^{(0)}(j) + a_j$, $\Theta_{j+1} = \hat{y}^{(0)}(j) + b_j$. Assume n_j is the number of original sequence, the transition probability from Θ_i to Θ_j can be established by Equation14 as follows

$$P_{ij}(k) = \frac{n_{ij}(k)}{n} \qquad i = 1, 2, 3, ..., n \tag{14}$$

Where $P_{ij}(k)$ is the transition probability of state Θ_j transferred from state Θ_i for k steps. Transition probability matrix can be expressed as $P(k) = (P_{ij}^{(k)})_{n\times n}$,where $P_{i1}^{(k)} + P_{i2}^{(k)} + ... + P_{in}^{(k)} = 1$;$i$=1,2,...,$n$. The transition probability matrix $P(k)$ reflects the transition rules of the states in a system, which is the foundation of the Grey-Markov model.In order to confirm future state transition, after the relative residual error range $[\Theta_{j-1}, \Theta_{j+1}]$ is obtained, the median in range $[\Theta_{j-1}, \Theta_{j+1}]$ (i.e.,$\hat{y}^{(0)}(j) + \frac{a_j + b_j}{2}$) is selected as the relative error. Furthermore, forecasting value of original data sequence is obtained by combining with Equation12 and Equation14.

5. Experimental Setup and Analysis

5.1. Working Process of Fog-assisted Healthcare Monitoring System

In this section, experiments to test the framework of Fog-assisted Healthcare Monitoring are provided. We developed the prototype system of proposed framework, which is connected with Health Fog and Health Cloud[12]. Health Fog has 10 Fog computing servers. Each server is configured with Linux operating system, Xen VMM and 5 virtual machines, as Fog-assisted gateway. Health Cloud has cloud resource pool with 40 vCPU, 100G memory, 10T storage, support a variety of health archives management including physiological factors (such as “age”, “sex”, “height”, “weight”, “smoking”, “diabetes family history”, “ECG”, “SpO2”, etc.). Health Cloud has collected personal health records more than 50000 copies. Wireless body sensor network is composed of physiological sensors, including oxygen sensor, glucose sensor, accelerometer, and so on. IPv6-based network

architecture is composed of wireless body sensor network, IPv6 transmission network, Health Fog, and Health Cloud, mobile intelligent device. Subsets of 2.4 GHz band wireless IEEE 802.11 and IEEE 802.15 family standards (Wi-Fi and Bluetooth) were needed. In order to determine the validity of the framework, we selected the 55-65 years old (male) in Xiamen District of Jimei from as the research object. Health monitoring data of patients was systematically generated for 30 days. The working process of the system is as follows (See Fig. 6):

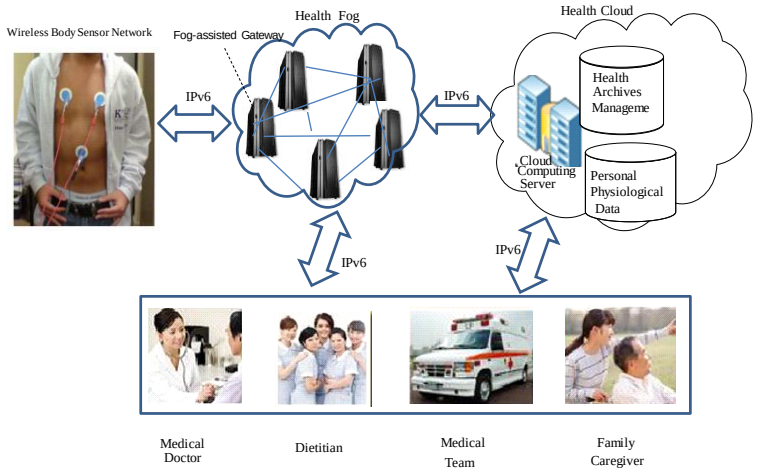


Fig. 6. Working process of Fog-assisted healthcare monitoring system

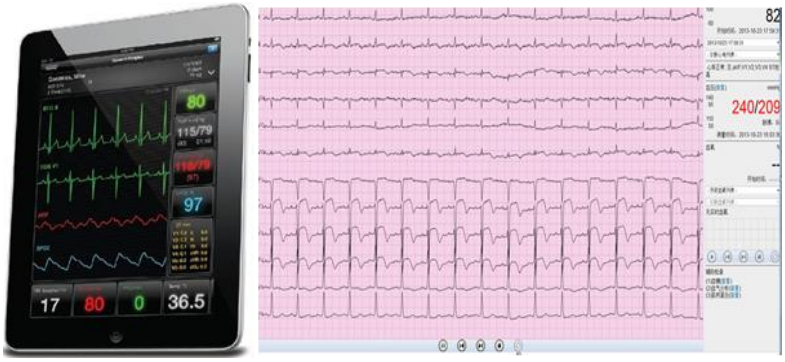


Fig. 7. Interface display of mobile intelligent device

(i) Mobile intelligent device connects Health Fog based on wireless IPv6 WIFI, and registers personal information, including personal data (“age”, “sex”, “degree of Educa-

tion", "diabetes family history", etc.), life preferences ("smoking", "drinking", "physical exercise", etc.), routine physical examination data (BMI(Body Mass Index; "weight", "height", "waist circumference", "hip circumference"), WHR(Waist-to-Hip Ratio)).

(ii) Wireless Body Sensor network is composed of a series of intelligent physiological sensors which generate physiological data, including "systolic blood pressure", "SpO2", "fasting blood glucose", and "heart rate", etc.

(iii) Health Fog is composed of Fog-assisted gateway, which has a built-in Bluetooth gateway as boundary node of Wireless Body monitoring network, which receives data from different subnetworks, performs protocol conversion, and provides other higher-level services such as data preprocessing, local analysis and services, including intelligent alarm, on-line real-time monitoring, and notification service. Apart from that, Fog-assisted gateway also provides substantial connectivity and allows sensor nodes to move within the Internet topology while maintaining reachability.

(iv) Fog-assisted gateway in Health Fog sends physiological data to Health Cloud for further analysis based on security framework of HL7-RIM data exchange. Health Cloud implements health archives management and personal physiological data. Health cloud provides intelligent classification service and risk assessment service of chronic disease. Health cloud also provides some healthcare services and health intervention with doctors, nutritionists and other medical team.

(v) Mobile intelligent device receives health risk assessment and alarm messages as quickly as possible. Fig. 7 shows the interface display of mobile intelligent device for continuous monitoring of the patient's health.

5.2. Intelligent Alarm Service

Whenever, the "heart rate", "blood pressure", "body temperature" and "fasting blood glucose" exceed the normal values, mobile intelligent device receives alarm messages from Fog-assisted gateway as quickly as possible. Mobile intelligent devices can send an alert message with clinical value to the doctor and family caregiver using wireless network through Health Fog. Takes ECG monitoring for example, the abrupt signal may be caused by heart disease, sensor periodic fault, sensor blockage fault, sensor bias fault, which need to be distinguished in time. Table 1 shows some subbands energy characteristics of abrupt signals.

Table 1. Subbands energy characteristics of abrupt signals

Chronic diseases	High frequency signals (%)	Low frequency signals (%)
Heart disease	0-1%	99%-100%
Sensor periodic fault	98.1%-99.8%	0.2%-1.9%
Sensor blockage fault	0	0
Sensor bias fault	3.1%-7.2%	92.3%-96.9%

When the abrupt signal is judged to be the cause of the human disease, the prototype system detects and determines two adjacent R wave time according to the mean absolute differential threshold algorithm, and makes further five types of electrocardiogram types:

- Ventricular Tachy $(RR_i < 500ms)$;
- Bradycardia $(RR_i > 1500ms)$;
- Ventricular Premature $(HRV_i > 120ms)$;
- Ventricular Parked $(RR_i > 3000ms)$;
- Ventricular Leakage $(2.3RR < RR_i < 2.6RR)$.

Under normal circumstances, if five consecutive times meet the above characteristics, prototype system sends out the alarm and messages to the relatives,the doctor,or caregivers. When it is determined that the abrupt signal is a sensor bias fault, the prototype system prompts the human body to take the correct posture to avoid device shaking.

5.3. Response Latency

In order to evaluate performance of protype system, we compared response latency of the system under IPv4-based Cloud-assisted environment or IPv6-based Fog-assisted environment. Response latency of the average system under IPv4-based cloud assisted environment is 118.43ms, and that under IPv6-based Fog assisted environment is 26.42ms. Mobile intelligent device received response latency in different environment are shown in Fig. 8. The main reason is as follows: (i) In IPv4-based Cloud-assisted environment, mobile intelligent device needs long-distance communication overhead with Health cloud. In cloud computing where raw data is transferred from sensor nodes to cloud, if network condition is unpredictable, it may cause uncertainty to response delay. However, in IPv6-based Fog-assisted environment, mobile intelligent device is close to Health Fog with short-distance communication. In Fog computing where implementing data analytics and making time-sensitive decisions within the local network makes the proposed system more robust and predictable. (ii) The efficiency of data preprocessing, on-line real-time mointoring, intelligent alarm and notification service are similar in two environments. Therefore, The IPv6-based Fog-assisted environment response latency proposed in this paper is much lower than that of IPv4-based Cloud-assisted environment.

5.4. Health Risk Assessment

The chosen factors in this case are comprehensive, including: “degree of education”, “smoking”, “physical exercise”, “heart Rate”, “BMI”, “systolic blood pressure”, “cerebral stroke”, “fasting blood glucose”. Table 2 shows the baseline incidences and Risk scores of different risk factors. According to personal physiological monitoring data and health archives in health cloud, we take one old people aged 61 for example, who has “primary education”, “no smoking”, “no physical exercise”, “BMI” (Obesity), “blood pressure”(140 mmHg/209 mmHg), “heart rate”(92), “no cerebral stroke”, “blood glucose” (super high). “Relative risk” of chronic disease is 15.666 (1.658+2.873+2.885+6.543 +1.778+5.325−6+0.952× 0.926 =15.666). The total incidence of Hypertension in Xiamen District of Jimei is close to 5%, so this old people’s current “absolute risk” of chronic disease is 15.666*5$\%$=78.78$\%$.

A hybrid approach combining Grey model GM (1,1) and Markov Model can be used to forecast absolute risk within 5 years. The working process of health risk assessment based on Fusion of Grey model and Markov Model is as follows:

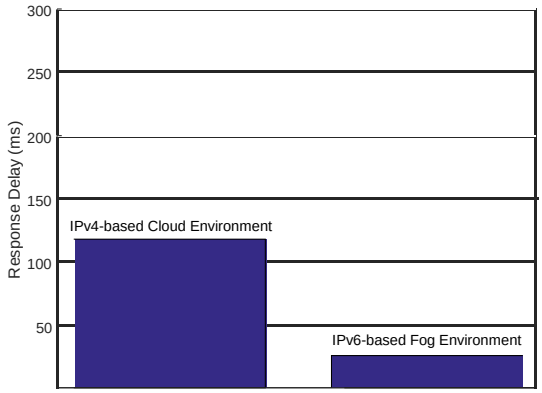


Fig. 8. The result of response latency testing

(i) applying the inverse accumulated generating operation (IAGO) to forecast condition of personal physiological monitoring;

(ii) computing the residual errors of GM(1,1) Model;

(iii) relying on Markov chain to predict the future data by the occurred events. For example, states of blood pressure includes:

- Hypotension ("Systolic"<90mmHg, "Diastolic"<60 mmHg);
- Normotensive ("Systolic" 90-119mmHg, "Diastolic" 60-79mmHg);
- Prehypertension ("Systolic" 120-139mmHg, "Diastolic" 80-90mmHg);
- Stage I Hypertension ("Systolic" 140-159mmHg, "Diastolic" 90-99mmHg);
- Stage II Hypertension ("Systolic" 160-179mmHg, "Diastolic" 100-109mmHg);
- Hypertensive Urgency ("Systolic" >180 mmHg, Diastolic">110 mmHg);
- Isolated Systolic Hypertension ("Systolic" >160 mmHg, "Diastolic" <90 mmHg).

The transition probability matrix is obtained by Equation 14 and Markov Model is used to predict next state by the occurred states.

6. Conclusions

Fog computing nodes as smart gateways at the proximity of sensor nodes can tackle many challenges in ubiquitous healthcare monitoring such as mobility, scalability, and network latency. In this paper, a framewrok for Fog-assisted healthcare monitoring is proposed. This framework is composite of body-sensing layer, fog layer and cloud layer. The body-sensing layer measures physiological signals and Fog-assisted gateway collects these information. Fog-assisted gateway also provides protocol conversion, intelligent alarm, online realtime monitoring and notification service. Physiological monitoring data is transferred to cloud layer for further processing and analysis, including health risk assessment and intelligent classification. The contribution of the paper can be summarized as follows: (i) proposing a hierarchical framework for Fog-assisted healthcare monitoring; (ii)

Table 2. Risk factors and risk score of diabetic inpatients in Xiamen (55-65 years old)

Factors	Factor Values	Baseline Incidences	Risk Scores
Degree of Education	Primary and Secondary Education	0.952	0.952
	Middle School	0.901	0.754
	College Degree or Above	1.105	0.524
Smoking	NO	1.658	1.658
	YES	0.680	2.572
Physical Exercise	NO	0.926	0.926
	YES	0.926	0.784
Heart Rate	<90 Normal	0.658	0.765
	>90 Abnormal	1.043	2.873
BMI	Normal	1.265	1.625
	Overweight	0.961	1.307
	Obesity	0.984	2.885
Blood Pressure	<120 mmHg	0.123	0.223
	120-140 mmHg	0.771	0.771
	140-160 mmHg	0.976	2.521
	160-180 mmHg	1.383	4.654
	>180 mmHg	1.573	6.543
Cerebral Stroke	NO	0.978	0.978
	YES	0.978	1.778
Fasting Blood Glucose	Normal	0.926	0.926
	High	0.978	4.152
	Super High	1.548	5.325

enabling underlying network (i.e., IPv6-based Network Architecture) to provide mobility of patients with different protocols; (iii) giving intelligent warning model based on subband energy feature; (iv) proposing security framework of HL7 RIM-based data exchange; (v) providing health risk assessment based on fusion of grey model and Markov model. The next step is to improve the intelligent alarm based on CNN (Convolutional Neural Network) and reduce communication delay in the presence of large-scale data collection.

Conflict of Interest. Authors Jianqiang Hu, Wei Liang, Zhiyong Zeng, Yong Xie and Jianxun Eileen Yang declare that they have no conflict of interest.

Acknowledgments. This work was supported by the CERNET Innovation Project under grant no. NGII20160708; Fujian Provincial Natural Science Foundation of China under grant no. 2019J01856; Open Research Fund Project of Key Laboratory of Internet of Things Application Technology under grant no. XMUTIoT201803; Fundamental Research Funds for the Committee of Science and Technology in Shenzhen under grant no. JSGG20160607161350293; Natural Science Foundation of China under grant no. 61872436.

References

1. Ahmad, M., Amin, M.B., Hussain, S., Kang, B.H., Cheong, T., Lee, S.: Health fog: A novel framework for health and wellness applications. The Journal of Supercomputing 72(10), 3677–3695 (2016)
2. Atkinson, B., Della-Libera, G., Hada, S., Hondo, M., Hallam-Baker, P., Klein, J., LaMacchia, B., Leach, P., Manferdelli, J., Maruyama, H., et al.: Web services security (ws-security). Specification, Microsoft Corporation (2002)
3. Banos, O., Villalonga, C., Damas, M., Gloesekoetter, P., Pomares, H., Rojas, I.: Physiodroid: Combining wearable health sensors and mobile devices for a ubiquitous, continuous, and personal monitoring. The Scientific World Journal 2014 (2014)
4. Brown, B.D., Badilini, F., et al.: Hl7 aecg implementation guide. Regulated Clinical Research Information Management Technical Committee (2005)
5. Chiang, H.P., Lai, C.F., Huang, Y.M.: A green cloud-assisted health monitoring service on wireless body area networks. Information Sciences 284, 118–129 (2014)
6. Dolin, R.H., Alschuler, L., Beebe, C., Biron, P.V., Boyer, S.L., Essin, D., Kimber, E., Lincoln, T., Mattison, J.E.: The hl7 clinical document architecture. Journal of the American Medical Informatics Association 8(6), 552–569 (2001)
7. Eastlake 3rd, D., Reagle, J., Solo, D.: Xml-signature syntax and processing. Tech. rep. (2001)
8. Ford, W., Hallam-Baker, P., Fox, B., Dillaway, B., LaMacchia, B., Epstein, J., Lapp, J.: Xml key management specification (xkms). W3C note, March (2001)
9. Greenland, P., LaBree, L., Azen, S.P., Doherty, T.M., Detrano, R.C.: Coronary artery calcium score combined with framingham score for risk prediction in asymptomatic individuals. Jama 291(2), 210–215 (2004)
10. Hameed, R.T., Mohamad, O.A., Ţăpuş, N.: Health monitoring system based on wearable sensors and cloud platform. In: 2016 20th International Conference on System Theory, Control and Computing (ICSTCC). pp. 543–548. IEEE (2016)
11. Hasman, A., et al.: Hl7 rim: an incoherent standard. In: Ubiquity: Technologies for Better Health in Aging Societies, Proceedings of Mie2006. vol. 124, p. 133 (2006)
12. Hu, J., Wu, K., Liang, W.: An ipv6-based framework for fog-assisted healthcare monitoring. Advances in Mechanical Engineering 11, 1–13 (2019)
13. Johnson, D., Perkins, C., Arkko, J.: Mobility support in ipv6. Tech. rep. (2004)
14. Kopsinis, Y., McLaughlin, S.: Development of emd-based denoising methods inspired by wavelet thresholding. IEEE Transactions on signal Processing 57(4), 1351–1362 (2009)
15. Lee, J., Kim, S., Moon, K., Chung, K., Sohn, S.: Method and apparatus for providing xml signature service in wireless environment (Jun 14 2007), uS Patent App. 11/635,367
16. Liang, W., Ruan, Z., Wang, Y., Chen, X.: Resh: a secure authentication algorithm based on regeneration encoding self-healing technology in wsn. Journal of Sensors 2016 (2016)
17. Nandyala, C.S., Kim, H.K.: From cloud to fog and iot-based real-time u-healthcare monitoring for smart homes and hospitals. International Journal of Smart Home 10(2), 187–196 (2016)
18. Negash, B., Gia, T.N., Anzanpour, A., Azimi, I., Jiang, M., Westerlund, T., Rahmani, A.M., Liljeberg, P., Tenhunen, H.: Leveraging fog computing for healthcare iot. In: Fog Computing in the Internet of Things, pp. 145–169. Springer (2018)
19. Rahmani, A.M., Gia, T.N., Negash, B., Anzanpour, A., Azimi, I., Jiang, M., Liljeberg, P.: Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach. Future Generation Computer Systems 78, 641–658 (2018)
20. Sahoo, S., Biswal, P., Das, T., Sabut, S.: De-noising of ecg signal and qrs detection using hilbert transform and adaptive thresholding. Procedia Technology 25, 68–75 (2016)
21. Sood, S.K., Mahajan, I.: A fog-based healthcare framework for chikungunya. IEEE Internet of Things Journal 5(2), 794–801 (2018)

22. Verma, P., Sood, S.K.: Cloud-centric iot based disease diagnosis healthcare framework. Journal of Parallel and Distributed Computing 116, 27–38 (2018)
23. Wang, S.L., Chen, Y.L., Kuo, A.M.H., Chen, H.M., Shiu, Y.S.: Design and evaluation of a cloud-based mobile health information recommendation system on wireless sensor networks. Computers & Electrical Engineering 49, 221–235 (2016)
24. Zhang, Y., Xiao, H.: Bluetooth-based sensor networks for remotely monitoring the physiological signals of a patient. IEEE transactions on Information Technology in Biomedicine 13(6), 1040–1048 (2009)

Jianqiang Hu (Co-correspondence Author) now is an associate professor in school of computer and information engineering, Xiamen University of Technology, China. He once worked as a postdoctoral researcher at Tsinghua University. He received his Ph.D. degree in computer science and engineering from National University of Defense Technology, China, in 2005. He is the author of more than 60 articles, and more than 5 inventions. His current research interests include Cloud computing, Big Data Analytics, and Internet of Things.

Wei Liang now is a professor of Xiamen University of Technology, China. He received his Ph.D. degree at Hunan University in 2013. He is a postdoctoral scholar of Department of Computer science and Engineering at Lehigh University in USA in 2014-2016. He has published more than 110 journal/conference papers. His research interests include Networks Security Protection, embedded system and Hardware/IP protection, and Fog computing, and Security management in WSN.

Zhiyong Zeng now is associate professor with the school of mathematics and informatics, Fujian Normal University. He is the author of more than 50 articles, and more than 5 inventions. His research interests include digital image processes and applications, computer vision, artificial intelligence, and machine learning.

Yong Xie now is an associate professor in school of computer and information engineering, Xiamen University of Technology, China. He received his Ph.D. degree in computer science and engineering from Hunan University, China, in 2013. His major interests include embedded real-time systems and cyber-physical systems.

Jianxun Eileen Yang (Co-correspondence Author) now is the Executive Dean of Shenzhen Research Institute of Sun Yat-sen University. She is a special foreign expert at Sun Yat-sen University, and level A high-level talents in Shenzhen. Dr. Yang has hosted and participated in dozens of national, provincial and municipal projects. She has won the Outstanding Contribution Award for China's MBA Education. Her research interests include technology transformation, finance, fintech and business management.

Received: September 30, 2018; Accepted: September 1, 2019.

Computer Science and Information Systems 16(3):773–795 https://doi.org/10.2298/CSIS181001026D

Design of Intrusion Detection System Based on Improved ABC_elite and BP Neural Networks

Letian Duan, Dezhi Han, and Qiuting Tian

College of Information Engineering, Shanghai Maritime University
Shanghai 201306, China
1812796760@qq.com, dzhan@shmtu.edu.cn, tqt2018@163.com

Abstract. Intrusion detection is a hot topic in network security. This paper proposes an intrusion detection method based on improved artificial bee colony algorithm with elite-guided search equations (IABC_elite) and Backprogation (BP) neural networks. The IABC_elite algorithm is based on the depth first search framework and the elite-guided search equations, which enhance the exploitation ability of artificial bee colony algorithm and accelerate the convergence. The IABC_elite algorithm is used to optimize the initial weight and threshold value of the BP neural networks, avoiding the BP neural networks falling into a local optimum during the training process and improving the training speed. In this paper, the BP neural networks optimized by IABC_elite algorithm is applied to intrusion detection. The simulation on the NSL-KDD dataset shows that the intrusion detection system based on the IABC_elite algorithm and the BP neural networks has good classification and high intrusion detection ability.

Keywords: Intrusion Detection, Machine Learning, BP Neural Networks, Improved ABC_elite.

1. Introduction

With the development of network technology, especially the widespread use of the Internet, it is more convenient to interconnect and interoperate. However, the problem of network security is also becoming increasingly prominent, and lawbreakers may seek benefits through damage to the network. Therefore, the detection and defense of network attacks is a hot topic of network security. Attackers usually leverage network protocol vulnerability to perform network attacks, including Denial of Service (DoS), User to Root (U2R), Probe and Root to Local (R2L). So far, methods for detecting network attacks include classification and clustering, which detect network attacks by analyzing network data flow.

Machine learning methods have been widely used to identify different types of attacks, and machine learning methods can help network administrators take appropriate measures to deal with network attacks. However, most of these traditional machine learning methods belong to shallow learning and require a lot of feature extraction and feature selection. They can't solve the problem of a large number of attacks and intrusion data classification problems in real network environments. In addition, shallow learning is not suitable for the intelligent analysis and forecasting of high-dimensional and massive data. However, the Backpropagation (BP) neural network model has good adaptability, self-learning ability and nonlinear approximation ability, can meet these requirements, and

has been widely used in prediction, modeling, classification, adaptive control and other fields.

The Swarm intelligence is simply defined as the collective behavior of decentralized and self-organizing swarms. As we all know, these swarms can be birds, fish and the colony of social insects such as ants and bees. In the 1990s, especially two methods, based on the ant colony and bee colony, attracted the interest of researchers. The Swarm intelligence requires a colony to meet self-organizing features. Since the 21st century, researchers have started to be interested in new intelligent methods using bee colony behavior. In the past decade, some algorithms based on various intelligent behaviors of honey bee colony have been proposed. The population algorithm originates from the resolution of numerical optimization problems and is a meta-heuristic target optimization algorithm.

In view of the shortcomings of traditional neural network training methods, researchers began to try to apply heuristic search algorithms to artificial neural network design and parameter optimization. That is to say, heuristic search methods can be used to train neural networks. The fusion of the heuristic search algorithms and neural networks are called evolutionary neural networks [29,30]. Artificial Bee Colony Algorithm (ABC) is a new heuristic search algorithm. It was proposed at Erciyes University in 2005. The algorithm is derived from the research and simulation of honey bee collective behavior. Compared with other heuristic search algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Differential Evolution (DE), ABC algorithm is simple, robust as well as need few parameters.

The optimization technology is an application technology that uses mathematics as the basis to find the optimal feasible solution for the objective optimization problem under certain conditional constraints. Before the 1980s, the optimization algorithms mainly were used to mathematical analysis, iterative solution and other methods to solve practical problems. They were called traditional optimization algorithms. These methods have complete theoretical analysis and mathematical proofs, and have also achieved good results in optimization problems such as continuous and low-dimension. But they seem to be powerless for multi-peak, high-dimension and discontinuous optimization problems. From the 1980s, some novel heuristic search algorithms emerge, which are different from traditional optimization algorithms. For example, the Genetic Algorithm is a computational model that simulates the natural selection and genetic mechanism of Darwin biological evolution. The idea of Simulated Annealing algorithm comes from the process of solid material annealing in physics. All these algorithms simulate some processes that occur in nature.

The remainder of this paper is organized as follows. In Section 2, we review the related work about the development of artificial bee colony algorithm, as well as using artificial bee colony algorithm to train the artificial neural network, and some intrusion detection methods in recent years. And we show the innovation about our intrusion detection method, which is different from other intrusion detection methods. In Section 3, the description of BP neural networks about neuron model, multilayer feedforward neural networks, and Backpropagation algorithm is introduced. In Section 4, we describe the artificial bee colony algorithm and propose the improved ABC_elite algorithm. In Section 5, we introduce the BP neural networks based on IABC_elite algorithm. Section 6 highlights the experiment about the dataset description, the data preprocessing, as well as the

experiment results. We analyze the experiment results in detail. Finally, the conclusion is discussed in Section 7.

2. Related Work

The artificial bee colony algorithm is a heuristic swarm intelligence algorithm designed by Karaboga in 2005 to mimic the collective behavior of the honey bee. It was originally designed to solve some numerical optimization problems [12]. The artificial bee colony algorithm was used to optimize multivariate functions, and compared with algorithms such as genetic algorithm (GA) and particle swarm optimization (PSO). The results show that ABC is superior to other algorithms [14]. However, the artificial bee colony algorithm is good at exploring the solution, but it is poor in exploitation and easy to fall into a local optimum. Ref. [32] proposed an improved ABC algorithm called gbest-guided artificial bee colony (GABC) algorithm, which combines the information of the global optimal solution into the solution search equation to improve the exploitation. Ref. [27] proposed a multi-strategy ensemble artificial bee colony (MEABC) algorithm. In MEABC, a pool of distinct solution search strategies coexist throughout the search process and compete to produce offspring. The MEABC method effectively improves the performance of ABC on continuous optimization problems. Ref. [2] proposed an artificial bee colony algorithm with Elite-Guided Search Equations combined with a depth-first framework, named DFSABC_elite. Prioritizing more computing resources for better solutions enhances the exploitation capabilities of the algorithm.

The use of artificial bee colony algorithm to train artificial neural networks also has a certain historical background. In Ref. [13], the artificial bee colony algorithm is used to train neural networks to solve the benchmark problems of XOR, 3-bit parity, 4-bit encoder-decoder, etc., which embodies the ability of artificial bee colony algorithm to train neural networks. Ref. [21] proposed a hybrid algorithm combining artificial bee colony algorithm and Levenberq-Marquardt algorithm (ABC-LM) to train artificial neural networks. The neural network is first trained with ABC, and then LM continues to train the neural network using the optimal weight set of the ABC algorithm and attempts to minimize training errors. The results of the experiments show that the hybrid ABC-LM algorithm has better performance. However, this method has only been tested on benchmark issues such as XOR, 3-bit parity, 4-bit encoder-decoder, etc., and has not been tested on high dimensional classification benchmark problems.

In 1999, Wenke et al. applied the machine learning method to intrusion detection for the first time. By analyzing the network data traffic, an anomaly detection model was obtained [18]. In Ref. [23], a distributed denial of service attack (DDos) detection method based on BP neural network is proposed, which is carried out by simple traffic classification and feature selection. The detection of DDos traffic reaches a certain detection rate. However, this method has a long training time and is easy to falling into a local optimum. Ref. [31] proposed a recurrent neural network (RNN) deep learning method for intrusion detection, which was simulated in binary-classification and multi-classification environments, but the method still needs to improve the detection accuracy and reduce the training time of the neural network model.

This paper studies the intrusion detection method based on the improved ABC_elite and BP neural networks. The IABC_elite algorithm proposed in this paper improves the

DFSABC_elite algorithm by integrating the Gaussian search equation in Ref. [19] into the DFSABC_elite algorithm, which improves the exploitation ability of the artificial bee colony algorithm and accelerates the search for the optimal solution. The use of IABC_elite to optimize the BP neural network avoids the neural network falling into local optimum and solves the problem of slow convergence of neural network training to some extent. The intrusion detection method based on IABC_elite and BP neural network improves the accuracy of intrusion detection and reduces the false positive rate. In the NSL-KDD dataset, KDDTrain+ and KDDTest-21 were respectively selected as training dataset and testing dataset to verify the performance of the algorithm.

3. BP Neural Networks

The Artificial Neural Network is an intelligent algorithm by simulating the structure of the neural system and information delivery of biological neural networks. Each single neuron performs simple operations. When numerous neurons are combined and operated together, they make up a huge distributed and parallel computing model.

3.1. Neuron Model

Neurons are basic information processing units for neural networks. Fig. 1 shows a model of a neuron which is the basic unit of artificial neural networks. The neuron model comprises three basic elements:

1) Synapses, each of them are characterized by its weight value. In particular, the synaptic weight value w_{ij} is a coefficient that multiplies the input signal x_i. The first subscript of the weight value refers to the query neuron, and the second subscript refers to the receptor neuron where the weight value is located.

2) A linear combiner, it is for summing the input signals multiplied by the weight value of the corresponding synapse according to Eq.(1).

$$net_j = \sum_{i=1}^{m} w_{ij} x_i + \theta_j \tag{1}$$

Where x_i is the ith component of the input signal, m is the dimension of the input signal, w_{ij} is the ith component of the synaptic weight value of neuron j, θ_j is the threshold value of neuron j, and net_j is the output value of the linear combiner.

3) The activation function, which limits the amplitude of the neuron output signal and limits the value to a certain value within the allowable range. In general, the normal amplitude of a neuron output signal value is in the range of [0, 1] or [-1, 1].

The output signal value y_j of the neuron model is calculated according to Eq.(2).

$$y_j = \varphi(net_j) \tag{2}$$

3.2. Multilayer Feedforward Neural Networks

In layer neural networks, neurons are organized by the layers. In the most simple layer neural networks, the source node constitutes the input layer. It is directly projected onto

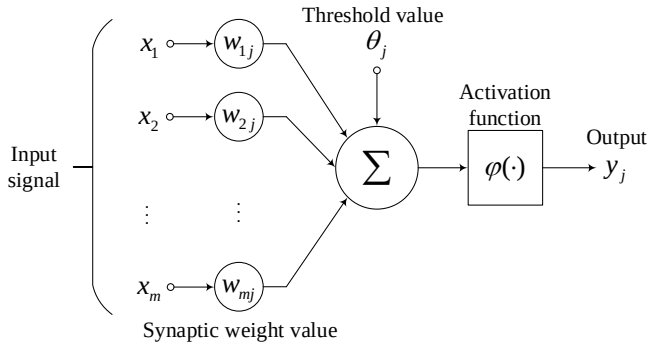


Fig. 1. Neuron model

the neuron of the output layer, rather than the opposite, which is called the feedforward neural networks. A multilayer feedforward neural network refers to one or more layers of hidden neurons in the neural network.

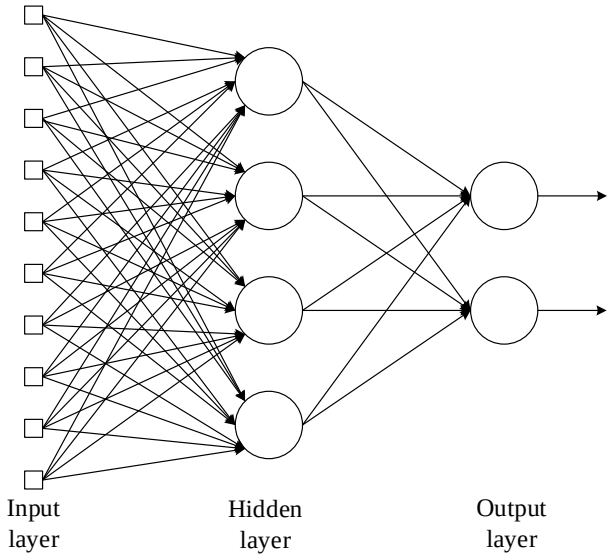


Fig. 2. Three-layer feedforward neural networks

The source nodes of the input layer provide an input signal to the hidden layer neurons. The input signal of each layer is the output signal of the previous layer. The output signal of one layer is used as the input signal of the next layer, and it is passed on. Finally, the output layer gives the neural network output signal of the corresponding original input signal. The structure is shown in Fig. 2. This is a 10-4-2 neural network with 10 source

nodes, 4 hidden neurons, and 2 output neurons. In general, a feedforward neural network has m source nodes, h neurons in the hidden layer and q neurons in the output layer, which can be called m-h-q neural networks. The output signal of output layer kth neuron is calculated according to Eq.(3).

$$z_k = \varphi_2 \left(\sum_{j=1}^{h} w_{jk} \varphi_1 \left(\sum_{i=1}^{m} w_{ij} x_i + \theta_j \right) + \theta_k \right) \tag{3}$$

Where $\varphi_1(\cdot)$ is the activation function of the hidden layer neuron, and $\varphi_2(\cdot)$ is the activation function of the output layer neuron.

3.3. Backpropagation Algorithm

Backpropagation (BP) algorithm is a common method to train artificial neural networks based on the gradient descent method. The gradient descent method minimizes the loss function by updating weight values and threshold values toward the fastest direction. Here, we derive the increment of weight values and threshold values.

For convenience of description, we use the square error function as the loss function according to the Eq.(4).

$$J(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{q} (t_k - z_k)^2 = \frac{1}{2} ||\mathbf{t} - \mathbf{z}||^2 \tag{4}$$

Where $\mathbf{t}$ and $\mathbf{z}$ are the target vector and the neural networks output vector whose length is q,$\mathbf{w}$ represents all the weight values in the neural networks.

The backpropagation rule is based on the gradient descent method. In the beginning, the weight values are initialized to random values and then adjusted toward the direction of reducing the error value.

$$\Delta \mathbf{w} = -\eta \frac{\partial J}{\partial \mathbf{w}} \tag{5}$$

This can be expressed in the form of a component,

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \tag{6}$$

Where η is the learning rate, which represents the relative change ratio of the weight values. We consider Eq.(6) in the three-layer neural networks.

1) We use the chain differentiation rule to entail the update rules of weight values and threshold values from the hidden layer to the output layer.

$$\frac{\partial J}{\partial w_{jk}} = \frac{\partial J}{\partial net_k} \frac{\partial net_k}{\partial w_{jk}} = -\delta_k \frac{\partial net_k}{\partial w_{jk}} \tag{7}$$

The sensitivity degree of unit k is defined as

$$\delta_k = -\frac{\partial J}{\partial net_k} \tag{8}$$

This sensitivity degree describes that the total error varies with the activation of the unit.

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k} = (t_k - z_k)\varphi_2{}'(net_k) \tag{9}$$

And due to

$$\frac{\partial net_k}{\partial w_{jk}} = y_j \tag{10}$$

So

$$\Delta w_{jk} = \eta\delta_k y_j = \eta(t_k - z_k)\varphi_2{}'(net_k)y_j \tag{11}$$

Same reason

$$\Delta\theta_k = \eta(t_k - z_k)\varphi_2{}'(net_k) \tag{12}$$

2) We entail the update rules of weight values and the threshold values from the input layer to the hidden layer.

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial y_j}\frac{\partial y_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} \tag{13}$$

Where

$$\begin{aligned}
&\tfrac{\partial J}{\partial y_j} = \tfrac{\partial}{\partial y_j}\left[\tfrac{1}{2}\sum_{k=1}^{q}(t_k - z_k)^2\right] \\
&= -\sum_{k=1}^{q}(t_k - z_k)\tfrac{\partial z_k}{\partial y_j} \\
&= -\sum_{k=1}^{q}(t_k - z_k)\tfrac{\partial z_k}{\partial net_k}\tfrac{\partial net_k}{\partial y_j} \\
&= -\sum_{k=1}^{q}(t_k - z_k)\varphi_2{}'(net_k)w_{jk}
\end{aligned} \tag{14}$$

So

$$\Delta w_{ij} = \eta\left[\sum_{k=1}^{q} w_{jk}\delta_k\right]\varphi_1{}'(net_j)x_i \tag{15}$$

Same reason

$$\Delta\theta_j = \eta\left[\sum_{k=1}^{q} w_{jk}\delta_k\right]\varphi_1{}'(net_j) \tag{16}$$

Where net_j is the weighted sum of the input x_j for the hidden layer neuron j, and net_k is the weighted sum of the input y_j for the output layer neuron k.

4. Artificial Bee Colony Algorithm

The artificial bee colony algorithm (ABC) is an algorithm that simulates honey bee collective behavior. Its role is divided into employed bee, onlooker bee, and scout bee. Assume that in the D dimensional space, the population size is $2 \times SN$, SN is the number of honey sources (employed bee number = onlooker bee number = SN). The honey sources and the employed bees correspond one to one, that is to say, the number of honey sources is SN, and the position of the $i\mathrm{th}$ source of honey is denoted as x_i. Each honey source location represents a candidate solution to the optimization problem, and the fitness of the honey source reflects the quality of the solution.

4.1. Four Phases of Artificial Bee Colony Algorithm

Initialization phase: At the beginning of the ABC, the initial food sources are generated randomly according to Eq.(17).

$$X_{i,j} = X_j^L + rand_j(X_j^U - X_j^L) \tag{17}$$

Where $i = \{1, 2, \ldots, SN\}$, $j = \{1, 2, \ldots, D\}$, X_j^L and X_j^U are the upper and lower bounds of the $j\mathrm{th}$ variable respectively. $rand_j$ is a random value in the range of [0,1]. Then, the fitness value of the food sources are calculated by Eq.(18).

$$fit_i = \begin{cases} \frac{1}{1+f(X_i)} & f(X_i) \geq 0 \\ 1 + |f(X_i)| & f(X_i) < 0 \end{cases} \tag{18}$$

Where fit_i is the fitness value of the food source X_i, $f(X_i)$ is the objective function value of food source X_i for the optimization problem. In addition, parameter $limit$ should be determined. The parameter $counter$ records the number of unsuccessful updates, is initialized to 0 for each food source.

Employed bee phase: Each Employed bee will fly to a distinct food source, looking for a candidate food source around the corresponding food source by using Eq.(19).

$$V_{i,j} = X_{i,j} + \phi_{i,j} \times (X_{i,j} - X_{k,j}) \tag{19}$$

Where i and k are randomly selected from $\{1, 2, \ldots, SN\}$, j is randomly selected from $\{1, 2, \ldots, D\}$, $V_{i,j}$ is the $j\mathrm{th}$ dimension of the $i\mathrm{th}$ candidate food source; $X_{k,j}$ is the $j\mathrm{th}$ dimension of the $k\mathrm{th}$ food source, $\phi_{i,j}$ is a random real number in the range of [-1,1].

After finding a new food source, the fitness value of the candidate food source is calculated by Eq.(18). If the candidate food source has a fitness value superior to the old food source, the food source replaces the old food source and the corresponding employed bee memorizes a new food source, and the $counter$ variable of this food source is reset to 0. Otherwise $counter$ is increased by 1.

Onlooker bee phase: According to the quality information of the food source shared by the employed bees, each onlooker bee will fly to a food source X_s, which is selected by the roulette wheel, in order to find a candidate food source according to Eq.(19). The probability of choosing the $i\mathrm{th}$ food source is calculated by Eq.(20).

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{20}$$

Obviously, the value of fitness is greater, the selection probability is higher. If a candidate food source V_s obtained by the onlooker bee is superior to the food source X_s, X_s will be replaced by the new food source. And the $counter$ variable of the food source is reset to 0. Otherwise $counter$ of the food source is increased by 1.

Scout bee phase: A food source with a highest $counter$ value is selected and compared with a predefined $limit$ value. If the value is greater than the value of $limit$, the selected food source is abandoned by its employed bee, and the employed bee is converted to a scout bee, and then this scout bee randomly finds a new source of food according to Eq.(17). After acquiring a new food source, the corresponding $counter$ is reset to 0 and the scout bee is changed to employed bee. Note that if the $j\mathrm{th}$ variable $V_{i,j}$ of the $i\mathrm{th}$ candidate food source violates the upper and lower bound constraints in the employed bee phase and the onlooker bee phase, the food source is reset according to Eq.(17).

4.2. The Improved ABC_elite Algorithm

ABC algorithm is a population-based, iteration-based and stochastic optimization algorithm, and it is very effective for solving numerical optimization problems. However, although the search equations are stronger in exploration, insufficient in exploitation. And the convergence performance of the ABC algorithm is not prominent. So, in order to better balance exploration and exploitation, Ref. [2] proposed a Depth First Search (DFS) framework and two search equations based on elite solutions, as shown in Eq.(21) and Eq.(22). This method named DFSABC_elite. The DFS framework can prioritize more computing resources to better solutions to enhance the exploitation ability of the algorithm. The elite-guided search equations keep the solution with the highest fitness value in the iteration, which speeds up the training process of the algorithm.

$$V_{i,j} = X_{e,j} + \phi_{i,j} \times (X_{e,j} - X_{k,j}) \tag{21}$$

$$V_{e,j} = \frac{1}{2}(X_{e,j} + X_{best,j}) + \phi_{e,j} \times (X_{best,j} - X_{k,j}) \tag{22}$$

Where X_e is a randomly selected solution from the elite solution, and X_k is a randomly selected solution from the current population. e is not equal to k, and k is not equal to i. X_{best} is the current best solution. $\phi_{i,j}$,$\phi_{e,j}$ are two random real numbers in the range of [-1, 1].

In order to better balance ABC exploration and exploitation capabilities, Ref. [6] addresses the problem that the candidate solution search equation has too large a disturbance to the elite solution in the search equation proposed in Ref. [2], and proposes an elite search equation. The candidate solutions for solutions and ordinary solutions should use different search equations. Ref. [17] proposed a novel search equation in Particle Swarm Optimization (PSO) algorithm, as shown in Eq.(23).

$$P_i = \frac{c_1 \times pbest_i + c_2 \times gbest}{c_1 + c_2} \tag{23}$$

Where c_1 and c_2 are two learning coefficients, $pbest$ is the personal best position, and $gbest$ is the population best solution found so far. Based on Eq.(23), a novel equation is proposed in Ref.[19].

$$X_i = N\left(\frac{gbest + pbest_i}{2}, |gbest - pbest_i|\right) \tag{24}$$

Where N represents the Gaussian distribution, $\frac{gbest+pbest_i}{2}$ represents the mean, and $|gbest - pbest_i|$ represents the standard deviation. The information around $pbest$ and $gbest$ is exploited by using the Gaussian distribution in Eq.(24). A similar Gaussian search equation is proposed according to Eq.(24).

$$V_{i,j} = N\left(\frac{X_{best,j} + X_{i,j}}{2}, |X_{best,j} - X_{i,j}|\right) \tag{25}$$

Where $X_{i,j}$ is the $j\mathrm{th}$ element of the elite solution X_i; $X_{best,j}$ is the $j\mathrm{th}$ element of the global best solution found so far, and j is randomly selected from $\{1, 2, ..., D\}$. According to Eq.(25), the elite solutions in employed bee phase search around X_{best} can improve the exploitation ability of ABC and the success rate of disturbance for elite solutions.

The search equation for the elite candidate solution is shown in Eq.(26).

$$V_{e,j} = \frac{1}{2}(X_{e,j} + X_{best,j}) + \phi_{e,j}(X_{best,j} - X_{e',j}) \tag{26}$$

The elite-guided search equations enhance ABC algorithm ability for exploiting solutions. The DFSABC_elite algorithm is combined with the two novel elite-guided search equations forming a new ABC variant called the Improved ABC_elite Algorithm (IABC_elite).

Define Fes (function evaluation) as the fitness function value, and $\max\_Fes$ (maximal function evaluation) as the maximal fitness function value. The pseudo code of the IABC_elite algorithm as follows:

Algorithm 1 The procedure of IABC_elite

Initialization: Generate SN solutions that contain variables according to Eq.(17);
while $Fes < \max\_Fes$ **do**
    Select the top solutions as elite solutions from populations;
    for $i = 1$ to SN **do**
        //employed bee phase
        if i is an elite solution **then**
            Generate a new candidate solution V_i in the neighborhood of X_i using Eq.(24);
        else
            Generate a new candidate solution V_i in the neighborhood of X_i using Eq.(21);
        end if
        Evaluate the new solution V_i;
        if $f(V_i) < f(X_i)$ **then**
            Replace X_i by V_i;
            counter(i)=0;
        else
            counter(i)=counter(i)+1;
        end if
    end for//end employed bee phase
    for $i = 1$ to SN **do**
        //onlooker bee phase
        Select a solution X_e from elite solutions randomly to search;
        $P_0 = 1 - Fes/\max\_Fes$
        if $rand(0,1) < P_0$ **then**
            Generate a new candidate solution V_e in neighborhood of X_e in neighborhood of X_e using Eq.(22)
        else
            Select a solution $X_{e'}$ from elite solutions randomly, where e' not equal to e;
            Generate a new candidate solution V_e using Eq.(26);
        end if
        Evaluate the new solution V_e;
        if $f(V_e) < f(X_e)$ **then**
            Replace X_e by V_e;
            counter(e)=0;
        else
            counter(e)=counter(e)+1;
        end if
    end for //end onlooker bee phase
    $Fes = Fes + SN * 2$
    Select the solution $X_{\max}$ with max counter value; //Scout bee phase
    if $counter(\max) > limit$ **then**
        Replace $X_{\max}$ by a new solution generated according to Eq.(17);
        $Fes = Fes + 1$,$counter(\max) = 0$
    end if//end scout bee phase
end while

5. BP Neural Networks Based on IABC_elite Algorithm

The Backprogation (BP) neural network based on IABC_elite algorithm is the combination of improved bee colony algorithm and BP neural networks algorithm. A new BP neural network optimized by improved ABC_elite algorithm is proposed and applied to intrusion detection. The research results of numerous papers show that the BP neural networks are sensitive to initial parameters [25,29,30]. Although there are many papers use particle swarm optimization (PSO), genetic algorithm (GA) and other algorithms to optimize the initial parameters of BP neural networks, the IABC_elite algorithm better balance the exploration and exploitation ability of the solution. With the strong ability to jump out of the local optimal solution, so using IABC_elite to optimize BP neural network can further improve the performance of BP neural networks. The optimized objection for IABC_elite algorithm are the weight values and threshold values of the neural networks. This method can improve the self-learning ability of BP neural network and speed up the convergence process of Backpropagation algorithm, so the neural network intrusion detection model have better detection ability. Its key techniques lie in:

Use the IABC_elite algorithm to train the neural network to generate the initial weight values and threshold values of the neural networks and use the error function of the neural networks as the fitness objective function of the IABC_elite algorithm.

The food source position in the IABC_elite algorithm represents the weight values and threshold values in the BP neural networks. The fitness value of each food source is calculated through the fitness function, which is the error function of the neural networks. The food source has a lower error value, has a higher fitness value. The IABC_elite algorithm tends to select the higher fitness value food source. The IABC_elite algorithm finds the honey source of the best fitness value in the search process. The best honey sources are obtained by comparing the current fitness value and historical value of each honey source within the process of iterations. The best honey source will be set as the initial weight values and threshold values of the BP neural network. When the error function value of the neural network generated by the best honey source meets the error precision requirement, the backpropagation algorithm performs further training for the neural network and tries to minimize the training error value. The algorithm avoids the neural network falling into a local optimum, achieves the goal of improving the accuracy and speeds up the neural network training. The basic idea of using IABC_elite algorithm to optimize BP neural network as follows:

First, we construct a neural network model, set the number of nodes in the input layer and the number of neurons in the hidden layer and the output layer. And construct the connection between the input layer and the hidden layer, as well as the hidden layer and the output layer. The activation function of the hidden layer neuron is set as Sigmoid function. The function equation is shown in Eq.(27), and the Sigmoid function curve is shown in Fig. 3. The Sigmoid function curve is S, which can map the output signal value of the linear combiner to the range of [0, 1].

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{27}$$

The activation function of the output layer neuron is set as Softmax function. This function is usually used in multi-classification process. It maps the output signal of multiple neurons to the range of [0, 1] and its equation is shown in Eq.(28).

$$Softmax(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad j = 1, 2, \ldots, K \tag{28}$$

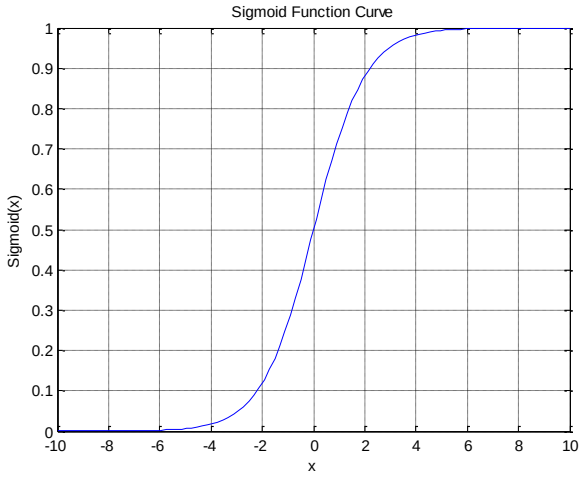


Fig. 3. Sigmoid Function Curve

Second, the IABC_elite algorithm is used to train the neural network and generate initial weight values and threshold values for the neural network. Set the population size of the IABC_elite algorithm and set the dimension of the solution according to the number of variables in the threshold values and weight values of the neural network. The position of the food source represents the weight values and threshold values of the neural network. The neural network error function is used as the food source fitness value function. High fitness food sources are those food sources with high fitness weight values and threshold values. The IABC_elite algorithm obtains a better solution by updating the honey sources during the iterative process. Table 1 show the mapping relation between neural networks training parameter and honey bee collective behavior. The IABC_elite algorithm choose the best fitness food source as the initial weight values and threshold values of the neural network.

Table 1. Mapping relation between neural networks training parameter and honey bee collective behavior

honey bee collective behavior	**neural networks training parameter**
Food source location	Weight values and threshold values
Food source fitness	Neural network error function value
High fitness food source	High fitness weight values and threshold values

Finally, the backpropagation algorithm trains the neural network according to the initial weight values and threshold values generated by IABC_elite algorithm. The backpropagation algorithm tries to minimize the training error by gradient descent. The weights and thresholds with the smallest training error will be used as parameters of the neural network for intrusion detection of network traffic.

The detail algorithm process is described as follows:

1) Select sample data for training, and randomly generate the weight values $\mathbf{w}_{ij}$ between the input layer neurons and the hidden layer neuron, the weight values $\mathbf{w}_{jk}$ between the hidden layer neurons and the output layer neurons. As well as generate the threshold values θ_j of the hidden layer neurons and the threshold values θ_k of the output layer neurons.

2) The output values of neural networks output layer neurons are calculated according to Eq.(29), Eq.(30), Eq.(31), and Eq.(32).

$$net_j = \sum_{i=1}^{m} w_{ij} x_i + \theta_j \tag{29}$$

$$y_j = \varphi_1(net_j) \tag{30}$$

$$net_k = \sum_{j=1}^{h} w_{jk} y_j + \theta_k \tag{31}$$

$$z_k = \varphi_2(net_k) \tag{32}$$

3) The error of the neural network is calculated according to Eq.(33). If it meets the required error, end training and go to step 5); otherwise go to step 4).

$$J(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{q} (t_k - z_k)^2 \tag{33}$$

4) The weight values and threshold values between the hidden layer and the output layer are adjusted according to Eq.(34) and Eq.(35). The weight values and threshold values between the input layer and the hidden layer are adjusted according to Eq.(36) and Eq.(37).

$$\Delta w_{jk} = \eta (t_k - z_k) \varphi_2{}'(net_k) y_j \tag{34}$$

$$\Delta \theta_k = \eta (t_k - z_k) \varphi_2{}'(net_k) \tag{35}$$

$$\Delta w_{ij} = \eta \left[\sum_{k=1}^{q} w_{jk} \delta_k \right] \varphi_1{}'(net_j) x_i \tag{36}$$

$$\Delta \theta_j = \eta \left[\sum_{k=1}^{q} w_{jk} \delta_k \right] \varphi_1{}'(net_j) \tag{37}$$

Where

$$\delta_k = -\frac{\partial J}{\partial net_k} = -\frac{\partial J}{\partial z_k}\frac{\partial z_k}{\partial net_k} = (t_k - z_k)\varphi_2{}'(net_k) \tag{38}$$

5) From the results obtained in step 4), we can get the new weight values $\mathbf{w}_{ij}$ and the new threshold values θ_j between the input layer and the hidden layer, as well as the new weight values $\mathbf{w}_{jk}$ and the new threshold values θ_k between the hidden layer and the output layer.

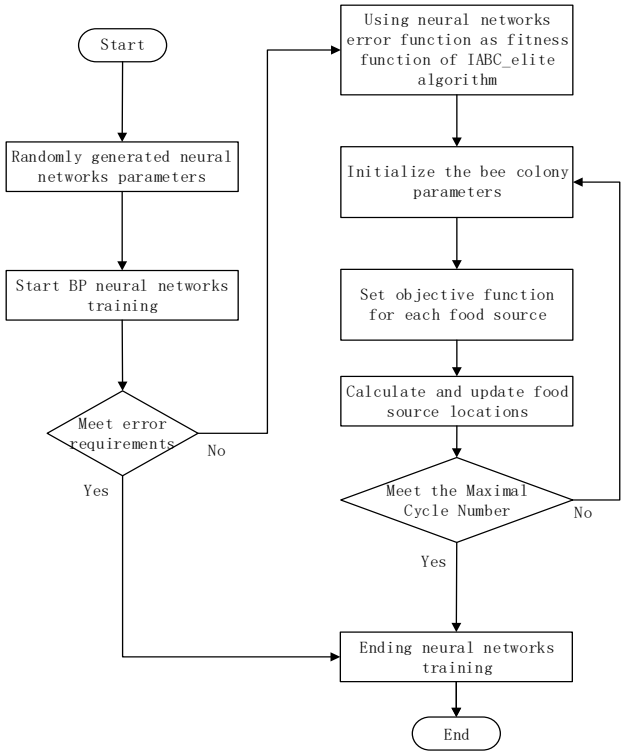


Fig. 4. Flowchart of IABC_elite algorithm to optimize BP neural networks

6) According to the new weight values and threshold values, re-execute step 1) to 3). If the error meets the requirements, end the training process. Otherwise, continue.

7) Using the current weights and thresholds as neural work input signal, and get the corresponding neural network output signal. The neural network loss function is set as the objective function of Eq. (39). Set the value of the Max Cycle Number (MCN).

$$fit_i = \begin{cases} \frac{1}{1+f(X_i)} & f(X_i) \geq 0 \\ 1 + |f(X_i)| & f(X_i) < 0 \end{cases} \tag{39}$$

8) Run the IABC_elite algorithm until iterations reaches the maximum number of iterations MCN. Set the weight values and threshold values from the IABC_elite algorithm as new initial parameters to train the neural network, then go to step 4).

9) End the training process; output the weight values $\mathbf{w}_{ij}$ and the threshold values θ_j as well as the weight values $\mathbf{w}_{jk}$ and the threshold values θ_k. Use the obtained neural networks model for testing data to predict the classification.

Table 2. Features of NSL-KDD datasets

No.	Features	Types	No.	Features	Types
1	duration	Continuous	22	is_guest_login	Symbolic
2	protocol_type	Symbolic	23	count	Continuous
3	service	Symbolic	24	srv_count	Continuous
4	flag	Symbolic	25	serror_rate	Continuous
5	src_bytes	Continuous	26	srv_serror_rate	Continuous
6	dst_types	Continuous	27	rerror_rate	Continuous
7	land	Symbolic	28	srv_rerror_rate	Continuous
8	wrong_fragment	Continuous	29	same_srv_rate	Continuous
9	urgent	Continuous	30	diff_srv_rate	Continuous
10	hot	Continuous	31	srv_diff_host_rate	Continuous
11	num_failed_logins	Continuous	32	dst_host_count	Continuous
12	logged_in	Symbolic	33	dst_host_srv_count	Continuous
13	num_compromised	Continuous	34	dst_host_same_srv_rate	Continuous
14	root_shell	Continuous	35	dst_host_diff_srv_rate	Continuous
15	su_attempted	Continuous	36	dst_host_same_src_port_rate	Continuous
16	num_root	Continuous	37	dst_host_srv_diff_host_rate	Continuous
17	num_file_creations	Continuous	38	dst_host_serror_rate	Continuous
18	num_shells	Continuous	39	dst_host_srv_serror_rate	Continuous
19	num_access_files	Continuous	40	dst_host_rerror_rate	Continuous
20	num_outbound_cmds	Continuous	41	dst_host_srv_rerror_rate	Continuous
21	is_host_login	Symbolic			

6. Experiments and Analysis

6.1. Dataset Description

Prof. Sal Stolfo from Columbia University and Prof. Wenke Lee from North Carolina State University used data mining and other techniques to perform feature analysis and data preprocessing on DARPA 98 and DARPA 99 datasets, which form a new dataset. This dataset was used in the KDD CUP competition held in 1999 and is the benchmark dataset for intrusion detection. The dataset is named KDD CUP 99 dataset. But the KDD CUP 99 dataset has some shortcomings. For the shortcomings of the KDD CUP 99 dataset, the NSL-KDD data set removes the redundant data in the KDD CUP 99 dataset, overcoming the problem that the classifier is biased towards recurring records, and the performance of the learning method is affected [26]. In addition, the proper selection of normal

and abnormal network data ratios makes the testing and training data more reasonable, making it more suitable for efficient evaluation of different machine learning techniques. Table 2 shows the features of the NSL-KDD dataset. NSL-KDD dataset contains training dataset KDDTrain+.txt, KDDTrain+_20Percent.txt. KDDTrain+_20Percent is 20% of KDDTrain+ data. NSL-KDD dataset contains testing dataset KDDTest+.txt, KDDTest-21.txt. The test accuracy is generally less than KDDTest+ on KDDTest-21. The number of samples of different classifications of the NSL-KDD dataset shows in Table 3.

Table 3. Different Classifications of NSL-KDD Dataset

	KDDTrain+	**KDDTest-21**
Total	125973	11850
Normal	67343	2152
DoS	45927	4342
Probe	11656	2402
R2L	995	2754
U2R	52	200

6.2. Data Preprocessing

1) Vectorization

The NSL-KDD dataset has 34 numeric features and 7 symbol features. The features of 2, 3, 4, 7, 12, 21, and 22 columns are the symbol features, where the features of 7, 12, 21, and 22 columns are '0' or '1', and the features of 2, 3, and 4 columns are the character type. Because the input value of the neural networks should be a numeric matrix, we must convert some non-numeric features into numeric form.

Generally in machine learning, if there is an 'order' relationship between the feature values for symbol features, they can be converted to continuous values. For example, the value of the binary feature 'height' is 'high' and 'low', it can be converted to 1.0, 0.0. The value of the 3-value feature 'height' is 'high', 'middle' and 'low', it can be converted to 1.0, 0.5, 0.0; if there is no 'order' relationship between the feature values, the features values are usually converted into dimension vectors. For example, the value of the feature 'melon' is 'watermelon', 'pumpkin' and 'cucumber', it can be converted into (1,0,0), (0,1,0), (0, 0,1).

Thus, for example, the feature 'protocol_type' has three types of attributes, 'tcp','udp' and 'icmp'. It can be converted into (1, 0, 0), (0, 1, 0), (0, 0, 1).

2) Normalization

The normalization is to scale the data, which make it fall into a small specific range. In this paper, the data is uniformly mapped to the range of [-1, 1]. The normalization formula is shown in Eq. (40).

$$y = (ymax - ymin) * \frac{x - xmin}{xmax - xmin} + ymin \tag{40}$$

6.3. Evaluation Metrics

In the intrusion detection system, the most important performance measure is the accuracy of the detection, which means the classification ability of the neural network model. In order to evaluate the accuracy of the experiment, we introduce the concept of detection rate and false positive rate in the confusion matrix. The True Positive (TP) refers to the number of samples that the abnormal sample is identified as abnormal. The False Positive (FP) refers to the number of samples that the normal sample is identified as abnormal. The True Negative (TN) refers to the normal number of samples that the normal sample is identified as normal. The False Negative (FN) refers to the number of samples that the abnormal sample is identified as normal. Table 4 shows the definition of the confusion matrix. We have the following notation:

Accuracy (AC) is the percentage of the number of samples identified correctly over the total number of samples, as shown in Eq. (41).

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \tag{41}$$

True Positive Rate (TPR) is the percentage of the number of anomaly samples correctly identified over the total number of anomaly samples, equal to the detection rate (DR), as shown in Eq.(42).

$$TPR = \frac{TP}{TP + FN} \tag{42}$$

False Positive Rate (FPR) is the percentage of the number of normal samples erroneously identified as anomaly samples over the total number of normal samples, as shown in Eq.(43).

$$FPR = \frac{FP}{FP + TN} \tag{43}$$

Therefore, the goal of the intrusion detection system is to obtain a higher accuracy and detection rate with a lower false positive rate.

Table 4. Confusion matrix

Actual Class \ Predicted Class	**anomaly**	**normal**
anomaly	TP	FN
normal	FP	TN

6.4. Experiment Results and Discussion

The experimental software environment for this paper is MATLAB 2014a. The experiment is performed on ThinkServer RD550, which has a configuration of two Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz and 16.0GB memory. There are five types of network packets in the NSL-KDD dataset. Therefore, we conducted experiments in five

categories: Normal, DoS, R2L, U2R, and Probe. In this experiment, KDDTrain+ is selected as the training dataset in NSL-KDD, and KDDTest-21 is selected as the testing dataset.

In Section 6.2, we map 41-dimensional features to 122 dimensions, so the input layer of the neural network has 122 nodes. The output result is 5 categories, so the output layer of the neural network has 5 nodes. We set the number of the hidden layer nodes of the neural network as 80 nodes. At this point, the neural networks need 10245 parameters to be optimized, so the solution in IABC_elite has a dimension of 10245. We set the population size of the bee colony as 100, the upper bound of the solution as 0.1, the lower bound as -0.1, and the maximum number of iterations of the IABC_elite algorithm as 100.

We select KDDTrain+ as the training dataset of BP neural network, and randomly take 30% of the data from it as a validation to avoid the BP neural network falling into overfitting. The cross-entropy is used as the error cost function of the backpropagation algorithm. In information theory, the cross entropy between two probability distributions over the same underlying set of events measures the average of bits needed to identify an event drawn from the set, if a coding scheme is used that is optimized for an 'unnatural' probability distribution, rather than the 'true' distribution. The average number of bits required for an event is also commonly used in error learning in machine learning. In order to speed up the neural network training process, the neural network is trained using the scaled conjugate gradient method [20].

Fig. 5 shows the BPNN cross entropy error iteration curve for randomly generating neural network initial parameters for the BP neural network training. Fig. 6 shows the ABC_BPNN cross entropy error iteration curve for the neural network training using the ABC algorithm pre-training parameters as initial parameters of the neural network. Fig. 7 shows the IABC_elite_BPNN cross entropy error iteration curve for the BP neural network training using the IABC_elite algorithm pre-training parameters as initial parameters of the neural network. The experimental results show that the initial parameters generated by the pre-training of the IABC_elite algorithm have lower initial errors, and the number of training epochs is smaller, which saves the time of neural network training.

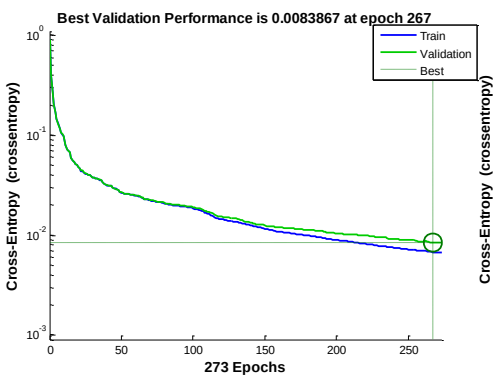


Fig. 5. BPNN Error Cost Iteration Curve

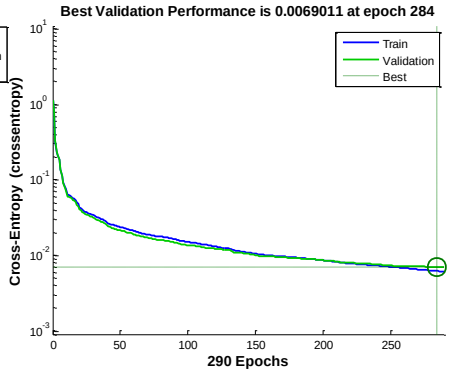


Fig. 6. ABC_BPNN Error Cost Iteration Curve

We test the neural network model on the KDDTest-21 testing dataset and use the five-classification confusion matrix to present the classification results. The class of labeling 1 in confusion matrix figure indicates the Normal class, the class of labeling 2 indicates the DoS class and the class of labeling 3 indicates the Probe class, the class of labeling 4 indicates the R2L class, and the class of labeling 5 indicates the U2R class. Fig. 8 is the classification result of five-classification confusion matrix of the neural network trained by a single BP algorithm. Fig. 9 is the classification result of five-classification confusion matrix of the neural network trained by BP algorithm optimized by ABC algorithm. Fig. 10 is the classification result of five-classification confusion matrix of the neural network trained by BP algorithm optimized by IABC_elite algorithm. Overall, the total prediction accuracy in Fig. 9 is 1.4% higher than Fig. 8. The total prediction accuracy in Fig. 10 is 2.8% higher than Fig. 9. In the view of the Normal class, the classification accuracy predicted by the BP neural network optimized by IABC_elite algorithm has been greatly improved, compared with the previous two figures. In the view of the five-classification confusion matrix, the BP neural networks optimized by IABC_elite algorithm have better generalization ability.

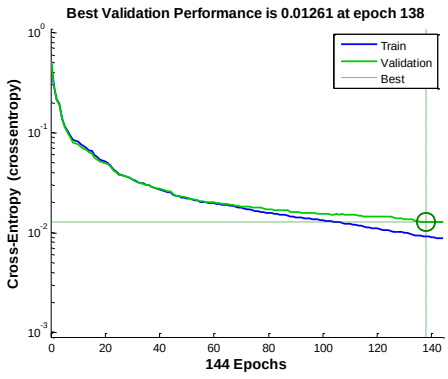


Fig. 7. IABC_elite_BPNN Error Cost Iteration Curve

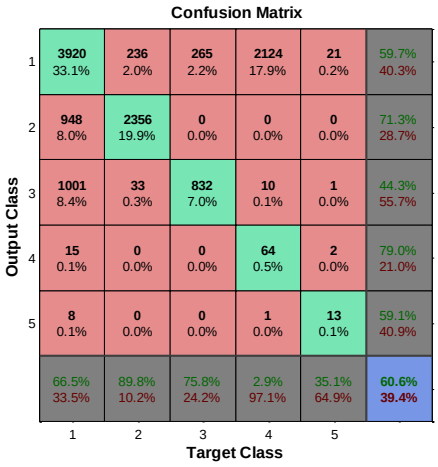


Fig. 8. BPNN Five-Classification Confusion Matrix

Therefore, in the training phase of the neural network, the BP neural network optimized by IABC_elite reduces the initial error of neural network training, requires less epoch training times, and save time of the neural network training. In the testing phase of the neural network, the BP neural network optimized by IABC_elite has better generalization ability, higher test accuracy and lower false positive rate, and better classification ability for normal and abnormal samples. This experiment results show that the initial parameter of the neural network generated by IABC_elite is better than the initial parameter generated by ABC and the randomly generated initial parameters. The initial weight values and threshold values generated by IABC_elite make it convenient to search

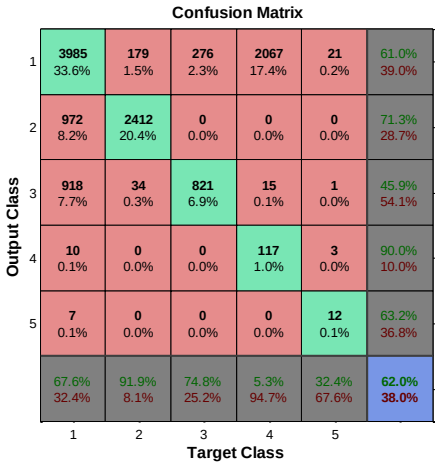


Fig. 9. ABC_BPNN Five-Classification Confusion Matrix

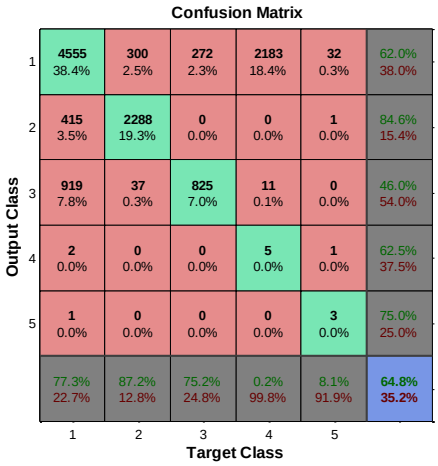


Fig. 10. IABC_elite_BPNN Five-Classification Confusion Matrix

the optimum weight values and threshold values for Backpropagation algorithm. It make the Backpropagation algorithm find a better solution which have a better generalization performance in testing phase of the neural network. The IABC_elite improved the convergence speed of the BP neural networks training and reduce the restrictions on initial weight values and threshold values, and avoid the neural network falling into a local optimum. The simulation verify the effectiveness of intrusion detection systems based on IABC_elite and BP neural networks.

7. Conclusion

This paper proposes a BP neural network model optimized by improved ABC_elite algorithm. The IABC_elite algorithm based on the depth first search framework better balances the ability of the exploration and exploitation of the artificial bee colony algorithm. It introduces the concept of elite solutions, uses two novel elite-guided search equations, and keep the highest fitness solutions. The highest fitness solutions accelerates the search process for optimal solutions. On the one hand, the IABC_elite improves the convergence speed of the BP neural network training; on the other hand, it reduces the restrictions to initial weight values and threshold values on the BP neural network training, and improves the robustness of the algorithm. We apply the neural network to intrusion detection, perform an experiment with NSL-KDD dataset, and create corresponding neural networks model according to the features of the NSL-KDD dataset. We select KDDTrain+ as training dataset from the NSL-KDD dataset to train the neural network, select KDDTest-21 as testing dataset to test the neural network. The confusion matrix is used to present the classification results, which shows that the neural network has better classification ability and verifies the effectiveness of the algorithm. The intrusion detection method IABC_elite and BP neural networks has a high detection rate and low false positive rate.

Acknowledgments. This work has been supported by the National Natural Science Foundation of China (No. 61672338 and No. 61873160).

References

1. Bullinaria, J.A., AlYahya, K.: Artificial bee colony training of neural networks. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2013), pp. 191–201. Springer (2014)
2. Cui, L., Li, G., Lin, Q., Du, Z., Gao, W., Chen, J., Lu, N.: A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. Information Sciences 367, 1012–1044 (2016)
3. Cui, M., Han, D., Wang, J.: An efficient and safe road condition monitoring authentication scheme based on fog computing. IEEE Internet of Things Journal pp. 1–1 (2019)
4. Deng, Y.: What is the future of disk drives, death or rebirth? ACM Computing Surveys (CSUR) 43(3), 23 (2011)
5. Denning, D.E.: An intrusion-detection model. IEEE Transactions on software engineering 13(2), 222–232 (1987)
6. Du, Z., Liu, G., Bi, K., Jia, J., Han, D.: An improved artificial bee colony algorithm with elite-guided search equations. Computer Science & Information Systems 14(3) (2017)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification. John Wiley & Sons (2012)
8. Han, D., Bi, K., B., X., L., H., Wang, R.: An anomaly detection on the application-layer-based qos in the cloud storage system. Computer Science & Information Systems 13(2) (2016)
9. Han, D., Bi, K., Liu, H., Jia, J.: A ddos attack detection system based on spark framework. Computer Science & Information Systems 14(3) (2017)
10. Jaggi, R., Sangade, J.: Detecting and classifying attacks using artificial neural network. Int. J. Recent Innov. Trends Comput. Commun 2(5), 1136–1142 (2014)
11. Jiang, J., Wang, Z., Chen, T., Zhu, C., Chen, B.: Adaptive ap clustering algorithm and its application on intrusion detection. Journal on Communications 36, 118–126 (2015)
12. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, Computer Engineering Department (2005)
13. Karaboga, D., Akay, B., Ozturk, C.: Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks. In: Modeling Decisions for Artificial Intelligence. pp. 318–329. Springer Berlin Heidelberg (2007)
14. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Journal of Global Optimization 39(3), 459–471 (Nov 2007)
15. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (abc) algorithm. Applied Soft Computing 8(1), 687–697 (2008)
16. Karaboga, D., Ozturk, C.: Neural networks training by artificial bee colony algorithm on pattern classification. Neural Network World 19(3), 279 (2009)
17. Kennedy, J.: Bare bones particle swarms. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706). pp. 80–87. IEEE (2003)
18. Lee, W., Stolfo, S.J., Mok, K.W.: A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No. 99CB36344). pp. 120–132. IEEE (1999)
19. Lim, W.H., Isa, N.A.M.: An adaptive two-layer particle swarm optimization with elitist learning strategy. Information Sciences 273, 49–72 (2014)
20. Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural networks 6(4), 525–533 (1993)

21. Ozturk, C., Karaboga, D.: Hybrid artificial bee colony algorithm for neural network training. In: 2011 IEEE congress of evolutionary computation (CEC). pp. 84–88. IEEE (2011)
22. Qian, Q., Cai, J., Zhang, R.: Intrusion detection based on neural networks and artificial bee colony algorithm. In: 2014 IEEE/ACIS 13th International Conference on Computer and Information Science (ICIS). pp. 257–262. IEEE (2014)
23. Qiu, C., Shan, J., Shandong, B., et al.: Research on intrusion detection algorithm based on bp neural network. International Journal of Security and its Applications 9(4), 247–258 (2015)
24. Revathi, S., Malathi, A.: A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT) 2(12), 1848–1853 (2013)
25. SHEN Xiajiong, WANG Long, H.D.: Application of bp neural network optimized by artificial bee colony in intrusion detection. Computer Engineering 42(2), 190 (2016)
26. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. pp. 1–6. IEEE (2009)
27. Wang, H., Wu, Z., Rahnamayan, S., Sun, H., Liu, Y., Pan, J.s.: Multi-strategy ensemble artificial bee colony algorithm. Information Sciences 279, 587–603 (2014)
28. Xie, J., Deng, Y., Min, G., Zhou, Y.: An incrementally scalable and cost-efficient interconnection structure for data centers. IEEE Transactions on Parallel and Distributed Systems 28(6), 1578–1592 (2017)
29. Yao, X.: Evolutionary artificial neural networks. International journal of neural systems 4(03), 203–222 (1993)
30. Yao, X., Islam, M.M.: Evolving artificial neural network ensembles. IEEE Computational Intelligence Magazine 3(1), 31–42 (2008)
31. Yin, C., Zhu, Y., Fei, J., He, X.: A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5, 21954–21961 (2017)
32. Zhu, G., Kwong, S.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied mathematics and computation 217(7), 3166–3173 (2010)

Letian Duan is currently a master student of Shanghai Maritime University. His research interests include cloud computing and Big Data.

Dezhi Han received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor of computer science and engineering with Shanghai Maritime University. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Qiuting Tian is currently a Ph.D. candidate at Shanghai Maritime University. Her research interests include cloud computing, mobile networking security and cloud security.

Received: October 1, 2018; Accepted: September 10, 2019.

Computer Science and Information Systems 16(3):797–813 https://doi.org/10.2298/CSIS180912027T

Hierarchical Authority Based Weighted Attribute Encryption Scheme

Qiuting Tian, Dezhi Han, and Yanmei Jiang

College of Information Engineering, Shanghai Maritime University
Shanghai, China
{tqt2018, jym2097}@163.com, dzhan@shmtu.edu.cn

Abstract. With the development of cloud storage technology, data storage security has become increasingly serious. Aiming at the problem that existing attribute-based encryption schemes do not consider hierarchical authorities and the weight of attribute. A hierarchical authority based weighted attribute encryption scheme is proposed. This scheme will introduce hierarchical authorities and the weight of attribute into the encryption scheme, so that the authorities have a hierarchical relationship and different attributes have different importance. At the same time, the introduction of the concept of weight makes this scheme more flexible in the cloud storage environment and enables fine-grained access control. In addition, this scheme implements an online/offline encryption mechanism to improve the security of stored data. Security proof and performance analysis show that the scheme is safe and effective, and it can resist collusion attacks by many malicious users and authorization centers. It is more suitable for cloud storage environments than other schemes.

Keywords: cloud storage, attribute-based encryption, weighted attribute, access control, hierarchical authority.

1. Introduction

With the large-scale deployment of cloud storage systems, a large amount of sensitive data is outsourced to cloud storage servers [19,7]. However, the outsourced storage mode of data also brings some security problems. For example, the cloud server can expose the privacy data of the user or the enterprise without the authorization of the user for certain benefits or curiosity. As one of the solutions, the attribute-based encryption (ABE) [20] mechanism emerged as the times require, and has become a hot topic in recent years. Most of the early ABE solutions were for a single authority, and the attribute private keys of all users were distributed by a single authority, which increased the burden on the authority. And in the current cloud environment [13], many scenarios involve massive users and massive data, such as personal health record sharing systems. If us still use a single authority in this scenario, it will cause a system bottleneck. Once the attacker breaks the server, it will bring inevitable losses to the user, and the early encryption method is not applicable to the current distributed computing environment. If it distributes the user's decryption key through multiple authorities, the resulting loss may be less. In practical applications, multi-authority can process user attributes quickly and efficiently, and manage the user's private key. To solve the problems caused by a single authority, Chase [5]

first proposed the ABE scheme of the multi-authority, but the scheme only supports the basic ABE algorithm and lacks the flexibility of access. In [6], in view of the hidden dangers of the existence of central authority (CA), an improved CA-free multi-authority ABE scheme is proposed. Yang et al. [24] proposed an effective data access control for multi-authority cloud storage systems, which outsources the decrypted main calculations to the cloud server. Gorasia et al. [12] proposed a multi-authority ABE scheme for fast decryption algorithm, but did not give corresponding security proof.

Since the disadvantage of the traditional ABE mechanism is that the attribute structure is not layered, all attributes have the same security level, and it supports only a single assignment of attributes. Such an attribute structure makes it difficult to have flexible, fine-grained access. Although there are many multi-authority based on attribute access control, most of these solutions do not consider the weight of attributes, that is to say, attributes are equal. But in practical applications, the attribute with weight has practical significance.

Aiming at the above problems, this paper designs a flexible and efficient hierarchical authority based weighted attribute encryption scheme. Our research contributions are summarized as follows:

1) This method uses a layered certification authority to distribute private keys for users. On the one hand, it avoids problems such as server load operation caused by a single certification authority; on the other hand, it facilitates management by trusted authorities. It can achieve more efficient hierarchical management between authorities.

2) The method introduces attribute weights into the encryption scheme, so that the highest weights of the attribute private keys distributed by the authorities of different levels to the users are different, thereby achieving fine-grained access control. By splitting the attributes into different sets, each level of authority selects the appropriate subset based on the weights and distributes them to the lower authority or user. This scheme is suitable for large authorities, which is convenient for managing users and authorities at each level, and also improves system security.

3) This method divides the encryption process into two phases, online and offline. The offline phase pre-processes complex calculations, while the online phase requires only a small amount of simple calculations. The online and offline double-layer encryption greatly improves the security of the entire encryption phase, and is suitable for cloud computing users with limited computing power.

The remaining of the paper is organized as follows. In Section 2, we introduce the research results of the other researchers, related to our research. Some preliminaries are given in Section 3. In Section 4, we propose the scheme formalization and security model. Next, the corresponding specific algorithm is presented. Section 6 presents the security analysis. Section 7 gives the performance analysis of our scheme. In the final section, the conclusion is given.

2. Related Work

The distribution of a large number of users in real life may be in a large hierarchical authority, and there may be a relationship between authorities when distributing private keys to users. At this time, it is necessary to consider how to achieve hierarchical access between authorities. Ref. [22] considered the hierarchy of the authority and proposed a

hierarchical ABE scheme, but in this scheme, there is no hierarchical difference in system attributes. Ref. [21] proposed a layered attribute set encryption scheme, which can achieve scalability through a layered structure. Ref. [27] began to study the attribute set and subset, and realized the stratification between attributes. The only drawback of these two schemes is that the connection between the authorities has not yet been given. If there is no intersection between the user's attribute sets, the application in the real scene is restricted. Subsequently, Ref. [17] proposed a hierarchical multi-authority and attribute-based encryption scheme, it employs character attribute subsets to achieve flexible fine-grained access control. In the system, trusted authority manages hierarchical attribute authorities, but must ensure that the trusted authority is absolutely trustworthy and that it is limited by the size of the attribute set. Ref. [1] gave the encryption scheme of the hierarchical authority, and the scheme does not consider the problem of attribute weights. In practical applications, each attribute has a different role and status, and the attribute has a practical value. In this scenario, the status of each attribute in the system is equal. Therefore, the concept of weights introduced into attributes is more suitable for the needs of practical systems.

Subsequently, Ref. [15] proposed a ciphertext-policy weighted ABE scheme, the attributes have different weights according to their importance. Ref. [23] proposed a multi-authority based weighted attribute encryption scheme. The attribute authorities assign different weights to attributes according to their importance. Ref. [9] proposed a weighted multi-authority attribute encryption based on ciphertext policy, which implements the encryption mechanism without a trusted center. However, in this scheme, the user's attribute private key and system key are calculated by the attribute authorization center, which causes a load of a single authorization center to be too large, and may eventually cause the attribute authority to crash. Although these schemes introduce the concept of weights, they do not consider the hierarchical relationship between the authorities. Ref. [8] proposed a multi-authority and hierarchical weighted attribute-based encryption scheme, which uses different levels of authorities to distribute different weighted attribute private keys. Thereby, finer-grained access control can be achieved, but the computational overhead of the scheme is large.

3. Preliminaries

3.1. Bilinear Pairing Map [4]

Let G_1 and G_2 be two multiplicative cyclic groups with the prime order P, and set the generator of group G_1 to be g. There exists a bilinear map e: $G_1 \times G_1 \rightarrow G_2$ that satisfies the following properties:

(1) Bilinearity: for all $u, v \in G_1$, $a, b \in Z_p$, there is $e(u^a, v^b) = e(u, v)^{ab}$.

(2) Non-degeneracy: for any $g \in G$, there is $e(g, g) \neq 1$.

(3) Computability: for all $u, v \in G_1$, there is an effective algorithm for calculating $e(u, v) \in G_2$.

3.2. Access Structure [8]

Let $\{p_1, p_2, ..., p_n\}$ denote a set of parties, set $P = 2^{\{p_1, p_2, ..., p_n\}}$. Access structure A is a non-empty subset of $\{p_1, p_2, ..., p_n\}$, i.e. $A \subseteq P \backslash \{\phi\}$. If the access structure A is

monotonic, then for $\forall B, C$, if $B \in A$ and $B \subseteq C$, then $C \in A$. The sets in A are the authorized sets, and the sets outside A are unauthorized sets.

3.3. Weighted Threshold Access Structure [2]

Let U be the set of all attributes, weight function is ω: $U \to N$, threshold is $T \in N$, define $\omega(A) = \sum_{u \in A} \omega(u)$ and $\Gamma = \{A \subset U : \omega(A) \geq T\}$. Then Γ is called as a weighted threshold access structure on U.

Each leaf node corresponds to the weight of one attribute, and the root node corresponds to the threshold. The size of the weight must be expressed as a positive integer. As is shown in Fig. 1, an example of a weighted threshold access structure is given. Suppose someone has three attributes: career, age and country. It distributes the corresponding weights of the three attributes in the leaf node. When the threshold value t is less than or equal to the sum of the weights of the three parts in the private key, it can decrypt the ciphertext. If user A's attribute set is $\{\mathrm{Director}, 55, \mathrm{China}\}$, User B's attribute set is $\{\mathrm{Manager}, 38, \mathrm{Australia}\}$. The system assigns weight values $\{7, 1, 1\}$ and $\{9, 4, 1\}$ to the attribute sets of users A and B, respectively. If the system threshold is set to 11, then the sum of the user's attribute weights must be greater than or equal to 11. All attribute weight values of user A add up to 9, which is less than the threshold value and cannot be decrypted. The total weight of the user B adds up to 14, which is greater than the threshold value, and it can decrypt the ciphertext to get the plaintext.

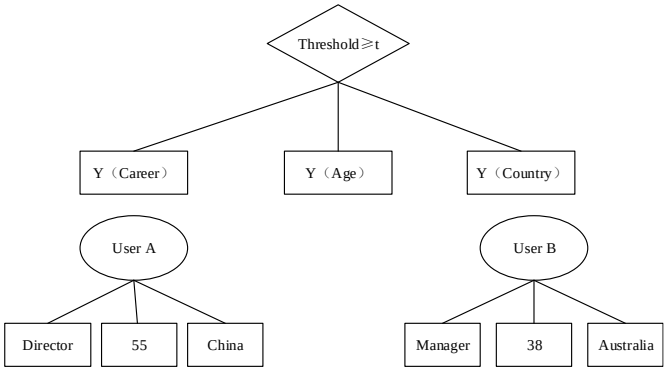


Fig. 1. Weighted threshold access structure

3.4. Attribute Set Split [16]

Split all the attributes in the system. For any attribute λ_i in the attribute set $\Gamma = \{\lambda_1, \lambda_2, ..., \lambda_n\}$, the maximum weight is computed as:

$$\omega_i = weight(\lambda_i) \tag{1}$$

According to the calculation result ω_i, the attribute λ_i is divided into $(\lambda_i, 1), ..., (\lambda_i, \omega_i)$. Assuming that the smallest split unit is 1 and the weights are only integers, the set formed is the split set Γ^* of weighted attributes.

4. Scheme Formalization and Security Model

This paper proposes a hierarchical authority based weighted attribute encryption scheme, which supports layering between authorities, and different attributes have different importance. Different levels of authority can be used to distribute the difference in attribute weights, enabling more fine-grained access control. In addition, because of the complexity of the access policy and the number of attributes in the existing ABE scheme, the overhead of encryption and key generation becomes large. This scheme introduces online/offline encryption technology, which divides the encryption process into an offline phase and an online phase [26]. The offline phase pre-processes complex calculations by the data owner during idle periods. In this way, the online phase requires only a small amount of simple calculations to generate ciphertext. This scheme is mainly composed of the following five parts: Cloud Service Provider (CSP), Trust Authority (TA), Attribute Authority (AA), Data owner (DO) and User.

CSP mainly provides cloud data storage service, stores data uploaded by DO; TA is responsible for generating and distributing system parameters, and manages upper-layer AA; upper-layer AA authorizes the lower-layer AA so that the lower-layer AA has the right to in charge some of the user's attributes. DO encrypts its own data and uploads it to the CSP, which is stored by the CSP; the user sends a file request to the CSP according to his/her needs, downloads the file from the cloud and decrypts it with the user's secret key. The structure of the system is shown in Fig. 2.

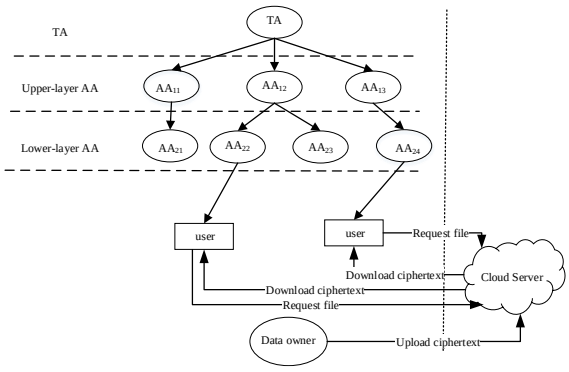


Fig. 2. System architecture of hierarchical authority

In this scheme, CSP is honest and curious. That is, CSP will process cloud data strictly according to the algorithms and protocols in the scheme, but, it will also snoop the owner's data as much as possible. TA is a trusted central authority, we believe TA is trustworthy, and lower-layer AA is semi-trustworthy. The user's attributes are primarily managed by

all AAs in the chain of authorization centers they are in. Suppose the user's attribute set is $A_u = \{A_1, A_2, ..., A_n\}$, and A_u is automatically divided into K disjoint subsets according to the hierarchy.

The divided subset is managed by K AAs on the AA chain and satisfies $\sum_{k=1}^{K} A_u^k = A_u$. As shown in Fig. 2 above, the user belongs to AA_{24}, assuming that all attribute sets owned by the user are: $\{Name : \text{Java}, \text{ID} : 687921, \{\text{Age} : 18, \text{Sex} : \text{Female}, \{\text{Job} : \text{Manager}, \text{Tel} : 1255766\}\}\}$.

Then TA manages attribute $\{Name : \text{Java}, \text{ID} : 687921\}$; A_{13} manages attribute $\{\text{Age} : 18, \text{Sex} : \text{Female}\}$; and A_{24} manages attribute $\{\text{Job} : \text{Manager}, \text{Tel} : 1255766\}$.

4.1. Concepts of the Scheme Formalization

This scheme is mainly composed of the following basic algorithms. The specific structure is as follows:

(1) TA $\text{Setup}(\gamma, \text{S}) \rightarrow (\text{PK}, \text{MK})$: Takes input as the security parameter γ and all attribute sets S in the system, and the system outputs public key PK and the master key MK.

(2) AA $\text{Setup}(A, \lambda, \lambda_i, \lambda_{i,j}) \rightarrow (\text{MK}_{k+1}, \text{M}_j)$: Operates the initial algorithm with every AA by inputting the arbitrary parameter λ, λ_i, $\lambda_{i,j}$ and the attribute set A of the upper-layer authority; the master key MK_{k+1} and the public key M_j of the lower-layer authority are output.

(3) $\text{KeyGen}(\text{MK}, \text{M}_j, \text{MK}_{k+1}, \text{u}) \rightarrow (PK_u, \text{SK}_u, \text{SK})$: The user applies for a private key to the AA. Takes input as the master key MK, M_j, user identifiers u and MK_{k+1} in the system, and the system outputs the user's system public key PK_u, system private key SK_u and user's private key SK.

(4) $\text{Encrypt}^{\text{Offline}}(PK, \text{M}_j, \Lambda) \rightarrow IT$: The data owner invokes the offline encryption algorithm according to the system public key PK, the authority public key M_j, and the access structure Λ to obtain the middle ciphertext IT.

(5) $\text{Encrypt}^{\text{Online}}(IT, \text{m}, S) \rightarrow CT$: The data owner invokes the online encryption algorithm according to the middle ciphertext IT, plaintext m and attribute set S, and obtains the complete ciphertext CT uploads it to the CSP.

(6) $\text{Decrypt}(\text{MK}_{k+1}, SK, \text{CT}) \rightarrow m$: The user accesses the CSP and downloads the file to the local, and then calls the decryption algorithm to decrypt the ciphertext CT according to the private key MK_{k+1} of the attribute authority and the user private key SK. If the attribute related to the user satisfies the access policy embedded in the ciphertext CT, the decryption obtains the plaintext m; otherwise the decryption fails.

4.2. Security Model

Theorem 1: If any attacker's advantage $Adv_{Ad}(k)$ in the security game is negligible, then the scheme is called CPA (Chosen-Plaintext Attack) [3] security. The following is the construction process of choosing the plaintext security model in the security game plan. Let the attacker be Ad and the challenger be C. The specific game process is as follows:

(1) Initial: Attacker Ad arbitrarily selects an access policy Λ^*, challenged by challenger C.

(2) Setup: TA first executes the initialization algorithm and obtains the system public key PK. At the same time, each attribute authority also executes an initialization algorithm to obtain the public key M_j. The public key of the system and the public key of the attribute authority are sent to the attacker Ad by the challenger C.

(3) Phase 1: Attacker Ad sends a challenged attribute split set $A_1^*, ..., A_q^*$ requesting the construction of a private key, which includes the subset of attributes being queried. However, arbitrary A_i^* $(1 \leq i \leq q)$ and access tree structure Λ^* are inconsistent. Challenger C sends the obtained private key A_j, A_j' $(1 \leq j \leq q)$ to attacker Ad after performing the key generation algorithm.

(4) Challenge: Attacker Ad selects two equal length plaintext M_0 and M_1, and sends them to challenger C. Challenger C randomly selects $b \in \{0, 1\}$ and sends the ciphertext CT to Challenger C by encrypting the plaintext.

(5) Phase 2: Phase 1 is repeated.

(6) Guess: Attacker Ad gives a guess value $b^* \in \{0, 1\}$. If $b^* = b$, attacker Ad wins the final game. The advantage of the attacker in this game is:

$$Adv_{Ad}(k) = \left| \Pr[b^* = b] - \frac{1}{2} \right| \tag{2}$$

If the advantage $Adv_{\mathrm{Ad}}(k)$ of attacker Ad is negligible in polynomial time, it indicates that this scheme satisfies CPA security conditions.

5. Hierarchical Authority Based Weighted Attribute Encryption Scheme

(1) TA Setup: TA randomly selects g as a generator at the initialization stage. A bilinear group G_0, whose order is prime p, and satisfies the bilinear map e: $G_0 \times G_0 \to G_1$. All identifiers that represent the AA and user identity are uniformly distributed by the TA. AA is used to verify whether the user is legitimate and the user's identity is highly secure. $AID(AA) = \{P(AA), index(AA), P(AA) \in Z_p\}$ represents the random serial number of TA, and $index(AA) \in Z_p$ represents the random serial number that the TA distributes for AA.

The first $|\Gamma^*|$ elements in Z_p are randomly selected, i.e. $1, 2, ..., |\Gamma^*|$ $(\bmod p)$. Set the recursion depth of the user key structure to 2 for the first time. Then randomly select l_1, l_2, set $L_1 = g^{l_1}, L_2 = g^{l_2}$, $D_1 = g^{\frac{1}{l_1}}, D_2 = g^{\frac{1}{l_2}}$. Finally, choose $\alpha, y, \{\beta_1, \beta_2\} \in Z_p, h \in G_0$ randomly from Z_p, output the system public key as $PK = \{G_0, G_T, g, L_1, L_2, D_1, D_2, e(g, g)^\alpha\}$ and the system master key $MK = \{l_1, l_2, g^\alpha\}$.

(2) AA Setup: After the TA is initialized, a global identifier AID is generated for each AA to indicate the uniqueness of the identity. When the number of AA is N, the authority of each layer distributes keys for users according to the weighted attributes. It is assumed that the attribute set of the kth layer authority is $A_k = \{a_1, ..., a_{n_k}\}$, and the corresponding attribute split set is A_k'. Let $|A'_k| = n_k'$, take the first n_k' elements in Z_p, i.e. $1, ..., n_k' (\mathrm{mod} p)$. Then, randomly select $r_1, ..., r'_{n_k} \in Z_p, 1 \leq k \leq N$, the public key of the AA is:

$$M_j = (e(g, g)^c, g^{r_j})(1 \leq j \leq n_k') \tag{3}$$

1) *Upper-layer AA Authorization*: The system randomly selects parameter c and attribute set $A = \{A_0, A_1, ..., A_n\}$. The first layer attribute is A_0, and $A_i = \{(\lambda_i, 1), ..., (\lambda_i, \omega_i)\}(1 \leq i \leq n)$ is the attribute split set. When TA initializes AA, A is represented by randomly selected $\lambda \in Z_p$, $A_i \in A$ is represented by $\lambda_i \in Z_p$, and $a_{i,j}$ is represented by $\lambda_{i,j} \in Z_p$. The master key of the upper-layer AA is:

$$\begin{aligned} &MK_0 = \{A, g^{\alpha}, K = g^{\frac{\lambda-\alpha}{\beta_1}}, K_{i,j} = g^{\lambda_i} \cdot H(a_{i,j})^{\lambda_{i,j}}, \\ &K'_{i,j} = g^{\frac{1}{\lambda_{i,j}}}(0 \leq i \leq n, 0 \leq j \leq n), D_i = g^{\frac{\lambda-\lambda_i}{\beta_2}}(1 \leq i \leq n)\} \end{aligned} \tag{4}$$

D_i can be converted during the match of attributes to decrypt. When converting, $\frac{D_i}{D'_i}$ can go from λ'_i to λ_i.

2) *Lower-layer AA Authorization*: The upper-layer authority authorizes the authority to enter the system after verifying the identity of the lower-layer authority. Let A denote the attribute set of the upper-layer AA and A' denote the attribute set of the lower-layer AA to satisfy $A' \subset A$. The AA_{k+1}'s master key is distributed through the AA_k. As above, AA_k randomly selects $\lambda' \in Z_p$ for A' and $\lambda'_i \in Z_p$ for $A'_i \in A'$, $\lambda'_{i,j} \in Z_p, a'_{i,j} \in A'_i, 0 \leq i \leq n, 1 \leq j \leq n$. Then AA_{k+1}'s master key is:

$$\begin{aligned} &MK_{k+1} = \{A', g^{\alpha}, K' = K \cdot g^{\frac{\lambda'(\lambda-\alpha)}{\beta_1}}, K'_{i,j} = K_{i,j} \cdot g^{\lambda'_i} \cdot H(a'_{i,j})^{\lambda'_{i,j}}, \\ &K''_{i,j} = K'_{i,j} \cdot g^{\lambda'_{i,j}}(0 \leq i \leq n, 1 \leq j \leq n), D'_i = D_i \cdot g^{\frac{\lambda'+\lambda'_i}{\beta_2}}\} \end{aligned} \tag{5}$$

Among them, K, $K_{i,j}$, D_i are the corresponding items in AA_k.

(3) Key Generation: The user first applies for the private key to the lower layer AA. If the lower layer AA cannot generate the required weight attribute private key component for the user, then the request is handed over to the upper layer AA. If the upper layer AA is still unable to authorize, the user continues to go up. The application is handed over until a level of authority is able to satisfy the user's request.

1) *User System Private Key Generation*: The TA generates a system private key for a user who temporarily joins the system and sends a certificate. After the MK and the user's identity identifier u are entered into the system, TA randomly selects $z_u \in Z_p$. Calculate the public key of the system according to equation (6) and output:

$$PK_u = g^{Z_u} \tag{6}$$

Calculate the user's system private key:

$$SK_u = g^{\alpha} h^{y Z_u} \tag{7}$$

2) *User Private Key Generation*: TA assigns each AA a unique identifier AID and sends a verification code to AA to verify the validity of its certificate. AA first verifies the validity of the user's uid and then accepts the user's request for a private key. When the user has S attribute sets, the split set of weighted attributes is S'. The TA transmits the user system private key SK_u to each AA. Each AA will obtain the certificate $Certificate\,(u)$ submitted by the user and will also receive the verification code sent by the TA. The validity of the user certificate is verified by the verification code. After the verification is passed, it is determined whether the user's attribute exists in the authority, and after the existence is confirmed, the user private key is distributed to the user.

If the kth layer authority of the system directly operates on the user, then the K TAs will jointly manage the user attributes dispersed at each layer. The user attribute split set on the $k \leq K$ layer AA is denoted by $A^k = \{{A_1}^k, {A_2}^k, ..., {A_n}^k\}$. The kth layer authority randomly generates an identifier AID, which is denoted by c. $\tilde{\lambda}^k \in Z_p$, $\tilde{\lambda}_i^k \in Z_p$, $\tilde{\lambda}_{i,j}^k \in Z_p$, $r' \in AID$ is arbitrarily selected by AA, output the user's private key:

$$SK = (g^c, A^k, D = g^{\frac{\tilde{\lambda}^k - r'}{\beta_{1,k}}}, P = g^{\alpha} g^{u/\lambda} g^{\lambda' r'}, \mathrm{P}' = g^{r'}, D_j = g^{\tilde{\lambda}_i^k/\lambda'} \cdot H(a_{i,j}{}^k)^{\tilde{\lambda}_{i,j}^k},$$
$$D'_j = g^{\lambda \cdot \tilde{\lambda}_{i,j}^k} (0 \leq i \leq n, 0 \leq j \leq n)) \tag{8}$$

(4) Encrypt: The encryption algorithm of this scheme divides the encryption process into two parts: online/offline. In this algorithm, user attributes are classified and each attribute has a public key. The intermediate ciphertext is obtained by encryption during the offline phase; after the user attribute set and plaintext are known in the online phase, it quickly generates these intermediate ciphertexts into complete ciphertexts by a simple calculation. The data owner encrypts his data through the encryption algorithm and then uploads the generated ciphertext to the CSP storage. The main encryption algorithms are as follows:

1) *Offline Encryption*: The system public key PK, the attribute authority public key M_j, and the access tree Λ are input into the system, and the node x threshold is represented by k_x. Each node x in the tree corresponds to a polynomial q_x. Starting from the root node R, the relation $d_x = k_x - 1$ is satisfied. If x is a leaf node, $d_x = 0$ is satisfied. Select random number $s \in Z_p$, let $q_R(0) = s$, the other node x set value: $q_x(0) = q_{parent}(index(x))$. Optionally a polynomial q_x whose order is d_x. Y denotes the set of all leaf nodes, Y' denotes the split set of weighted attributes, $attr\,(x)$ denotes $x \in Y$ attributes, and the middle ciphertext is calculated: $IT = (C_0, C_1, C_2, M, \Lambda)$.

$$\begin{cases} M = \prod\limits_{c \in IA} e(g,g)^{\alpha s}, C_0 = g^s \\ C_1 = g^{\lambda' q_x(0)}, C_2 = H(attr(x))^{q_x(0)} \end{cases} \tag{9}$$

2) *Online Encryption*: Input middle ciphertext IT, plaintext message m and user's attribute set $A_k = \{a_1, ..., a_{n_k}\}$ into the system. Calculated according to equation (10), where I_A is the smallest attribute set of the user and the complete ciphertext is output: $CT = (IT, C)$.

$$C = m \prod_{c \in I_A, x_{n_i} \in A} \left(e(g,g)^{\lambda_n}\right)^{x_{n_i}} = m \prod_{c \in I_A, x_{n_i} \in A} e(g,g)^{\lambda_n {x_{n_i}}} = mM \tag{10}$$

(5) Decrypt: The user downloads the file to be accessed from the CSP and then decrypts it with the obtained user private key. The premise of ciphertext decryption is that the identifier AID of the TA in the ciphertext and the private key component of the user are the same, and when the attribute possessed by the user satisfies the structure of the access tree Λ, i.e. $\sum\limits_{p \in \{S' \cap Y'\}} weight(p) \geq t$. Selecting t elements from the set $K = \{S' \cap Y'\}$ decrypts the ciphertext to get the plaintext. otherwise it returns $\perp$. Call $\mathrm{Decrypt}(\mathrm{MK}_{k+1}, SK, \mathrm{CT}) \rightarrow m$, for any node x in the access tree, after executing the recursive algorithm, get a set S_x. When the user's attribute set matches the access structure of the ciphertext in the authority of this layer, an arbitrary value $i \in S$ is selected,

and the recursive function is executed from the root node. Otherwise, the null value is returned. For any node x: If $i \in A_{c\in I_A}$, set $\mathrm{i} = \mathrm{attr(x)}$, calculate:

$$\begin{aligned}\frac{e(C_1(i),D_i)}{e(C_2(i),D'_i)} &= \frac{e(g^{\lambda' q_x(0)},g^{\tilde{\lambda}_i^k/\lambda'}\cdot H(a_{i,j}^k)^{\tilde{\lambda}_{i,j}^k})}{e(H(i)^{q_x(0)},g^{\lambda'\cdot\tilde{\lambda}_{i,j}^k})} = \frac{e(g^{\lambda' q_x(0)},g^{t\tilde{\lambda}_{i,j}^k})\cdot e(g^{\lambda' q_x(0)},g^{\tilde{\lambda}_i^k/\lambda'})}{e(g^{tq_x(0)},g^{\lambda'\cdot\tilde{\lambda}_{i,j}^k})} \\ &= e(g,g)^{\tilde{\lambda}_i^k q_x(0)}\end{aligned} \tag{11}$$

If $i \notin A_{c\in I_A}$, this algorithm has no solution and returns $\perp$. If and only if the user's attribute set $\sum_{c\in I_A} A$ matches the access tree Λ, the equation (12) is calculated by the Lagrange mean value theorem, then the equation (13) is calculated, and the equation (14) is finally calculated, as follows:

$$\begin{cases} e(g,g)^{\tilde{\lambda}_i^k q_x(0)} = e(g,g)^{\tilde{\lambda}_i^k s} \\ e(g,g)^{I_A\tilde{\lambda}_i^k q_x(0)} = e(g,g)^{I_A\tilde{\lambda}_i^k s} \end{cases} \tag{12}$$

$$\begin{aligned}\prod_{c\in I_A}\frac{e(C_0,P)}{e(C_1,\mathrm{P}')} &= \prod_{c\in I_A}\frac{e(g^s,g^\alpha g^{u/\lambda}g^{\lambda' r'})}{e(g^{\lambda' s},g^{r'})} = \prod_{c\in I_A}\frac{e(g^s,g^{\lambda' r'})e(g^s,g^\alpha g^{u/\lambda})}{e(g^{\lambda' s},g^{r'})} \\ &= e(g,g)^{I_A us/\lambda}\prod_{c\in I_A} e(g,g)^{\alpha s}\end{aligned} \tag{13}$$

$$\frac{e(g,g)^{I_A us/\lambda}\prod_{c\in I_A} e(g,g)^{\alpha s}}{e(g,g)^{I_A us/\lambda}} = \prod_{c\in I_A} e(g,g)^{\alpha s} = M \tag{14}$$

Output plaintext: $m = \frac{C}{M}$.

Finally, the user who meets the ciphertext access policy can decrypt the ciphertext CT and get the plaintext m to be accessed; if it is not satisfied, it can not be decrypted, and the file to be accessed cannot be obtained.

6. Security Analysis

6.1. Security Proof

The proof of security of this scheme is based on the difficult problem of DBDH (Decisional Bilinear Diffie-Hellman) [10]. If the attacker's advantage $Adv_{Ad}(k)$ is negligible, it indicates that the solution is to satisfy CPA security. The theorem of CPA security is given below, which is proved by proof by contradiction.

Theorem 2: If the advantage of solving DBDH problem on (G_0, G_1) is negligible, then the DBDH assumption is established on (G_0, G_1), i.e. the scheme is CPA safe under the standard model.

Proof: Use proof by contradiction to prove. Assuming an attacker's advantage $Adv_{Ad}(k)$ is not negligible in polynomial time, the attacker wins. Use the following games to further illustrate:

(1) Initial: Attacker Ad randomly selects an access policy Λ^*.

(2) Setup: Simulator K runs an "Initial" algorithm to obtain the system's public key $PK = \{G_0, G_T, g, L_1, L_2, D_1, D_2, e(g,g)^\alpha\}$ and the system's master key $MK = \{l_1, l_2,$

$g^{\alpha}\}$. Challenger C saves the master key MK itself, and the system public key is sent to attacker Ad.

(3) Phase 1: Attacker Ad sends a private key access request for the partitioned attribute set ${A_1}^*, {A_2}^*, ..., {A_q}^*$. Note that ${A_i}^*(1 < i < q)$ is completely mismatched with the structure Λ^* of the access tree. Perform private key generation algorithm:

$$\begin{aligned} &SK = (g^c, A^k, D = g^{\frac{\tilde{\lambda}^k - r'}{\beta_{1,k}}}, P = g^{\alpha} g^{u/\lambda} g^{\lambda' r'}, \mathrm{P}' = g^{r'}, D_j = g^{\tilde{\lambda}_i^k/\lambda'} \cdot H(a_{i.j}^k)^{\tilde{\lambda}_{i,j}^k}, \\ &{D'}_j = g^{\lambda \cdot \tilde{\lambda}_{i,j}^k} (0 \le i \le n, 0 \le j \le n)) \end{aligned} \tag{15}$$

Send the private key to attacker Ad.

(4) Challenge: Attacker Ad sends two equal length plaintext M_0, M_1 and challenged access structure Λ^* to challenger C. Attacker Ad did not find Challenger C's key during the query phase and the simulator randomly selected a bit attribute $\eta \in \{0, 1\}$. First, calculate C_0, C_1,C_2 and M. Under access structure Λ^*, encrypt the plaintext M_η offline to obtain the middle ciphertext: $IT = (C_0, C_1, C_2, M, \Lambda^*)$. Calculate:

$$\hat{C} = M_\eta \prod_{c \in I_A, x_{n_i} \in A} \left(e(g,g)^{\lambda_n}\right)^{x_{n_i}} = M_\eta \prod_{c \in I_A, x_{n_i} \in A} e(g,g)^{\lambda_n x_{n_i}} = M_\eta M \tag{16}$$

When $b = 0$, define equation (17)

$$Z = e(g,g)^{abc} \tag{17}$$

Let $bc = \theta$, $\hat{C}$ be ciphertext, so:

$$\hat{C} = M\eta \cdot e(g,g)^{a\theta} \tag{18}$$

Otherwise, when $b = 1$, there is:

$$\begin{cases} Z = e(g,g)^z \\ \hat{C} = M\eta \cdot Z = M\eta \cdot e(g,g)^z \end{cases} \tag{19}$$

Since z is randomly selected, $\hat{C}$ obtained at this time is a random element, and $\hat{C}$ does not include information related to M_η.

(5) Phase 2: Attacker Ad repeatedly sends a private key request for user attribute set $A_1^*, A_2^*, ..., A_q^*$, i.e. $A_1^*, A_2^*, ..., A_q^*$ does not meet the access structure Λ^* in the ciphertext.

(6) Guess: Attacker Ad outputs a conjecture on η. If $\eta' = \eta$, the attacker guesses correctly and the simulator outputs $b' = 0$, indicating that the tuple received by the simulator is the DBDH tuple $(g, g^a, g^b, g^c, e(g,g)^{abc})$. Otherwise, the simulator outputs $b' = 1$, which is a five-tuple $(g, g^a, g^b, g^c, e(g,g)^z)$. In the above game, if $b = 1$, the attacker is not receiving any information related to the ciphertext M_η, i.e.

$$\Pr[\eta' \ne \eta \,|b = 1\,] = \frac{1}{2} \tag{20}$$

Because, if $\eta' \ne \eta$, the simulator guesses $b' = 1$, then

$$\Pr[b' = b \,|b = 1\,] = \frac{1}{2} \tag{21}$$

When $b = 0$, the attacker gets a ciphertext M_η of a scheme. By definition, the attacker has a non-negligible advantage ε to break this scheme, so

$$\Pr[\eta' = \eta \,|b = 0] = \varepsilon + \frac{1}{2} \tag{22}$$

Because, if $\eta' = \eta$, the simulator will guess $b' = 0$, so there is

$$\Pr[b' = b \,|b = 0] = \varepsilon + \frac{1}{2} \tag{23}$$

So in the above game, the simulator correctly guesses the advantage of $b' = b$ is:

$$\begin{array}{l} Adv_C = \Pr[b' = b] - \frac{1}{2} \\ = \frac{1}{2}\Pr[b' = b \,|b = 1] + \frac{1}{2}\Pr[b' = b \,|b = 0] - \frac{1}{2} \\ = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot (\frac{1}{2} + \varepsilon) - \frac{1}{2} \\ = \frac{\varepsilon}{2} \end{array} \tag{24}$$

Based on the definition and the above security game, in any polynomial-time algorithm, the advantage of attacker Ad winning the game is negligible, so the scheme is CPA safe.

6.2. Security Analysis

(1) Collusion Resistance: When each user is registered, the TA will distribute a unique uid for it. Generate the user's system private key by randomizing parameters:

$$SK_u = g^\alpha h^{yz_u} \tag{25}$$

The TA distributes random AID to each layer of AA. The TA first splits the user's attribute set according to the size of the weight value, and the attribute split set is sent by the TA to the AA of each layer. Before AA distributes the attribute private key for users, it first verifies the validity of each user's AID. If the attribute of the user belongs to the layer AA, the user private key $SK = (g^c, A^k, D = g^{\frac{\tilde{\lambda}^k - r'}{\beta_{1,k}}}, P = g^\alpha g^{u/\lambda} g^{\lambda' r'}, \mathrm{P}' = g^{r'}, D_j = g^{\tilde{\lambda}_i^k/\lambda'} \cdot H(a_{i.j}^k)^{\tilde{\lambda}_{i,j}^k}, D_j' = g^{\lambda \cdot \tilde{\lambda}_{i,j}^k} (0 \le i \le n, 0 \le j \le n))$ is distributed for it. Each user's private key is also formed by the serial number and randomization parameters, so that at the time of decryption, each user's private key is embedded with a different random value [11]. During encryption, the encryption exponent s, the random numbers λ, λ_n and the weighted split set parameter x are all embedded in different ciphertext segments. If multiple users collusion, even if $H(attr(x))^{q_x(0)}$ is recovered, the random numbers of different users cannot recover g^s and $g^{\lambda' q_x(0)}$. Therefore, the middle ciphertext IT cannot be recovered, and the final ciphertext cannot be recovered. In the end, the collusion user could not decrypt the ciphertext and thus well achieved the collusion resistance attack.

(2) Hierarchical Authority Security: When the user decrypts, the TA needs to generate the user's system private key, and the AA of this layer generates the AA private key to jointly constitute the user's private key [25]. If the TA is compromised, the user's system private key may be exposed, but it cannot be decrypted with the system private key. If any AA is compromised, the attribute key may be exposed and the ciphertext cannot be

decrypted accurately. In summary, multi-authority can collectively manage keys, reducing the heavy workload of a single authority, and improve the security of the system and reduce the overall risk.

(3) Access Policy Flexibility: It allows the intersection between user attributes of this scheme and the expression of access policy is also more flexible. The attribute sets of multiple users can also be queried. The entire process is based on the encryption of data by the data owner.

(4) System Scalability: This scheme extends from TA to multi-authority to jointly manage user's keys. This has reduced the workload of the previous single authority and improved the work efficiency. At the same time, the security of the entire system has also been improved.

7. Performance Analysis

7.1. Performance Comparison

The above security proofs show that this scheme is CPA safe. This scheme and existing schemes [6], [23] and [14] are compared and analyzed in terms of function and performance. It shows the results in Table 1.

In Table 1, suppose n is the number of attributes in the access policy. The pairing operation, exponentiation operation of G_1, exponentiation operation of G_2 are denoted by T_{p}, T_{e1}, T_{e2} respectively.

Table 1. Performance Comparison of Related Schemes

Schemes	Data encryption	Data decryption	Multi-authority	Weighted attribute	Hierarchical
[6]	$2nT_{e1}+T_{e2}+T_p$	$3T_{e2}+T_p$	√	×	×
[14]	$(2n+1)T_{e1}+T_{e2}+T_p$	T_{p}	√	×	√
[23]	$nT_{e1}+T_{e2}+T_p$	$2T_{e2}$	√	√	×
Our scheme	nT_{e2}	T_{e2}	√	√	√

From Table 1, we can see that in [6], the use of multi-authority to reduce the pressure on the central authority to manage and distribute private keys is considered. However, in this scheme, the problem of authority hierarchy and weighted attribute is not considered, and more fine-grained access control policies cannot be implemented. Although Ref. [14] introduces a hierarchical approach to solve the hierarchical relationship between multi-authority. However, in this scheme, the attribute sets managed by different levels of authority are equally important, and the status of each attribute in the system is equal. Ref. [23] distinguishes the importance of different attributes by introducing weighted attributes, but the scheme does not consider the hierarchical constraints among the various authorities. This makes it possible for some unreliable authorities to distribute the private key corresponding to the attribute with a relatively large weight, which brings certain security risks to the encryption system. The scheme proposed in this paper, it adopts the hierarchy to solve the hierarchical relationship among multi-authority, and weighted attributes are introduced to distinguish the importance of different attributes, making the actual system more secure and flexible.

From Table 1, it can be seen that the encryption cost in [14] is large. Since the encryption phase of the scheme proposed in this paper is divided into two online and offline processes, most of the tedious calculation processes are placed in the offline stage, which greatly reduces the computational cost of the encryption stage. From the table, it can be seen that the encryption cost of the scheme proposed in this paper is much smaller than the other three schemes. In terms of decryption cost, the cost of [6] has experienced three exponential operations and one pairing operation. Ref. [14] runs a partial decryption algorithm, which requires a pairing operation to recover the plaintext. However, in [23], users need to perform two exponential operations when decrypting. When the scheme proposed in this paper is decrypted, only one exponential operation needs to be performed. Overall, the scheme proposed in this paper is relatively excellent, and it has improved in terms of function and performance, and is more suitable for cloud environments.

7.2. Simulation Evaluation

In order to show the performance of this scheme more intuitively, this paper selects scheme [1] and scheme [8] to carry out simulation experiments, and evaluates the effectiveness of the scheme through experiments. The implementation is on a Linux system with CPU i5-6500 at 3.20GHZ and the memory is 8GB. The implementation adopts a 224-bit elliptic curve group from Pairing-Based Cryptography Library [18] and based on the CP-ABE package. It shows the experimental results in Fig. 3 and Fig. 4.

The complexity of the access policy affects computational and communication overhead. We generate the form of the access strategy $(A_1, A_2, ..., A_n)$ to simulate the worst environment, and $A_{i,i\in(1,n)}$ is an attribute. We set distinct access policies in this form and the number of user's attributes vary from 10 to 50. It must compare all the attributes of user that can know whether his/her attributes satisfy the access policy or not. The time is given in milliseconds.

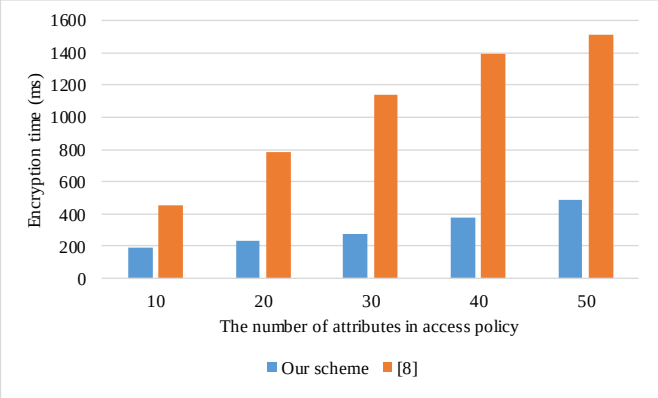


Fig. 3. Comparison of encryption time

Fig. 3 shows the relationship between the encryption time and the number of attributes. The encryption time of the two schemes increases as the number of attributes in-

creases. As can be seen from the figure, the encryption time cost of the proposed scheme is obviously less than that of scheme [8]. This is because the encryption phase of the scheme proposed in this paper is divided into two phases: online and offline. Most of the encryption calculations are completed in the offline phase.

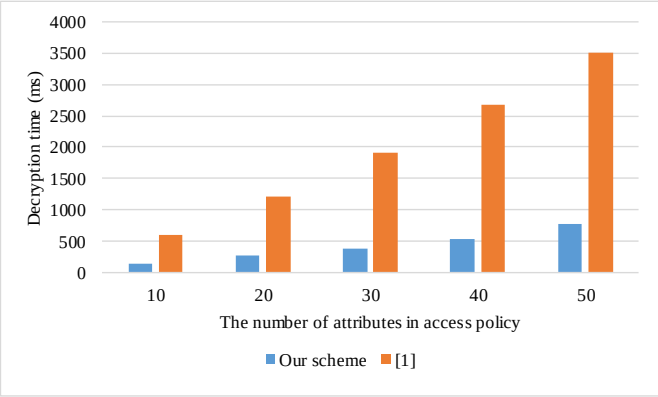


Fig. 4. Comparison of decryption time

As shown in Fig. 4, the comparison of the decryption time consumption for the scheme proposed in this paper and scheme [1]. As can be seen from the figure, the decryption time of the scheme proposed in this paper is less than that of scheme [1], and with the increase of the number of attributes, the advantages of the scheme proposed in this paper become more obvious. This is because in the process of decryption, the amount of calculation of scheme [1] is large, while the scheme proposed in this paper only needs to perform one exponent operation and the efficiency is high.

8. Conclusions

For the problem of cloud storage security, this paper proposes a hierarchical authority based weighted attribute encryption scheme. Through the introduction of the mechanisms of hierarchical authorities and the weight of attribute. There is a hierarchical relationship among the authorities, different attributes have different importance, a more fine-grained access control is achieved, and the security of the system is enhanced. By performing online and offline encryption mechanisms, it places most of the tedious calculation processes on the offline stage, greatly reducing the amount of computation in the encryption stage and reducing the computational cost. Compared with other schemes, the decryption cost is also reduced, which improves the operating efficiency of the system. Through security analysis and comparison, it is shown that the scheme proposed in this paper is safe and efficient under the cloud storage environment.

Acknowledgment. This work has been supported by the National Natural Science Foundation of China (No. 61672338 and No. 61873160).

References

1. Ai, Q., Dan, T., Wang, Z., Jianwei, L.: Hierarchical authority attribute-based encryption. Journal of Wuhan University (Natural Science Edition) 60(5), 441–446 (2014)
2. Beimel, A., Tassa, T., Weinreb, E.: Characterizing ideal weighted threshold secret sharing. In: Theory of Cryptography Conference. pp. 600–619. Springer (2005)
3. Bergen, H.A., Hogan, J.M.: A chosen plaintext attack on an adaptive arithmetic coding compression algorithm. Computers & Security 12(2), 157–167 (1993)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Annual international cryptology conference. pp. 213–229. Springer (2001)
5. Chase, M.: Multi-authority attribute based encryption. In: Theory of Cryptography Conference. pp. 515–534. Springer (2007)
6. Chase, M., Chow, S.S.: Improving privacy and security in multi-authority attribute-based encryption. In: Proceedings of the 16th ACM conference on Computer and communications security. pp. 121–130. ACM (2009)
7. Chattopadhyay, A.K., Nag, A., Majumder, K.: Secure data outsourcing on cloud using secret sharing scheme. IJ Network Security 19(6), 912–921 (2017)
8. Chen, D., Liu, S., Li, Q.: Multi-authority and hierarchical weighted attribute-based encryption scheme. Journal of Nanjing University of Posts and Telecommunications 36(5), 18–23 (2016)
9. Chen, W., Zhang, Y., Zheng, D.: Weighted multi-authority attribute encryption based on ciphertext policy. Journal of GuiLin University of Electronic Technology 35(5), 386–390 (2015)
10. Cheung, L., Newport, C.: Provably secure ciphertext policy abe. In: Proceedings of the 14th ACM conference on Computer and communications security. pp. 456–465. ACM (2007)
11. Cui, M., Han, D., Wang, J.: An efficient and safe road condition monitoring authentication scheme based on fog computing. IEEE Internet of Things Journal (2019)
12. Gorasia, N., Srikanth, R., Doshi, N., Rupareliya, J.: Improving security in multi authority attribute based encryption with fast decryption. Procedia Computer Science 79, 632–639 (2016)
13. Han, D., Bi, K., Xie, B., Huang, L., Wang, R.: An anomaly detection on the application-layer-based qos in the cloud storage system. Comput. Sci. Inf. Syst. 13(2), 659–676 (2016)
14. Huang, Q., Yang, Y., Shen, M.: Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing. Future Generation Computer Systems 72, 239–249 (2017)
15. Liu, X., Ma, J., Xiong, J., Li, Q., Ma, J.: Ciphertext-policy weighted attribute based encryption for fine-grained access control. In: 2013 5th International Conference On Intelligent Networking And Collaborative Systems. pp. 51–57. IEEE (2013)
16. Liu, X., Ma, J., Xiong, J., Li, Q., Zhang, T.: Ciphertext-policy weighted attribute based encryption scheme. Journal of Xi'an Jiaotong University 47(8), 44–48 (2013)
17. Luo, E., Liu, Q., Wang, G.: Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks. IEEE Communications Letters 20(9), 1772–1775 (2016)
18. Lynn, B., et al.: Pbc library. Online: http://crypto. stanford. edu/pbc 59, 76–99 (2006)
19. Nag, A., Choudhary, S., Dawn, S., Basu, S.: Secure data outsourcing in the cloud using multi-secret sharing scheme (msss). In: Proceedings of the First International Conference on Intelligent Computing and Communication. pp. 337–343. Springer (2017)
20. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 457–473. Springer (2005)
21. Wan, Z., Liu, J., Deng, R.H.: Hasbe: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing. IEEE transactions on information forensics and security 7(2), 743–754 (2012)
22. Wang, G., Liu, Q., Wu, J.: Hierarchical attribute-based encryption for fine-grained access control in cloud storage services. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 735–737. ACM (2010)

23. Wang, Y., Zhang, D., Zhong, H.: Multi-authority based weighted attribute encryption scheme in cloud computing. In: 2014 10th International Conference on Natural Computation (ICNC). pp. 1033–1038. IEEE (2014)
24. Yang, K., Jia, X., Ren, K., Zhang, B., Xie, R.: Dac-macs: Effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security 8(11), 1790–1801 (2013)
25. Yang, L., Deng, Y., Yang, L.T., Lin, R.: Reducing the cooling power of data centers by intelligently assigning tasks. IEEE Internet of Things Journal 5(3), 1667–1678 (2018)
26. Zhou, Y., Deng, Y., Xie, J., Yang, L.T.: Epas: A sampling based similarity identification algorithm for the cloud. IEEE Transactions on Cloud Computing 6(3), 720–733 (2018)
27. Zou, X.: A hierarchical attribute-based encryption scheme. Wuhan University Journal of Natural Sciences 18(3), 259–264 (2013)

Qiuting Tian is currently a Ph.D. at Shanghai Maritime University. Her research interests include cloud computing, mobile networking security and cloud security.

Dezhi Han received the Ph.D. degree from the Huazhong University of Science and Technology. He is currently a Professor of computer science and engineering with Shanghai Maritime University. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Yanmei Jiang is currently a master student of Shanghai Maritime University. Her research interests include cloud computing and mobile networking security.

Received: September 12, 2018; Accepted: August 8, 2019.

Computer Science and Information Systems 16(3):815–830 https://doi.org/10.2298/CSIS180915028L

Algorithm of Web Page Similarity Comparison Based on Visual Block

Xingchen Li[1], Weizhe Zhang[1,2], Desheng Wang[1], Bin Zhang[2], and Hui He[1]

[1] School of Computer Science and Technology,
Harbin Institute of Technology, Harbin,China
16s003084@stu.hit.edu.cn, (wzzhang,wangdesheng0821,hehui)@hit.edu.cn
[2] Cyberspace Security Research Center,
Peng Cheng Laboratory, Shenzhen, China
bin.zhang@pcl.ac.cn

Abstract. Phishing often deceives users due to the relative similarity to the true pages on a layout and leads to considerable losses for the society. Consequently, detecting phishing sites has been an urgent activity. By researching phishing web pages using web page screenshots, we discover that this kind of web pages use numerous web page screenshots to achieve the close similarity to the true page and avoid the text and structure similarity detection. This study introduces a new similarity matching algorithm based on visual blocks. First, the RenderLayer tree of the web page is obtained to extract the visual block. Second, an algorithm that will settle the jumbled visual blocks, including the deletion of the small visual blocks and the emergence of the overlapping visual blocks, is designed. Finally, the similarity between the two web pages is assessed. The proposed algorithm sets different thresholds to achieve the optimal missing and false alarm rates.

Keywords: phishing, similarity comparison, visual block, web rendering.

1. Introduction

With the advent of the network information age, the Internet has significantly improved people's works and lives [16] and has accelerated the spread of information. However, prohibited tools for illegal profit-making have emerged simultaneously. Phishing websites is one of the killers that threaten network development in China. These websites confuse users when a domain name similar to the real website is used [18], thereby seriously damaging the vital interests of regular enterprises and numerous netizens and seriously disrupting China's network trading environment. At present, phishing websites mainly target financial and e-commerce websites [7] to defraud money. With the increase in the popularity of family digitalization and network broadband and the emergence of new consumption patterns (e.g., e-commerce, online bank payment, and online settlement), phishing attacks have become the most serious threat to the realistic safety of e-commerce [17].

Web page similarity comparison relates to network rendering technology in addition to active detection technology. Network rendering technology involves the rendering principles of a browser and a kernel and the generation process of the rendering tree.

The rendering principle of a browser is presented in Fig. 1. A browser parses a web page into three steps.

1) A document object model (DOM) tree is produced using HTML, XHTML, or SVG files..
2) The CSS files will generate a CSS rule tree.
3) JavaScript scripts, mainly through the DOM and CSSOM API, are used to operate the DOM and CSS rule trees.

After completing the parsing, the browser engine constructs the rendering tree through the DOM and CSS rule trees. The main steps are as follows:

1) The rendering tree is not equivalent to the DOM tree because of a header or a display; none is unnecessary in the rendering tree.
2) The CSS rule tree must complete the match and attach the CSS rule to each element (or frame) on the rendering Tree, which is the DOM node.
3) The position of each frame (i.e., each element) is calculated. This procedure is called the layout and reflow process.

Finally, the API of the operating system is drawn which is called native GUI.

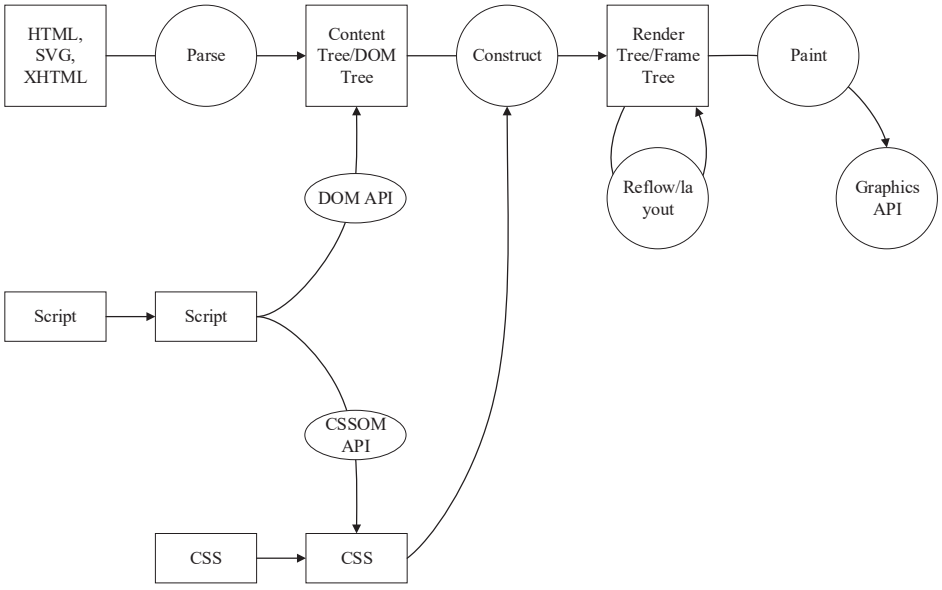


Fig. 1. Browser rendering principle

The construction process of the rendering tree is illustrated in Fig. 2. The browser's rendering page must first build the DOM and CSSOM trees. The DOM tree captures the attributes and relationships of the document markup, whereas the CSSOM tree presents the appearance of all elements after rendering. The CSSOM and the DOM trees are merged into the rendering tree and then used to calculate the layout of each visible element. The results are outputted into the rendering process to render the pixels to the

screen. The rendering tree, which contains only the nodes required to render the web pages, traverses the nodes in each DOM tree, thereby finding the style of the current node in the CSSOM rule tree to generate the rendering tree. In addition, the rendering tree calculates the exact location and size of each object in the layout of the web page. Every visible node is traversed from the root node of the DOM tree during the traversal process. The invisible nodes that are not reflected in the rendering output (e.g., script and meta tags) will be ignored. Similarly, some nodes that are hidden by CSS will also be ignored in the rendering tree. The suitable CSSOM rules for each node are then determined and applied. The matching begins from the right side of the selector to the left, thereby indicating that the matching begins from the child node of the CSSOM tree to the parent node. In addition, the time required to execute the rendering tree construction, layout, and drawing will depend on the size and application style of the document and on the device that runs the document. That is, a large document indicates additional work that the browser must complete. In addition, the drawing is time-consuming when the style is increasingly complex.

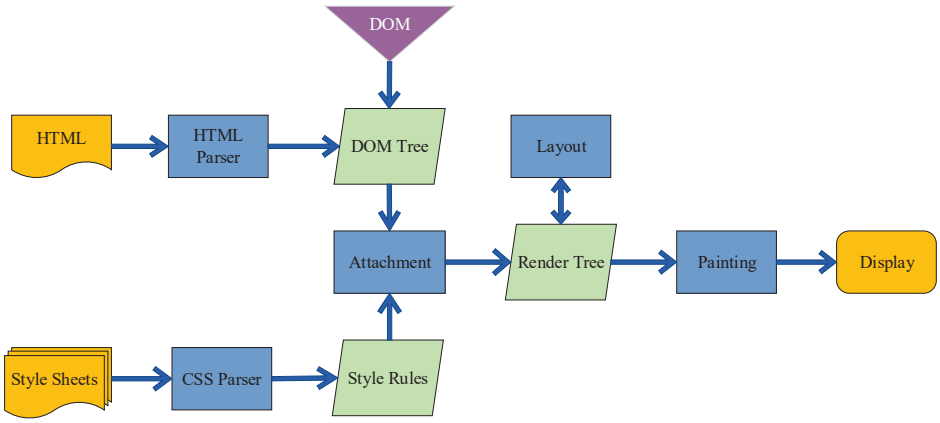


Fig. 2. The construction process of render tree

On the basis of some visual nodes of the DOM tree, the browser will build the corresponding nodes in accordance with the node properties. These nodes will also form a rendering tree. The browser will create new nodes on the basis of this rendering tree to form a RenderLayer tree. The structural relationship between the three types of trees is demonstrated in Fig. 3.

On the one hand, the rendering tree is a new tree that is built in accordance with the DOM tree. The nodes of the rendering tree do not directly correspond to those of the DOM tree. When encountering the non-visual nodes in the DOM tree, the rendering tree will not create new nodes. After building the rendering tree, the layout operation will calculate the related attributes, including location, size, and floating. This information will allow the rendering engine to know the location and the manner of drawing the elements.

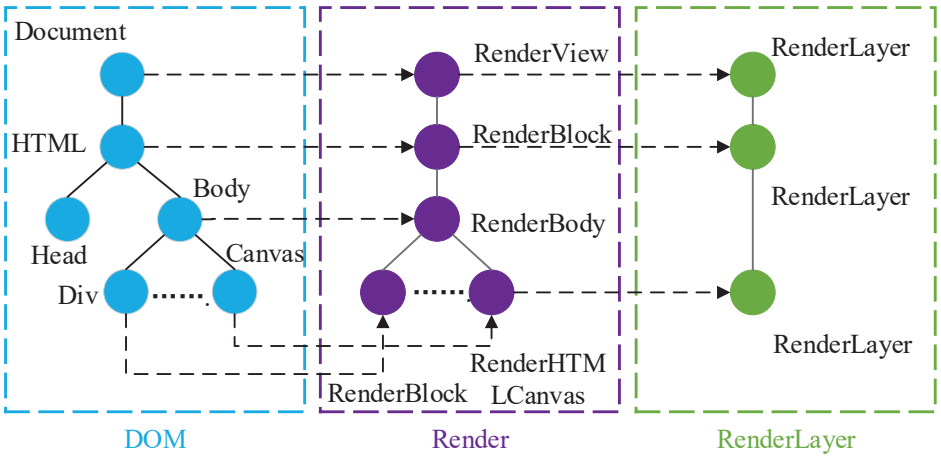


Fig. 3. The relationship between the three trees

On the other hand, the RenderLayer tree is a new tree that is built on the basis of the rendering tree. In addition, the nodes of this tree do not directly correspond to those of the rendering tree. Alternatively, a one-to-many relationship occurs among the nodes. In some cases, the tree must create new RenderLayer points. In the present study, the visual information of the web pages is obtained by analyzing the RenderLayer trees of the web pages and then extracting the information.

In this study, we devise a method to detect phishing sites based on visual blocks. The major contributions of this study are presented as follows:

1) Obtain the RenderLayer tree of the web page where the visual block will be extracted.
2) Design an algorithm that will settle jumbled visual blocks. This process involves deleting the small visual blocks and merging the overlapping visual blocks.
3) Define the similarity of the two web pages, and set different thresholds for the algorithm to achieve the optimal miss and error rates.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 discusses the similarity matching algorithm. Section 4 introduces the evaluation results. Section 5 provides the conclusions drawn from this study.

2. Related Work

Phishing detection methods are mainly divided into two categories, namely, detecting URL and identifying the content of the web pages. The former identifies a phishing website in accordance with the URL through mathematical modeling and is suitable for detecting large quantities of web pages.

The research on web page similarity can be divided into the following categories: traditional text, structural, and visual similarity studies.

The text-based similarity detection extracts the text contents of the web page and compares them with the text template of the phishing web page to determine if the latter is a phishing website. The commonly used algorithms are Simhash and Imatch algorithms. Takama et al. [3] proposed a fast HTML web page detection approach, which provided direct access to node information by hashing the web page. The detection of the changes in the two versions of a page was accomplished by performing similarity computations after transforming the web page into an XML-like structure. It showed rapid improvements when compared to the results of a previous approach. Zhuang et al. [21] extracted 10 different types of features such as title, keyword and link text information to represent the website. Roopak et al. [14] proposed a novel method for detecting phishing pages by searching similar web pages through web mining and then comparing the web pages by matching the HTML source codes and computing the cosine similarity of the textual contents. The experiment results indicated that the detection rate is higher in the proposed mechanism than in other existing methods. Nichele et al. [12] presented a domain taxonomy-based clustering approach, which integrated page similarities to compute the session similarity. In the paper [15], authors used the Jaccard coefficient formula to determine the similarity among the web pages. This approach can be applied for usage and navigation clustering purposes. Huo et al. [13] proposed an N-gram algorithm to realize the comparison of web page similarities. Anari et al. [2] determined the similarity of web pages based on learning automata and probabilistic grammar.

DOMs are first used to express the internal structure of HTML files, thus contributing significantly to HTML parsing. Typical algorithms for structural similarity research are web page similarity measurement methods based on edit distance and statistical features. Edit distance, which is also known as Levenshtein distance, refers to the minimum number of edit operations required between two strings, from one to another. Licensed editing operations involve replacing one character with another, inserting one character, and deleting one character. In general, a small edit distance indicates a considerable similarity between two strings. Similar to the string, the number of edit operations is defined by inserting, deleting, and modifying the nodes of the tree. Similarly, a small edit distance of the two trees implies a huge similarity between the two trees. Therefore, the similarity of the web page can be calculated by comparing the edit distance of the web page's DOM tree. Web page similarity measurement methods based on statistical features include a method based on link and node features. The link feature focuses on the order of the nodes, while the node feature focuses on the content of the nodes. In the paper [9], authors constructed the links from the root node to the leaf node in the DOM tree of the independent web page and regarded the link frequency as a feature of the link to judge the similar web pages based on link frequency similarity. This algorithm is difficult but simple because it only considers tag information. Alpuente et al. [1] proposed a functional technique that transforms each web page into a compressed, normalized tree to represent a visual structure. An optimization of this technique, which was developed on the basis of memorization, achieved remarkable improvements in efficiency in time and space.

Visual similarity detection uses the similarity of the image to match the similarity of the web pages [8,19,11]. By cutting the web page image and extracting the size, color, pixel, and other features of each sub-graph, the distance between the two features and the similarity with the feature template are calculated. The visual similarity detection system of the DOM structures of web pages proposed by Liu et al. [20] focused on the similarity

of web page typesetting. Liu et al. [10] detected phishing web pages in accordance with the similarity among the visual blocks after rendering the screening page, which involves two processes. The first process ran on local email servers and monitored emails for keywords and suspicious URLs, while the second process compared the potential phishing pages against actual pages and assessed the visual similarities among them in terms of key regions, page layouts, and overall styles. Fu et al. [6] proposed a phishing detection scheme that calculated the visual similarity of web pages based on Earth mover distance. In the paper [5], authors developed an automated process for crawling web pages. The Porter stemmer algorithm was used to convert the keywords into basic forms, and the term frequency–inverse document frequency method was utilized to obtain the importance of a keyword. In the paper [4], authors detected phishing using website components, such as source code, CSS, and logo. Furthermore, this method adopted the Jaccard coefficient formula to determine the similarity among web pages.

3. Similarity Matching Algorithm

This section introduces a web page similarity matching algorithm based on visual blocks. The algorithm consists of three parts. The first part is the extraction of the web page's visual information. In this part, the RenderLayer tree of the web page must be analyzed, and the visual information, including the location and size information of the visual nodes, must be extracted. The second part is the merging of visual nodes. Given the scattered numerous visual nodes of the web, the small ones can be ignored directly, while the overlapping ones require collision analysis and merge processing to maintain the consistency of the visual blocks with the original web page and improve the calculation efficiency. The last part is the web page similarity comparison, which utilizes the optimal free matching algorithm. For each visual block in the target web page, given that the visual blocks with coordinates, length, and width within a certain threshold range is found in the template web page, the blocks are considered similar. When the number of similar visual blocks is greater than a certain threshold, the two web pages are considered similar

3.1. Extracting Visual Blocks

Most web visual information are stored in the CSS file. Thus, obtaining the coordinates, lengths, and widths of the visual nodes directly from the HTML source code is difficult. The RenderLayer tree is more consistent with the visual structure of the original web page than the rendering tree. The non-visual nodes in the original web page (e.g., HEAD, META, and SCRIRT nodes) have been deleted.

The structure of the RenderLayer tree is shown in Fig 4. The first line represents the total size of the layer point, and the remaining lines signify the visual nodes of the layer point. The coordinates of the visual nodes in the layer are all relative coordinates. Thus, each visual node must convert the coordinates into the absolute coordinates of the web page.

To find the father node of each visual node, the structure of the RenderLayer tree must be analyzed. This study introduces an algorithm that converts the text form of the RenderLayer tree into an HTML source code to facilitate parsing. The conversion process is presented in Algorithm 1.

```
layer at (0,0) size 1010x6635
  RenderBlock {HTML} at (0,0) size 1010x6635 [bgcolor=#FFFFFF]
    RenderBody {BODY} at (0,0) size 1010x6635
      RenderBlock {DIV} at (0,0) size 1010x6635
        RenderBlock {DIV} at (0,0) size 1010x30 [color=#555555] [bgcolor=#FFFFFF] [border: none (
        1px solid #DDDDDD) none]
        RenderBlock {DIV} at (0,30) size 1010x149 [bgcolor=#F6F6F6]
          RenderBlock {DIV} at (10,109) size 990x40
            RenderBlock (floating) {UL} at (180,0) size 810x40 [bgcolor=#FF7300]
        RenderBlock {DIV} at (0,838) size 1010x266 [bgcolor=#FAFAFA]
          RenderBlock {DIV} at (10,20) size 990x263
```

Fig. 4. Structure of the renderlayer tree

Algorithm 1 Converting algorithm

Input: D_c: Depth of the current line, D_p: Depth of the previous line, T_c: Tag of the current line, E_c: End tag of the current line

Output: F: Format of HTML source code

```
1:  function CONVERTING
2:      for line do
3:          F.wirte(T_c)
4:          if D_c > D_p then
5:              S.push(E_c)
6:          end if
7:          if D_c < D_p and and D_p is the last line of the layer node then
8:              while ! S.empty() do
9:                  F.write(S.pop())
10:             end while
11:         end if
12:         if D_c < D_p and and D_p is not the last line of the layer node then
13:             depth = D_p − D_c
14:             while depth != 0 do
15:                 F.write(S.pop())
16:                 depth = depth - 1
17:             end while
18:         end if
19:         if D_c == D_p then
20:             F.write(S.pop())
21:             S.push(E_c)
22:         end if
23:     end for
24: end function
```

The tags of each line in the RenderLayer tree is written in the output result. The tags include the name, coordinate position, and the length-width attributes of the tag. Then, through the comparison of the depths, the relationship between the current and the previous lines can be determined. On the one hand, if the depth is larger in the current row than in the previous line, then the node is the child node of the previous ones. Thus, the tail tag is added to the stack. For example, if the tag is <HTML>, then the tail tag is also <HTML>. The stack is used to save the sequence of the nodes. On the other

hand, if the depth is smaller in the current line than in the previous line and the previous line is the last line of the RenderLayer point, then the line is the starting line of the new layer point (i.e., the previous layer point has ended). The stack stores the tag order of the previous layer point. Thus, all elements in the stack must be discarded. The tail tags are then written in the output results. However, if the depth is smaller in the current line than in the previous line, but the previous line is not the last line of the layer point, the depth difference between the current and the previous lines must be determined to select the number of element output of the stack. If the depth of the current line is the same as that of the previous line, then the current node is the brother node of the last node. The top element output of the stack is written in the output results, and the current node tag is attached to the stack. The top end tag in the stack is popped and written in the output, and then the end tag of this node is attached to the stack. The whole algorithmic process is equivalent to restoring the visual attributes to the HTML source code, thereby unifying all tag formats in the code. The output results are shown in Fig. 5.

```
<html x=0 y=0 length=1010 width=6635>
<body x=0 y=0 length=1010 width=6635>
<div x=0 y=0 length=1010 width=6635>
<div x=0 y=0 length=1010 width=30>
</div>
<div x=0 y=30 length=1010 width=149>
<div x=10 y=109 length=990 width=40>
<ul x=180 y=0 length=810 width=40>
</ul>
</div>
</div>
<div x=0 y=838 length=1010 width=266>
<div x=10 y=20 length=990 width=263>
</div>
</div>
</div>
</body>
</html>
```

Fig. 5. The output of algorithm 1

3.2. Merging Visual Blocks

Most web pages contain thousands of visual nodes. These visual nodes are disorganized and discrete, and some visual blocks are overlapped. To make the extracted visual blocks consistent with the original web pages, the visual blocks must be processed by deleting the small visual blocks (i.e., overlooking) and merging the overlapped ones. The reduction in the number of visual blocks can facilitate the improvement of computational efficiency.

Each visual block is a rectangle. The merging process of the visual blocks may be horizontal and vertical. The coordinates represent the position of a visual block in a web page. The visual blocks with proximal abscissas can be merged after the abscissa ordering. Similarly, the visual blocks with proximal ordinates can be merged after the ordinate ordering.

The current study presents a merging algorithm for visual blocks. Assuming that a list stores the visual blocks after ordering, in Algorithm 2, Pointer *i* points at the first element in the list of the visual blocks before merging, whereas Pointer *j* points at the second one. First, the visual blocks pointed by Pointers *i* and *j* must be assessed for suitability in merging. If the points are unsuitable, then both pointers are moved backward for one bit. Otherwise, the merging of two visual blocks is considered. Second, multiple overlapping similar visual blocks must be merged. Therefore, Pointer *j* must move backward for one bit until the visual blocks pointed by both pointers cannot be merged. At this time, the visual block pointed by Pointer *i* and the previous visual block pointed by Pointer *j* are merged. Pointer *i* points at Pointer *j*, which will move backward for one bit. When Pointer *j* points at the end of the list, one merge process ends. Finally, when the length of the list of two attached merging results did not change, or if no visual blocks were merged, then the algorithm ends.

Algorithm 2 merging algorithm

Input: L_{before}: A list saving visual blocks before merging, i: Pointer i, j: Pointer j
Output: L_{after}: A list saving visual blocks after merging

```
1:  function MERGING
2:      while L_after.size() not change do
3:          i ← L_before[0]
4:          j ← L_before[0]
5:          while j != L_before.size() do
6:              if !merge(i,j) then i ← i + 1 j ← j + 1
7:              end if
8:              if merge(i,j) then
9:                  while merge(i,j) do j ← j + 1
10:                 end while
11:                 L_after.add(merge(i,j-1))
12:                 i ← j
13:                 j ← j + 1
14:             end if
15:         end while
16:     end while
17: end function
```

Visual block merging mainly depends on the blocks' coordinates. Two blocks can be merged if their coordinates are adjacent. However, the merging rules vary for different cases.

Fig. 6 presents some possible cases of the horizontal merging of visual blocks. The rectangle with a thick line is called the reference block, and the one with a thin line is called the fitting block. The coordinates, lengths, and widths of the reference and fitting

blocks are $(X1, Y1)$, $a1$, and $b1$, and $(X2, Y2)$, $a2$, and $b2$, correspondingly. The conditions for horizontal merging are $Y1 = Y2$ and $b1 = b2$. If the abscissa is larger in the target block than in the reference block, then three cases are possible. The first case is where the reference and target blocks are overlapped, which is the most common case (Fig. 6(a)). The length after merging is $X2 - X1 + a2$. Fig. 6(b) illustrates the second case, wherein the target block is within the reference block because the length of the former is small. This case is generally the search box on the web page, and the length after merging is $a1$. The third case is depicted in Fig. 6(c), wherein both blocks are not overlapped. The length after merging is the same as that in the first case. In summary, the coordinates, length, and width of the visual block after merging are $(X1, Y1)$, max $(a1, X2 - X1 + a2)$, and b, respectively.

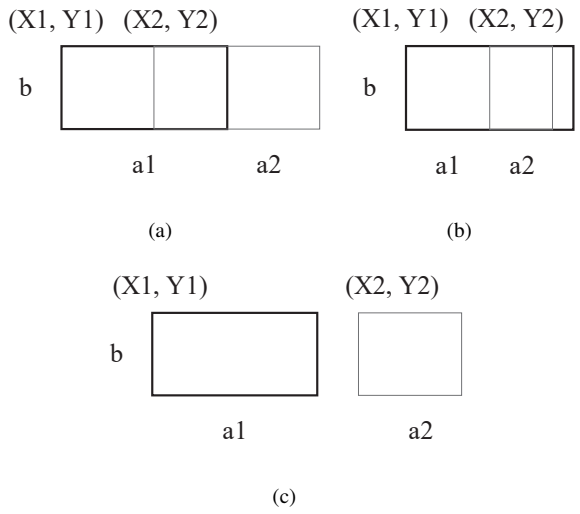


Fig. 6. Visual block merging rules

3.3. Similarity Comparison

The traditional ordered matching principle possesses many limitations. For example, the similarity of the two web pages cannot be objectively measured, and the deviation of the visual block in the web page causes errors. In addition, the similarity matching principle must have reflexivity and symmetry. Therefore, this study proposes an optimal free matching principle. For each visual block in the target web page, given a visual block with coordinates, length, and width within a certain threshold in the template web page, the blocks are considered similar.

The specific matching process is as follows:

1) Assume that (X, Y, L, W) represents the structure of each visual block, and X and Y are the abscissa and ordinate of the visual block, correspondingly, and L and W are the

respective length and width. The structure of the visual block of the target web page is (GX, GY, GL, GW), while that of the template web page is (TX, TY, TL, TW). Considering that two nodes are similar in position and the lengths and widths are only slightly different, the nodes are considered similar. In particular, when $|GX - TX|$, $|GY - TY|, |GL - TL|, |GW - TW|$ are within a certain threshold, the visual blocks are similar.

2) The nodes $\{GN1, GN2, GN3...\}$ in the target web page are compared with the nodes $\{TN1, TN2, TN3...\}$ in the template web page. When one node in the template web page is similar to that in the target page, the similar nodes can be added, and the comparison of the nodes is continued. The final number of similar nodes is denoted as SIM.
3) Assume that the total nodes of the target and template web page are G and T, respectively. In most cases, the total nodes of the target page are different from that of the template page. Therefore, the maximum values of G and T are taken. The similarity of the two web pages is denoted as $SIM/MAX(G, T)$.

4. Evaluation Result

The evaluation process consists of three parts. The first part is the evaluation of the extraction effect, which assesses if the extracted and merged visual blocks are visually consistent with the original web page. The second part evaluates the threshold of similar visual blocks. The similarity of two similar web pages (i.e., experimental objects) is compared under different thresholds. The threshold of the highest similarity is selected as the threshold of similar visual blocks. The last part examines the threshold of similar web pages, in which several similar and dissimilar web pages are regarded as datasets. The threshold with the lowest missing report and false alarm rates is taken as the threshold of similar web pages.

4.1. Extraction Effect

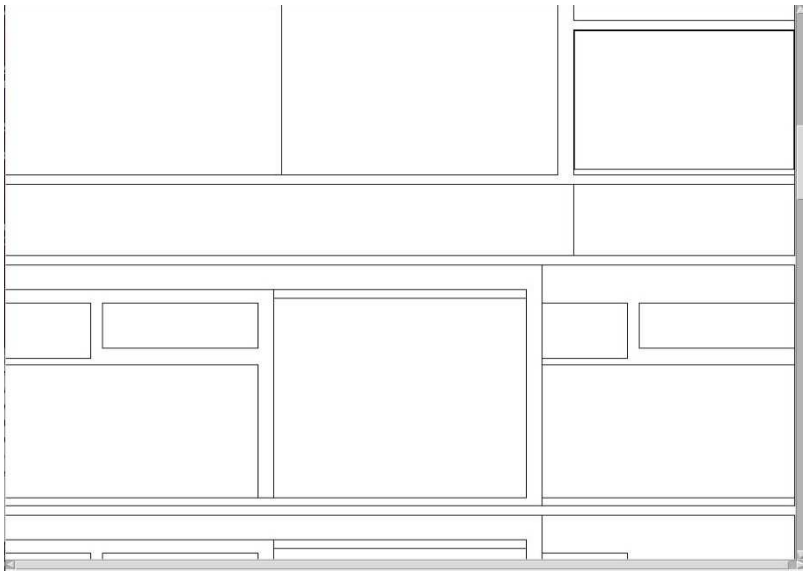
To verify if the extracted and merged visual blocks of the web page are visually consistent with the original web page, the drawing module TkInter is used to draw the visual block. This module is the interface of the standard Tk GUI toolkit and the built-in module of Python. The size of the web page is set as the size of the canvas. Then, the blocks in the visual block list are drawn on the canvas. The visual effect after drawing is compared with the original page, and the features of the web page determined by the algorithm are tested.

Fig. 7(a) presents a screenshot of Tencent. The web page is divided into blocks from the visual angle. The red line divides the web page into various visual blocks. Fig. 7(b) illustrates the extracted visual blocks and the distribution diagram. In this figure, the block division results of the web page from the visual angle and the results obtained by the algorithm are nearly the same. Therefore, the algorithm is validated.

4.2. Threshold of Similar Node

The visual block similarity threshold evaluation mainly assesses if two visual blocks are similar, and the threshold value when the web page similarity is the largest is considered the optimal result.

(a)



(b)

Fig. 7. Extraction effect result

Table 1. Visual block threshold testing table

Coordinate threshold	Length-width threshold	Similarity command
50	50	0.6216
50	60	0.6293
50	70	0.6293
...	...	...
190	180	0.9421

Table 1 shows a part of the data in the visual block similarity threshold test. When different coordinate and length/width thresholds are selected, the similarity degrees of the web pages vary. The similarity is low when the coordinate and length/width thresholds are small. The similarity degree also increases with the coordinate and the length-width thresholds. For example, when the coordinate threshold is 190 and the length/width threshold is 180, the web page similarity is the largest, that is, nearly reaching 95%. Therefore, 190 and 180 are taken as the corresponding coordinate and the length/width thresholds of the similar visual blocks.

4.3. Threshold of Similar Web Pages

The web page similarity threshold is determined as follows. Some similar and dissimilar web pages are selected as the target datasets. When the missing report and false alarm rates of similar web pages are at the minimum, the web page similarity threshold is considered the optimal value. The web pages of some post bars (e.g., Baidu post bar) are selected as the test object because of the similar typesetting. A total of 3000 URLs in the Baidu post bar are used as template web pages. These URLs are divided into 3 groups with 1000 URLs each. The final results are averaged to reduce the error. The test set includes some web pages of post bars and other web pages. If the web page is that of a post bar but is not considered similar to the template web page, then this web page is considered missing. If the web page is not a post bar web page but is regarded as similar to the template web page, then the web page is considered an error alarm.

During the comparison, the threshold of the similarity judgment starts from 0.35 and ends at 0.8, with an interval of 0.05, thereby yielding 10 similarity thresholds (Fig. 8). With the increase in the threshold, the false alarm and missing report rates decrease initially and then increase. When the threshold is 0.45, the error and miss rates obtain minimal results. Thus, 0.45 is selected as the threshold of the web page similarity. The minimal error and miss rates are 0.26 and 0.04, respectively.

Some popular phishing website detection methods, such as Kaspersky and IPDCM [21], are available; from which, we cannot publicly access codes. These methods may not achieve improved performance in this experiment because they use numerous webpage screenshots to obtain similarity with the true page and avoid text and structure similarity detections. In summary, the proposed algorithm is a more effective tool than other traditional tools in assessing and identifying phishing websites given its natural way of dealing with quality factors rather than exact values.

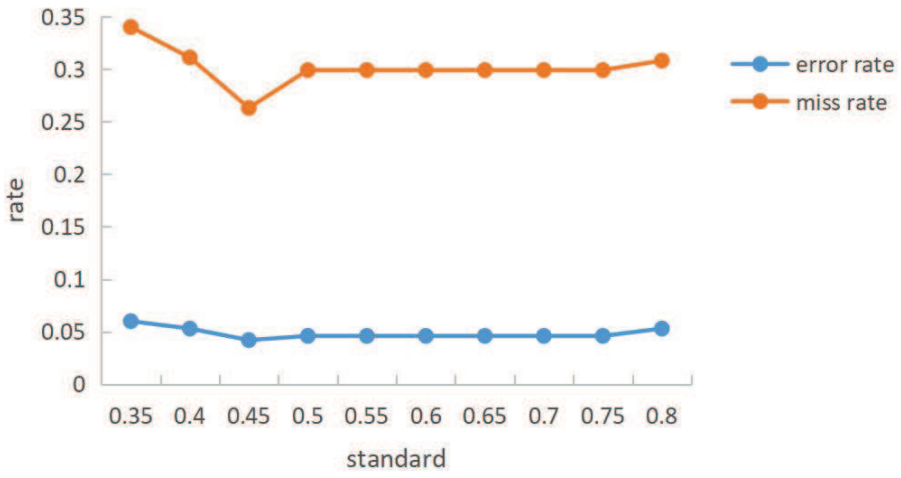


Fig. 8. Web pages similarity threshold testing result

5. Conclusion

This study investigates the visual block similarity of the web pages and proposes a new algorithm for visual block extraction and similarity comparison. The proposed algorithm analyzes the RenderLayer tree of the web page, extracts visual information by making the matching rules, determines the feature sets of the visual blocks corresponding to a web page by processing the visual nodes, and obtains similarity matching in accordance with the optimal free matching principle. After testing, the final number of the acquired visual nodes is approximately 5% of that of the original web page. In addition, to determine the threshold of the algorithm, numerous similar web pages are selected and tested under different threshold settings to achieve the optimal error and miss rates.

Acknowledgment This work was supported in part by the Key Research and Development Program for Guangdong Province (2019B010136001) and the National Key Research and Development Plan under Grant 2017YFB0801801, in part by the National Natural Science Foundation of China (NSFC) under Grant 61672186 and Grant 61872110. Professor Zhang is the corresponding author.

References

1. Alpuente, M., Romero, D.: A tool for computing the visual similarity of web pages. In: Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on. pp. 45–51. IEEE (2010)
2. Anari, Z., Anari, B.: Determining the similarity of web pages based on learning automata and probabilistic grammar. Advances in Computer Science: an International Journal 4(3), 44–53 (2015)

3. Artail, H., Fawaz, K.: A fast html web page change detection approach based on hashing and reducing the number of similarity computations. Data & Knowledge Engineering 66(2), 326–337 (2008)
4. Chiew, K.L., Chang, E.H., Tiong, W.K., et al.: Utilisation of website logo for phishing detection. Computers & Security 54, 16–26 (2015)
5. Cruz, I.F., Borisov, S., Marks, M.A., Webb, T.R.: Measuring structural similarity among web documents: preliminary results. In: Electronic Publishing, Artistic Imaging, and Digital Typography, pp. 513–524. Springer (1998)
6. Fu, A.Y., Wenyin, L., Deng, X.: Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd). IEEE transactions on dependable and secure computing 3(4), 301–311 (2006)
7. Goel, D., Jain, A.K.: Mobile phishing attacks and defence mechanisms: State of art and open research challenges. Computers & Security 73, 519–544 (2018)
8. Jain, A.K., Gupta, B.B.: Phishing detection: analysis of visual similarity based approaches. Security and Communication Networks 2017 (2017)
9. Joshi, S., Agrawal, N., Krishnapuram, R., Negi, S.: A bag of paths model for measuring structural similarity in web documents. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 577–582. ACM (2003)
10. Liu, W., Deng, X., Huang, G., Fu, A.Y.: An antiphishing strategy based on visual similarity assessment. IEEE Internet Computing 10(2), 58–65 (2006)
11. Marchal, S., Armano, G., Gröndahl, T., Saari, K., Singh, N., Asokan, N.: Off-the-hook: an efficient and usable client-side phishing prevention application. IEEE Transactions on Computers 66(10), 1717–1733 (2017)
12. Nichele, C.M., Becker, K.: Clustering web sessions by levels of page similarity. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 346–350. Springer (2006)
13. Pengyu, L., Lijun, J., Bin, J.: The research of the maximum length n-grams priority chinese word segmentation method based on corpus type frequency information. In: Proceedings Of The National Conference On Information Technology And Computer Science. pp. 71–74 (2012)
14. Roopak, S., Thomas, T.: A novel phishing page detection mechanism using html source code comparison and cosine similarity. In: Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on. pp. 167–170. IEEE (2014)
15. Satiabudhi, G., Andjarwirawan, J., Setiadi, R.S.: Web Page Similarity Searching Based on Web Content. Ph.D. thesis, Petra Christian University (2012)
16. Sonowal, G., Kuppusamy, K.: Phidma–a phishing detection model with multi-filter approach. Journal of King Saud University-Computer and Information Sciences (2017)
17. Srinivasa Rao, R., Pais, A.R.: Detecting phishing websites using automation of human behavior. In: Proceedings of the 3rd ACM Workshop on Cyber-Physical System Security. pp. 33–42. ACM (2017)
18. Varshney, G., Misra, M., Atrey, P.K.: A phish detector using lightweight search features. Computers & Security 62, 213–228 (2016)
19. Varshney, G., Misra, M., Atrey, P.K.: A survey and classification of web phishing detection schemes. Security and Communication Networks 9(18), 6266–6284 (2016)
20. Wenyin, L., Huang, G., Xiaoyue, L., Min, Z., Deng, X.: Detection of phishing webpages based on visual similarity. In: Special interest tracks and posters of the 14th international conference on World Wide Web. pp. 1060–1061. ACM (2005)
21. Zhuang, W., Jiang, Q., Xiong, T.: An intelligent anti-phishing strategy model for phishing website detection. In: International Conference on Distributed Computing Systems Workshops (2012)

Xingchen Li received the BS degree in information security from Harbin Institute of Technology, Weihai, China, in 2016 and received Master degree in Computer Science and Technology from Harbin Institute of Technology, China, in 2018. His research interests include virtualization techniques for cloud computing and information security, etc.

Weizhe Zhang is corresponding author of this paper. He is currently a professor in the School of Computer Science and Technology at Harbin Institute of Technology, China, and director in the Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China. His research interests are primarily in parallel computing, distributed computing, cloud and grid computing, and computer network. He has published more than 100 academic papers in journals, books, and conference proceedings. He is a senior member of the IEEE. Contact him at wzzhang@hit.edu.cn.

Desheng Wang received the BS degree in computer science and engineering from Harbin Engineering University, China, in 2015. He is currently working toward the PhD degree in the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include virtualization techniques for cloud computing and machine learning, etc.

Bin Zhang received his Ph.D. degree in Department of Computer Science and Technology, Tsinghua University, China in 2012. He worked as a post doctor in Nanjing Telecommunication Technology Institute from 2014 to 2017. He is now a researcher in the Cyberspace Security Research Center of Peng Cheng Laboratory. He has published more than 30 papers in refereed international conferences and journals. His current research interests focus on network anomaly detection, Internet architecture and its protocols, network traffic measurement, information privacy security, etc.

Hui He is currently a full professor of network security center in the Department of Computer Science, China. She received the Ph.D. in department of computer science from the Harbin Institute of Technology, China. Her research interests are mainly focused on distributed computing, IoT and big data analysis. She is a member of the IEEE.

Received: September 15, 2018; Accepted: July 11, 2019.

Computer Science and Information Systems 16(3):831–847 https://doi.org/10.2298/CSIS180830029Y

Privacy-preserving Multi-authority Attribute-based Encryption with Dynamic Policy Updating in PHR

Xixi Yan[1], Hao Ni[1], Yuan Liu[1], and Dezhi Han[2]

[1] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454003, China
yanxx@hpu.edu.cn
[2] College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
dzhan@shmtu.edu.cn

Abstract. As a new kind of patient-centred health-records model, the personal health record (PHR) system can support the patient in sharing his/her health information online. Attribute-Based Encryption (ABE), as a new public key cryptosystem that guarantees fine-grained access control of outsourced encrypted data, has been used to design the PHR system. Considering that privacy preservation and policy updating are the key problems in PHR, a privacy-preserving multi-authority attribute-based encryption scheme with dynamic policy updating in PHR was proposed. In the scheme, each of the patient's attributes is divided into two parts: attribute name and attribute value. The values of the user's attributes will be hidden to prevent them from being revealed to any third parties. In addition, the Linear Secret-Sharing Scheme (LSSS) access structure and policy-updating algorithms are designed to support all types of policy updating (based on "and", "or", and "not" operations). Finally, the scheme is demonstrated to be secure against chosen-plaintext attack under the standard model. Compared to the existing related schemes, the sizes of the user's secret key and ciphertext are reduced, and the lower computing cost makes it more effective in the PHR system.

Keywords: attribute-based encryption, privacy preserving, policy dynamic updating, fine-grained access control, personal health record.

1. Introduction

As a kind of patient-centred health-record model, the Personal Health Record (PHR) system has been gradually popularized. PHR is a system that is created and maintained by a patient or medical consumer, and the health information is based on his/her own health status, medications, laboratory tests, diagnostic studies, questions, allergies, vaccinations, etc. In the PHR system, a patient can maintain his/her personal health information from every channel for local storage, organization, management, sharing and tracking. Patients can access the system online anytime and anywhere. Besides, these records can help patients explain their health problems simply when they go to the doctor, and can also provide useful information when they are filing medical insurance claims.

In practical applications, to reduce the PHR centers' overhead of creation and control, large amounts of data are often outsourced to third-party PHR cloud servers. However, this comes with some additional security and privacy concerns, especially since the patients

do not want their health information to be completely transparent to the cloud server. And worst of all, once the cloud server is hacked, all the patients' health information will be leaked. In addition, when doctors, family members, insurance company staff or other users access a patient's health information, the PHR cloud server needs to interact with them to allow or restrict the visitor's access ability and scope. This will lead to lower efficiency. Obviously, considering the growth of user dimensions and data traffic, and the privacy requirements of the user's personal health information, new encryption technology and access-control mechanisms have been put forward for information management in the PHR system.

Attribute-based encryption (ABE)[1], a new public-key encryption technology, ties the user's identity with a series of attributes. The user's private key or ciphertext is defined according to the attribute set or access structure, and only when the attribute set and the access structure match can the user decrypt to obtain the message. It can guarantee the confidentiality of patient health information data in the PHR cloud server, and realize non-interactive access control with ABE. However, in practice, some sensitive private information is closely related to the patient's attributes in the access policy, so it will lead to privacy disclosure. For example, if a patient sets his/her access policy as People's Hospital, Doctor, Psychiatry, only those that match the three attributes can access his/her health records, yet this attribute information easily reveals the patient's private data. Moreover, the access policy may be changed when a patient needs a referral for treatment. If a user who satisfies Red Cross Hospital, Doctor, Psychiatry or Doctor, Psychiatry or Red Cross Hospital, Doctor, Internal Medicine should be able to access his/her health records, it will require the PHR system to support changing the access policy. Therefore, determining how to both protect privacy and support access-policy dynamic updating is a key problem when the ABE mechanism is applied in the PHR system.

On the other hand, in the pratice of ABE in PHR, user's attributes may be issued by different authorities. For instance, Alice wants to encrypt a message under access policy("DOCTORATE" and "DOCTOR"). Only the recipient who has received a doctorate and now is a doctor, can recover the message. School is responsible to issue attributes (bachelor's degree, master's degree or doctorate) to student , while hospital is responsible to issue attributes (doctor or nurse) to its employees. Therefore, the scheme in which the attribute universe is assumed to be managed by a single authority is not apply this scenarios.

In this paper, our proposed scheme aims to construct the MA-ABE scheme for PHR scenario which has the requirement for privacy preserving and policy updating. In the paper, a partially hidden policy mechanism is used to protect the user's GID and attribute privacy. Futhermore, our scheme also allows policy updating when patient wants to change the access policy associated with ciphertext.

1.1. Related Work

The traditional ABE [2-5] has only one credible organization to manage all the attributes, but in practice, the attributes are often managed by multiple authorities. Moreover, the attribute authority needs to distribute keys to all authenticated users. This will lead to a heavy workload that will become the system-performance bottleneck. Hence, Chase [6] proposed the first multi-authority attribute-based encryption scheme. In the scheme, there are a trusted central authority and some attribute authorities, and it also allows

any polynomial number of independent authorities to manage attributes and distribute privacy keys. In MA-ABE, attribute authorities take on the tasks of monitoring attributes and issuing keys. This results in a significant reduction in the system burden and high flexibility, so MA-ABE is more suitable for the PHR system. In terms of privacy protection, Chase and Chow [7] improve privacy and security in the MA-ABE scheme by removing the trusted central authority and introducing an anonymous credential to protect the users' global identifiers (GID). However, it is only supported an AND policy. Lewko and Waters [8] proposed a decentralizing attribute-based encryption scheme. In the scheme, the authorities work independently without coordination among them. It actually generalizes to handle any policy that can be expressed as a Linear Secret Sharing Scheme (LSSS) or equivalently a monotone span program. Han et al. [9] proposed a privacy-preserving decentralized key-policy attribute-based encryption scheme. But they[8-9] also consider the privacy of the GID. Recently, Xhafa et al. [10] proposed a privacy-aware MA-ABE PHR scheme that incorporates user accountability in cloud computing. In the scheme, the access policy is hidden to protect the patient's privacy. But it is only support AND gates and can't support policy updating. Han et al. [11] proposed an improved decentralized ciphertext-policy attribute-based encryption scheme, which is the first scheme to protect the privacy of attributes in MA-ABE. In the scheme, a central authority is required to generate the global key and issue secret keys to users. Commitment schemes and zero-knowledge proof technology are introduced to protect the user's GID and attributes. However, note that it has been proved not to protect the user's attributes effectively in scheme [12]. Qian et al. [13] proposed a privacy-preserving personal health record scheme using multi-authority attribute-based encryption with revocation. An anonymous key-issuing protocol, which is used to generate a secret key for the patient, is introduced to protect the privacy, but it is also only protects users' GIDs. Moreover, the scheme support policy update when patient wants to change the access policy associated with ciphertext, but simple access structures can be implemented. Xia et al. [14] proposed attribute-based access control scheme with efficient revocation in cloud computing. It could support an efficient user revocation mechanism, but it can't protect the user's attribute privacy.

The idea of policy updating for ABE originates from the delegation of the ciphertext in [15]. A proxy method is designed to update the ciphertext, but its new policy is more stringent than the original one. Yang et al. [16] proposed a scheme that supports verifiable policy-update outsourcing for big-data access control in the cloud. In this scheme, they analyzed in detail how to update the ciphertext, which is encrypted with a Boolean formula and a linear secret-sharing scheme (LSSS) structure separately. Then they proposed policy-updating algorithms for all types of policies. However, their scheme is secure only under the general group model. Ying et al. [17] proposed a partially policy-hidden CP-ABE scheme that supports dynamic policy updating. It can support any type of policy updating and the users' privacy is effectively protected by using partial policy hiding, and it is proved to be chosen-plaintext attack (CPA) secure in the standard model. Li et al. [18] proposed a scheme enabling fine-grained access control with efficient attribute revocation and policy updating in the smart grid. It not only defines the system model in the smart grid, but also leverages Third-party Auditor to support attribute revocation. Moreover, it can support dynamic policy updating and apply ABE to the smart grid effectively. Wu [19] constructed a multi-authority CP-ABE scheme

with policy updating in cloud storage. The multi-authority established by the scheme can prevent the key from being leaked and it can support policy updates under the access-tree structure. Ying et al. [20] designed an adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating. This scheme is regarded as the first MA-ABE scheme with dynamic policy updating that is proved to be adaptively secure under the standard model. A signature subsystem is introduced to resist collusion attacks. However, the system consists of multiple central authorities and multiple attribute authorities. The users' attribute keys are generated by all the central authorities, which leads to high communication cost and storage cost. Liu et al. [21] proposed an attribute-based encryption with outsourcing decryption, attribute revocation and policy updating. It is claimed that the scheme is more useful and flexible in practice, but it is lack of privacy protection. Jiang et al. [22] presented a CP-ABE supporting access policy update and its extension with preserved attributes. However , it dose not take into account the application scenarios of multi-authority.

Table 1. Comparison Among Related MA-ABE Schemes and Ours

Scheme	Multi-authority	Without an authentication center	Authority independence	Flexible access policy	Privacy protection	Policy updating
Scheme [7]	√	√	-	AND	GID	-
Scheme [8]	√	√	√	√	GID	-
Scheme [10]	√	√	√	AND	Access policy	-
Scheme [11]	√	√	√	√	GID, attribute	-
Scheme [17]	-	-	-	√	attribute	√
Scheme [18]	√	-	√	√	version number	√
Scheme [20]	√	-	√	√	GID	√
Scheme [21]	√	-	√	√	-	√
Our Scheme	√	√	√	√	GID	√

Based on the above depiction, the comparisons among related MA-ABE schemes and our proposed scheme are listed in Table 1. From the above table, it is shown that the most MA-ABE scheme towards privacy protection [7-8,11, 20] generally protect the user's privacy by protecting the user's GID, and rarely consider the privacy of attributes. Besides, the second problem is that the scheme towards policy updating [17-18,20-21] needs an authentication center which is responsible to integrate the secret keys. The authentication center can decrypt every ciphertext in the system, which would endanger the whole system if it's corrupted. Futhermore, it would also increase the computation and communication cost to run and maintain such a fully trusted authority. In contrast, our work in this paper addresses both of these limitations simultaneously.

1.2. This Study's Contribution

On addressing the above issues, we propose a privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR. It can protect the privacy of the patient's attributes and support dynamic policy updating. The theoretical innovations of this study are as follows:

(1)The privacy of attributes is considered in our scheme, and the GID is also considered as an attribute that would be protected. In the scheme, we utilize a partially hidden policy mechanism[23] to encrypt the patient's health information. Each of the patient's attributes is divided into two parts: attribute name and attribute value. The values of the user's attributes will be hidden to prevent them from being revealed to any third parties, so the user's privacy will be effectively preserved. In this way, it is more effective in attribute preservation than the traditional privacy-preserving ABE scheme, and it has lower storage cost.

(2)To meet the requirements for policy updating, we design a dynamic policy updating algorithm for Linear Secret-Sharing Scheme (LSSS) access structure in PHR. Our policy updating scheme supports the updating of any type of fine-grained policy involving 'AND', 'OR', or 'NOT', as compared to the ciphertext delegation method[15] which can only re-encrypt the ciphertext under a more restrictive policy. Other, the scheme is designed for LSSS access structure, which is more efficient than the straight-forward policy updating implementation.

(3)Our scheme has no central authority, and it has no private interactivity among attribute authorities. Unlike[17-18,20], there isn't a central authority in our scheme, while there are multiple authorities which generate user's private key by their monitored attributes without any interactivity. The security could be enhanced and the communication cost and computing cost will be reduced since the central authority is removed. Therefore, our scheme does not require central authority and it is more efficient and acceptable for real-world applications.

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we recall the basic definition of cryptographic primitives used in this paper. The definition and security model for our scheme is given in Section 3. In Section 4, a privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in PHR is proposed, and it is proved to be correct and secure in Section 5. Subsequently, we give a performance analysis in Section 6. Finally, we conclude this paper in Section 7.

2. Background

2.1. Bilinear Maps

G_0 and G_T are two multiplicative cyclic groups of prime order p.Let g_0 be the generator of G_0.There is a bilinear map,$e : G_0 \times G_0 \to G_T$, with the following properties:

(1) Bilinearity: $\forall x, y \in Z_P$,$\forall a, b \in G_0$,we have $e(a^x, b^y) = e(a, b)^{xy}$;

(2) Computability: $\forall a, b \in G_0$, we have an effective algorithm to compute $e(a, b)$;

(3) Non-degeneracy: $e(g_0, g_0) \neq 1$.

2.2. Access Structure

Suppose $\{p_1, p_2, ..., p_n\}$ is a set of parties. A set $P \subseteq 2^{\{p_1,p_2,...,p_n\}}$ is monotone if $\forall X, Y$: if $X \in P$ and $X \subseteq Y$, then $Y \in P$. An access structure is a set P of non-empty subsets of $\{p_1, p_2, ..., p_n\}$, i.e.,$P \subseteq 2^{\{p_1,p_2,...,p_n\}} \backslash \{\emptyset\}$. The collections in P are called the authorized sets, and the collections not in P are called the unauthorized sets.

2.3. Linear Secret-Sharing Scheme (LSSS)

A secret-sharing scheme over a collection P is linear if: (1) the shares for each party form a vector over Z_p ; and (2) there exists a matrix M of size $l \times n$ such that for all $i = 1, ..., l$, the i'th row is labeled with a function $\rho(i)$.Randomly choose $s \in Z_p$ and a vector $v = (s, v_2, ..., v_n) \in Z_p^n$, where s is the secret to be shared. The share $\lambda_i = M_i \cdot v$ belongs to party $\rho(i)$, where M_i is the i'th row of M.

Linear reconstruction property: Suppose a scheme's access structure is LSSS. Let S be an authorized set and $I = \{i|\rho(i) \in S\}$. There exists a set of constants $\{\omega_i \in Z_p\}_{i \in I}$ that can be used to compute the secret $s : \sum_{i \in I} \omega_i \lambda_i = s$.

2.4. Decisional q-Parallel Bilinear Diffie-Hellman Exponent Assumption (q-PBDHE)

Let G and G_T be two multiplicative cyclic groups of prime order p. Let g be the generator of G. There is a bilinear map $e : G_0 \times G_0 \rightarrow G_T$. Choose $a, s, b_1, ..., b_n \in Z_p$ and $T \in G_T$. For a tuple, $y = (g, g^s, g^a, ..., g^{(a^q)}, g^{(a^{q+2})}, ..., g^{(a^{2q})})$; $\forall 1 \leq j \leq q$ $g^{s \cdot b_j}, g^{\frac{a}{b_j}}, ..., g^{\frac{a^q}{b_j}}, g^{\frac{a^{q+2}}{b_j}}, ..., g^{\frac{a^{2q}}{b_j}}$; $\forall 1 \leq j, k \leq q, k \neq j, g^{\frac{a \cdot s \cdot b_k}{b_j}}, ..., g^{\frac{a^q \cdot s \cdot b_k}{b_j}}$). It is hard to distinguish T and $e(g, g)^{a^{q+1}S}$ in the group G_T.

We say the q-PBDHE assumption holds on the bilinear group (e, p, G, G_T) if there is no polynomial-time adversary A that can distinguish $(y, e(g, g)^{a^{q+1}S})$ and (y, T) with a negligible advantage $Adv_A = |Pr[A(y, e(g, g)^{a^{q+1}S}) = 1] - Pr[A(y, T) = 1]| \geq \varepsilon$.

3. System Model and Design Model

3.1. System Model

As shown in Figure 1, our PHR system model consists of four parts: Attribute authority, PHR cloud server, patients and PHR users. The N attribute authorities manage users' attributes and generate privacy keys. The PHR cloud server is responsible for providing health-information data-storage services and outsourcing services for the authorized user, such as ciphertext updating and fine-grained access control. The patient sends the encrypted PHR data to the PHR cloud server. If the PHR user applies for access to the patient's PHR data, the attribute authority will check whether his private-key attributes satisfy the access structure. Only an authorized user can decrypt the patient's private information. When the patient wants to alter the access policy, he only calculates the update key and sends it to the PHR cloud server. The ciphertext updating is performed by the PHR cloud server.

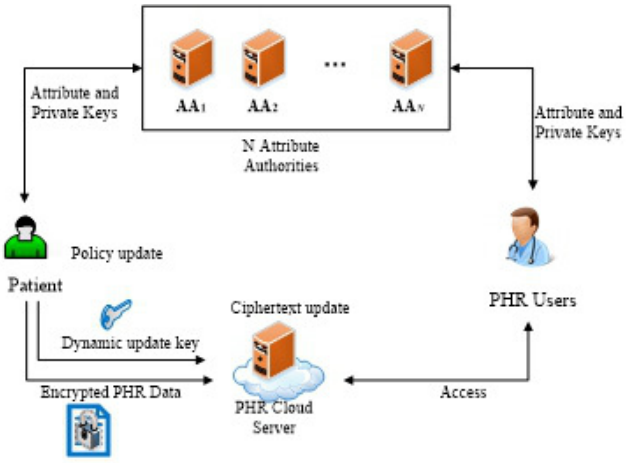


Fig. 1. System model

3.2. Definition of the System Algorithm

The scheme consists of the following algorithms:

1. $Initialization$

(1)$Global\ Setup(1^{\lambda}) \rightarrow PP$: Taking as input a security parameter 1^{λ}, the algorithm output the public parameter PP.

(2)$Authority\ Setup(PP) \rightarrow (PK_i, SK_i)$: This algorithm is run by the attribute authority. Taking the public parameter PP as input, it outputs a secret-public key pair (PK_i, SK_i) for each attribute authority.

2. $Encrypt(PP, m, (M, \rho, Z), PK_i) \rightarrow C$: This algorithm is run by the patient. Taking as input the public parameter PP, the PHR message m, access structure (M, ρ, Z), and the attribute-authority public keys PK_i, it outputs the ciphertext C, and the encryption information $En(m)$ is preserved by the patient.

3. $KeyGen(PP, SK_i, A_u) \rightarrow SK_u$: This algorithm inputs the public parameter PP, the attribute-authority secret keys SK_i, and the user's attributes A_u,and outputs the user's privacy key SK_u.

4. $Decrypt(PP, C, SK_u) \rightarrow m$: This algorithm inputs the public parameter PP, the user's privacy key SK_u, and the ciphertext C and outputs the message m.

5. $Policyupdate$

(1)$DK_mGen(En(m), (M', \rho', Z'), (M, \rho, Z)) \rightarrow DK_m$: This algorithm is run by the patient. Taking as input the encryption information $En(m)$ of m, the current policy (M, ρ, Z) and the new policy (M', ρ', Z'), it outputs the dynamic policy-update key DK_m.

(2)$CUpdate(C, DK_m) \rightarrow C'$: This algorithm is run by the PHR cloud server. Taking as input the current ciphertext C and the dynamic policy-update key DK_m, it outputs the new ciphertext C'.

3.3. Security Model

The semantic security against chosen-plaintext attack (CPA) is modeled in the selective access structure model (SAS), in which the adversary must provide the access structure he wishes to attack before he receives the public parameters from the challenger. The game is carried out between a challenger and an adversary. The game is as follows.

Initialization : The adversary A sends the challenge access structure (M^*, ρ^*, Z^*) to the challenger.

Global Setup : The challenger runs the Global Setup algorithm to generate the public parameter PP and sends it to A.

Authorities Setup : The challenger runs the Authorities Setup algorithm to generate the attribute authority's secret-public keys and sends them to A.

Phase 1 : The adversary makes many attribute private-key queries and determines which attributes do not appear in (M^*, ρ^*, Z^*). The challenger generates the privacy key and sends it to A.

Challenge : The adversary submits two equal-length messages $m0$ and $m1$. The challenger randomly chooses and encrypts mc under (M^*, ρ^*, Z^*). The ciphertext is given to A.

Phase 2 : Phase 1 is repeated.

Guess : outputs his guess c' on c.

Definition 1: Our scheme is secure if any probably polynomial-time adversary A making q secret-key queries can win the above game with a negligible advantange $\varepsilon = |Pr[c = c'] - \frac{1}{2}|$.

4. Scheme of the PHR System

Table 2. Notations

Symbols	Definition
Z_q	An integer set of mod q
$AA_i(i = 1, ..., n)$	Attribute authority
$a_i(i = 1, ..., n)$	Attribute name
$a_{i,j}(i = 1, ..., n, j = 1, ..., n_i)$	Attribute value
$(M, \rho, Z)(M', \rho', Z')$	Old access policy and new access policy. M is a matrix, ρ is a function which maps each row M_i to an attribute name, Z denotes the set of attribute values that the patient designs and hides in the policy. (M', ρ', Z') is defined similar to (M, ρ, Z)
PP	Public parameter
PK_i, SK_i	The secret-public key pair for each attribute authority AA_i
SK_u	The user's privacy key
$En(m)$	The encryption information of m, and m is the patient's health data

Before the PHR system operation, we assume that the universe of attributes consists of n attribute names and $n = (a_1, a_2, \ldots, a_n)$ such that each a_i has n_i attribute values and $A_i = (a_{i,1}, a_{i,2}, ..., a_{i,n_i})$. We use $A_u = (a_1 : a_{1,t_1}, a_2 : a_{2,t_2}, ..., a_n : a_{n,t_n})$ to denote the user's attribute set. It is obvious that $a_{i,t_i} \in A_i$. Suppose that there are N authorities $\{AA_1, AA_2, ..., AA_N\}$ and AA_i monitors for attribute name a_i and a set of attribute values corresponding to a_i.

A patient's access structure is defined as (M, ρ, Z). M is a matrix with l rows and n columns. The function maps each row M_i to an attribute name. Z denotes the attribute values that the patient designs and hides in the policy, and $Z = (Z_{\rho(1)}, ..., Z_{\rho(l)})$. A PHR user's attribute set matches (M, ρ, Z) if and only if for all $i \in \{1, ..., l\}$, $a_{\rho(i)} = Z_{\rho(i)}$, and there exist constants $\omega_i \in Z_p$ such that $\sum_{i \in I} \omega_i \lambda_i = s$.

The algorithms of our N-authority CP-ABE scheme are presented in the rest of the section.

4.1. Initialization

1)$Global\ Setup(1^\lambda) \rightarrow PP$: Taking as input a security parameter 1^λ, the algorithm outputs the public parameter $PP = (e, p, g, h, G, G_T)$, where $e : G \times G \rightarrow G_T$. G and G_T are two multiplicative cyclic groups of prime order p. Let g and h be the generators of G.

2)$Authority\ Setup(PP) \rightarrow (PK_i, SK_i)$: This algorithm is run by the attribute authority. Each attribute authority AA_i chooses $\alpha_i \in Z_p$ and computes

$$A_i = e(g, g)^{\alpha_i} \tag{1}$$

For attribute name a_i and each attribute value a_{i,n_i}, attribute authority AA_i chooses $r_i, t_{i,n_i} \in Z_p$ and computes

$$R_i = g^{r_i} \tag{2}$$

$$T_{i,n_i} = g^{t_{i,n_i}} \tag{3}$$

Then, the attribute authority AA_i publishes the public key $PK_i = \{A_i, R_i,$ $(T_{i,n_i})_{\forall a_{i,n_i}} \in A_i\}$ and keeps the private key $SK_i = \{\alpha_i, r_i, (t_{i,n_i})_{\forall a_{i,n_i}} \in A_i\}$.

4.2. Encrypt

This algorithm is run by the patient. The patient encrypts his or her PHR data with an access policy (M, ρ, Z). The ciphertext is sent to the PHR cloud server. The algorithm is run as follows:

Step 1: The patient chooses $s \in Z_p$ and a vector $v = (s, v_2, ..., v_n) \in Z_p^n$ randomly.

Step 2: For each row of the matrix, compute $\lambda_i = M_i \cdot v$.

Step 3: The patient chooses $q_1, ..., q_l \in Z_p$ randomly and computes

$$c_0 = m \cdot \prod\nolimits_{i \in I_A} e(g, g)^{a_i s} \tag{4}$$

$$c_1 = g^s \tag{5}$$

for $i \in (1, ..., l)$

$$C_{2,i} = h^{\lambda_i}(T_{\rho(i)}^{Z_{\rho(i)}})^{-q_i} \tag{6}$$

I_A is a set that consists of the indices of the authorities that are selected to encrypt m.

Step 4: Finally, the patient sends the ciphertext $C = \{C_0, C_1, (C_{2,i}, C_{3,i})_{i\in(1,\ldots,l)}\}$ to the PHR cloud server, and the encryption information $En(m)$ is preserved by the patient.

4.3. KeyGen

When a doctor or other PHR users request access to the patient's PHR data, the $KeyGen$ algorithm is run to generate the doctor's privacy key. This algorithm is executed by the attribute authority. AA_i checks whether it handles the doctor's attribute set A_u. If this is the case, it computes

$$D_i = g^{\alpha_i} h^{r_i} \tag{7}$$

$$R_{i,t_i} = (T_{i,t_i}^{a_{i,t_i}})^{r_i} \tag{8}$$

Then AA_i sends the user's privacy key $SK_u = \{D_i, R_{i,t_i}\}_{1\le i\le n}$ to the doctor. Otherwise, it outputs NULL.

4.4. Decrypt

When the doctor obtains his privacy key, he can decrypt the ciphertext. The Decrypt algorithm is executed as follows:

Step 1: The system checks whether the conjunction of (M, ρ, Z) can be satisfied by the doctor's attributes.

Step 2: If this is the case, it computes

$$\begin{aligned} e(C_{2,i}, R_i) &= e(h^{\lambda_i}(T_{\rho(i)}^{Z_{\rho(i)}})^{-q_i}, g^{r_i}) = e(h^{\lambda_i}(g^{t_{\rho(i)}Z_{\rho(i)}})^{-q_i}, g^{r_i}) \\ &= e(h,g)^{\lambda_i r_i} e(g,g)^{-q_i t_{\rho(i)} Z_{\rho(i)} r_i} \end{aligned} \tag{9}$$

$$e(C_{3,i}, R_{i,n_i}) = e(g^{q_i}, (T_{\rho(i)}^{u_{\rho(i)}})^{r_i}) = e(g,g)^{q_i t_{\rho(i)} Z_{\rho(i)} r_i} \tag{10}$$

Step 3:Finally, he computes $m = \frac{C_0 \cdot \prod_{i\in I_A}(e(C_{2,i},R_i)e(C_{3,i},R_{i,n_i}))^{\omega_i}}{\prod_{i\in I_A} e(C_1,D_i)}$, where $\sum_{i=1}^{l} \omega_i \lambda_i = s$.

4.5. Policy Update

In this subsection, we describe the access policy update in detail. When a patient needs to convert a current access structure (M, ρ, Z) to a new one (M', ρ', Z'), he only needs to use the encryption information $En(m)$ and run the DK_mGen algorithm to construct the update keys and send them to the PHR cloud server. The PHR cloud server will run the $CUpdate$ algorithm to update the ciphertext. The algorithm is defined as follows.

1) DK_mGen: This algorithm is run by the patient. Taking as input the encryption information $En(m)$ of m, the current policy (M, ρ, Z) and the new policy (M', ρ', Z'), it

outputs the dynamic policy-update key DK_m. M' is a new $l' \times n'$ matrix with ρ' mapping the rows to the attribute names , and Z' is the new attribute values.

Step 1: It first executes the policy-comparison algorithm to compare the new access policy (M', ρ', Z') with the current one (M, ρ, Z), and outputs three sets of row indices A'_1,A'_2, and A'_3 of M'. Let $n_{\rho(i),M}$ and $n_{\rho(i),M'}$ denote the numbers of attribute names $\rho(i)$ in M and M', respectively. A'_1 and A'_2 represent the sets of indices j where $\rho(j)'$ exists in M and $\rho(i) = \rho(j)'$. If $n_{\rho(j)',M'} \leq n_{\rho(j)',M}$, the indices j are put into A'_1. If $n_{\rho(j)',M'} > n_{\rho(j)',M}$, will include the indices j that exceed $(n_{\rho(j)',M'} - n_{\rho(j)',M})$. A'_3 denotes the set of indices j such that $\rho(j)$ is a new attribute name.

Step 2: The DK_mGen algorithm chooses a random vector $v' \in Z_p^{n'}$ and s is the first entry. Let $\lambda'_j = M'_j \cdot v'$, and M'_j is the j'th row of M'. For each $j \in [1, l']$, the update key can be divided into three types.

(1) Type 1: If $(j, i) \in A'_1$, the update key will be $DK = (DK = h^{\lambda'_j - \lambda_i})$,and let $q'_j = q_i$:

(2) Type 2: If $(j, i) \in A'_2$,the algorithm chooses $x_j, q'_j \in Z_p$ randomly and computes the update key $DK = (x_j, DK = h^{\lambda'_j - x_j \lambda_i})$;

(3) Type 3: If $(j, i) \in A'_3$,the algorithm chooses $q'_j \in Z_p$ and computes the update key $DK = (DK(1) = h^{\lambda'_j}(T_{\rho'(j)}^{Z'_\rho(j)})^{-q'_j}, DK(2) = g^{q'_j})$.

Step 3: Finally, the patient sends the update key $DK_m = \{(DK)_{(j,i)\in A'_1}$, $(DK)_{(j,i)\in A'_2}, (DK)_{(j,i)\in A'_3}\}$to the PHR cloud server.

2) $CUpdate$: This algorithm is run by the PHR cloud server. Taking as input the current ciphertext C and the dynamic policy-update key DK_m , it outputs the new ciphertext C'.

(1) Type 1: For each $j \in A'_1$, it computes the ciphertext elements

$$C'_{2,j} = C_{2,i} \cdot DK = h^{\lambda'_j}(T_{\rho'(j)}^{Z'_\rho(j)})^{-q'_j} \tag{11}$$

$$C'_{3,j} = C_{3,j} = g^{q'_j} \tag{12}$$

where $q'_j = q_i$;

(2) Type 2: For each $j \in A'_2$, it computes

$$C'_{2,j} = (C_{2,i})^{x_j} \cdot DK = h^{\lambda'_j}(T_{\rho'(j)}^{Z'_\rho(j)})^{-q'_j} \tag{13}$$

$$C'_{3,j} = g^{q'_j} \tag{14}$$

where $q'_j = x_j q_i$;

(3) Type 3: For each $j \in A'_3$, it computes

$$C'_{2,j} = DK(1) = h^{\lambda'_j}(T_{\rho'(j)}^{Z'_\rho(j)})^{-q'_j} \tag{15}$$

$$C'_{3,j} = DK(2) = g^{q'_j} \tag{16}$$

Finally, the new ciphertext is $C' = \{C_0, C_1, (C'_{2,j}, C'_{3,j})_{j\in(1,\ldots,l')}\}$

5. Security Analysis

In this section, our scheme is demonstrated to be secure against chosen-plaintext attack under the standard model. The specific certification process is as follows.

Definition 2: Our scheme is secure in the selective-access structure and chosen-plaintext attack game if the decisional q-PBDHE assumption holds. Then, no polynomial-time adversary can break our scheme with a challenge structure (M^*, ρ^*, Z^*).

Initialization: The challenger initiates a q-parallel BDHE challenge (y, T). The adversary submits the challenge structure (M^*, ρ^*, Z^*), where M^* is an $l^* \times n^*$ matrix and $l^*, n^* \leq q$.

Global Setup: The challenger chooses random $m \in Z_p$ and computes $h = g^m$. Then it sends the public parameter $PP = (e, p, g, h, G, G_T)$ to A.

Authority Setup: For every AA_i, it chooses random $\alpha_i' \in Z_p$ and sets $\alpha_i = \alpha_i' + a^{q+1}$. Then, it computes

$$A_i = e(g,g)^{\alpha_i} = e(g,g)^{\alpha_i'} e(g^a, g^{a^q}) \tag{17}$$

Let X be the set of the indices i with $\rho^*(i) = x$ for $i = 1, ..., l^*$. For each attribute name a_i with $\rho^*(i) = x$,it chooses random $r_i \in Z_p$ and computes

$$R_i = g^{r_i} \prod_{i \in X} g^{aM_{i,1}^*/b_i} \cdot g^{a^2 M_{i,2}^*/b_i} \cdot ... \cdot g^{a^{n^*} M_{i,n^*}^*/b_i} \tag{18}$$

For each attribute name a_i with $\rho^*(i) \neq x$,it chooses random $r_i \in Z_p$ and computes $R_i = g^{r_i}$. For each attribute value a_{i,n_i}, it chooses random $t_{i,n_i} \in Z_p$ and computes $T_{i,n_i} = g^{t_{i,n_i}}$. The master secret key of AA_i is $SK_i = \{a_i, r_i, (t_{i,n_i})_{\forall a_{i,n_i} \in A_i}\}$ and the public key is $PK_i = \{A_i, R_i, (T_{i,n_i})_{\forall a_{i,n_i} \in A_i}\}$. The challenger sends PK_i to the adversary A.

Phase 1: The adversary A can adaptively query the secret key for a set $S = (a_i : a_{i,t_i})$, where S does not satisfy (M^*, ρ^*, Z^*). The challenger first checks the set S. For each attribute name a_i with $\rho^*(i) = x$, it computes

$$D_i = g^{\alpha_i'} g^{a^{q+1}} g^{mr_i} \prod_{i \in X} g^{aM_{i,1}^*/b_i} \cdot g^{a^2 M_{i,2}^*/b_i} \cdot ... \cdot g^{a^{n^*} M_{i,n^*}^*/b_i} \tag{19}$$

For each attribute value a_{i,t_i}, it computes

$$R_{i,t_i} = (T_{i,n_i}^{a_{i,t_i}})^{r_i} \tag{20}$$

For each attribute name a_i with $\rho^*(i) \neq x$, it computes

$$D_i = g^{\alpha_i} h^{r_i} = g^{\alpha_i'} g^{a^{q+1}} g^{mr_i} \tag{21}$$

For each attribute value a_{i,t_i}, it computes $R_{i,t_i} = (T_{i,n_i}^{a_{i,t_i}})^{r_i}$. Then the challenger sends the privacy key $SK_A = \{D_i, R_{i,t_i}\}_{a_{i,t_i} \in S \cap A_i}$ to the adversary.

Challenge: The adversary submits two equal-length messages m_0 and m_1. The challenger chooses random $c \in \{0, 1\}$, and computes

$$C_0 = m_c \cdot T \cdot \prod\nolimits_i e(g,g)^{\alpha_i' S} \tag{22}$$

$$C_1 = g^S \tag{23}$$

Then it chooses $v^* = (s, sa + v_2^*, sa^2 + v_3^*, ..., sa^{n-1} + v_{n^*}^*) \in Z_p^{n^*}$ and $q_1^*, ..., q_{l^*}^* \in Z_p$. For $i = 1, ..., n^*$, we define H_i as the set of all $i \neq j$ such that $\rho^*(i) = \rho^*(j)$. The challenge ciphertext elements are

$$\begin{aligned} C_{2,i} &= (T_{\rho^*(i)}^{Z_{\rho^*(i)}})^{q_i^*} (\prod_{j=1,...,n^*} (h)^{M_{i,j}^* v_j^*})(g^{b_i s})^{-t_{\rho^*(i)}} (\prod_{k \in H_i} \prod_{j=1,...,n^*} (g^{a^j s \cdot (b_i/b_k)})^{M_{k,j}^*}) \\ &= T_{\rho^*(i)}^{Z_{\rho^*(i)}})^{q_i^*} (\prod_{j=1,...,n^*} (g^m)^{M_{i,j}^* v_j^*})(g^{b_i s})^{-t_{\rho^*(i)}} (\prod_{k \in H_i} \prod_{j=1,...,n^*} (g^{a^j s \cdot (b_i/b_k)})^{M_{k,j}^*}) \end{aligned} \tag{24}$$

$$C_{3,i} = g^{-q_i^*} g^{-s b_i} \tag{25}$$

The ciphertext $C = \{C_0, C_1, (C_{2,i}, C_{3,i})_{i \in (1,...,l^*)}\}$ is given to A.

Phase 2: Phase 1 is repeated.

Guess: outputs his guess c' on c. The challenger outputs $\theta = 0$ to guess that $T = e(g,g)^{a^{q+1} s}$ if $c' = c$. Therefore, the adversary can output $c' = c$ with the advantage $Pr[c' = c|\theta = 0] = 1/2 + \varepsilon$. Otherwise, it outputs $\theta = 1$ to guess that T is a random group element in G_T, and the advantage is $Pr[c' = c|\theta = 1] = 1/2$.

Therefore, the adversary can break the decisional q-PBDHE assumption with a negligible advantage $|\frac{1}{2} Pr[c' = c|\theta = 0] + \frac{1}{2} Pr[c' = c|\theta = 1] - \frac{1}{2}| = \frac{1}{2} \times \frac{1}{2} + \frac{1}{2} \times (\frac{1}{2} + \varepsilon) - \frac{1}{2} = \frac{1}{2}\varepsilon$.

6. Performance Analysis

As shown in Table 3 and Table 4, we compare our scheme with the previous methods [8, 10, 11, 15-17, 19] in terms of system function, communication cost and security model. The function includes whether it supports multiple authorities, privacy protection and dynamic policy updating. The communication cost refers to the sizes of the user's secret keys, ciphertext and dynamic update keys. Let S be the number of user attributes, D be the number of CAs, N be the number of AAs, n be the number of attributes of the system,ρ be the bit length of the PHR user's GID, l be the number of attributes encrypted by the patient, I_c be the number of AA_i encrypted by the patient, k be the number of attributes decrypted by the PHR user, and n_i be the number of attributes managed by AA_i. The decrypted cost is the number of bilinear pairing operations, and l_1', l_2', andl_3' are the numbers of attributes of types 1, 2, and 3, respectively.

From Table 3, we can see that our scheme supports the protection of attribute privacy and dynamic policy updating. The schemes in [8, 10, 11] have privacy protection but they can't support policy updating. In the schemes in [8, 12, 17], only the privacy of the GID has been considered. Obviously, they can't protect the privacy of attributes. The whole access policy is hidden to protect privacy in the scheme in [10]. The scheme in [11] can protect the privacy of the GID and attributes. This is the first scheme that considers the privacy of attributes. However, it has been proved in the scheme in [12] that the scheme proposed in [11] can't effectively protect the user's attributes. In our scheme, not only the privacy of GID has been considered, but also the attribute values are hidden in the access structure to prevent the attributes from being revealed to others. In terms of security, the

Table 3. Comparison of Function and Security Model

Scheme	Function			Security model
	Multi-authority	Privacy protection	Policy updating	
Scheme [8]	$\surd$	Protect user's GID	$\times$	Random Oracle
Scheme [10]	$\surd$	Hide the access policy to protect privacy	$\times$	Standard
Scheme [11]	$\surd$	Protect user's GID and attribute	$\times$	Standard
Scheme [15]	$\surd$	Protect user's GID	$\surd$	Random Oracle
Scheme [16]	$\times$	Protect user's attribute	$\surd$	Standard
Scheme [17]	$\surd$	Protect version number	$\surd$	Standard
Scheme [19]	$\surd$	Protect user's GID	$\surd$	Standard
Our Scheme	$\surd$	Protect user's GID and attribute	$\surd$	Standard

schemes in [8,15] are only secure under the random oracle model. Our scheme and the schemes in [10, 11, 16, 17, 19] are secure under the standard model.

Table 4. Comparison of Communication Cost

Scheme	Communication cost					
	The size of user's privacy key	The size of ciphertext	The size of dynamic update key			Decryption cost
			Type 1	Type 2	Type 3	
Scheme[8]	S	$3l+1$	$\times$	$\times$	$\times$	$2k$
Scheme[10]	$4S+4\rho+1$	$4n+4\rho+2$	$\times$	$\times$	$\times$	$\sum_{i\in[N]} 4n_i+4\rho+2$
Scheme[11]	$S+6$	$(2l+3)I_c+1$	$\times$	$\times$	$\times$	$2k+5I_c$
Scheme[15]	S	$3l+1$	$2l'_1$	$3l'_2$	$3l'_3$	$2k$
Scheme[16]	$S+2$	$2(2l+2)$	$5l'_1$	$6l'_2$	$4l'_3$	$4k+2$
Scheme[17]	$2S$	$3l+2$	$2l'_1$	$3l'_2$	$3l'_2$	$2k+1$
Scheme[19]	$S+D(N+2)$	$2l+2$	l'_1	$2l'_2$	$2l'_3$	$2k+1$
Our Scheme	$2S+1$	$2l+2$	l'_1	$2l'_2$	$2l'_3$	$2k+2I_c$

The complexity comparison of communication cost is presented in Table 4. The user's private key in the schemes in [8, 11, 15, 16] is relatively small, but the ciphertext is relatively long. The whole communication cost is high in the scheme in [10]; particularly, the ciphertext length is related to the number of attributes of the system, while in the other schemes, it is related to the number of encrypted attributes in the access policy. Considered the policy updating, the schemes in [8, 10, 11] do not provide the policy-updating function, and the update keys in our scheme are shorter than those in [15, 16, 17] and the same length as those in the scheme in [19]. Moreover, the decryption cost of the scheme in [10] is related to the number of system attributes and user GID length, and the cost is higher. In [11] and in our scheme, the decryption cost is related to the number of encrypted attributes and the number of encrypted attribute authorities, but our

scheme has $3I + c$ lower cost than that in [11]. Compared with the scheme in [19], the length of the user's keys in our scheme is much smaller, because it is only related to the number of user attributes. Instead, the user's private keys in [19] are generated by the D central authorities and N attribute authorities. The product of D and N is multiples of S, therefore, the length of the user's keys in our scheme is much smaller. The size of ciphertext and dynamic update key are same. As a whole, it is showed that our scheme has optimized in terms of both the communication and computational costs.

Based on the comprehensive analysis, our PHR system scheme can protect the patient's attributes and support access-policy dynamic updating. Moreover, the proposed scheme also has certain advantages in performance when compared with other schemes. It is applicable to the PHR system, which needs to protect the privacy of the user and update the access policy.

7. Conclusions

To protect the patient's attribute privacy and support policy updating in the PHR cloud environment, a privacy-preserving multi-authority attribute-based encryption scheme with dynamic policy updating in PHR is proposed. A semi-policy-hidden technology is introduced to protect the privacy of the patient's attributes. The access policy of ciphertext updating is divided into three types, and for each type, there is a method to update the policy. In this paper, a method of access-policy dynamic updating is designed, with which the patient only needs to send the update key to the PHR cloud server. The implementation delegates the most computationally intensive jobs to the PHR cloud server, so the communication cost and computational cost of our system are greatly reduced. The users not only require strong functioning of the system, but also high performance in the big-data environment. Therefore, a secure MA-ABE scheme with shorter ciphertext, shorter private keys and richer expression ability would be worthwhile to investigate in our future work.

Acknowledgments This work was supported in part by the "13th Five-Year" National Crypto Development Foundation under Grant MMJJ20170122, in part by the National Natural Science Foundation of China under Grant 61802117, and in part by the Projects of Henan Provincial Department of Science and Technology under Grant 192102210280.

References

1. Sahai, A, Waters, B.: Fuzzy identity-based encryption. Advances in Cryptology - EUROCRYPT 2005. Berlin, Heidelberg, Springer - Verlag, 457-473. (2005)
2. Goyal, V, Pandey, O, Sahai, A, et al.: Attribute-based Encryption for Fine Grained Access Control of Encrypted Data. Proceedings of the ACM Conference on Computer and Communications Security (CCS 2006). ACM Press, 89-98. (2006)
3. Agrawal, S, Boyen, X, Vaikuntanathan, V, et.al.: Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. Proceedings of Public Key Cryptography (PKC 2012). Berlin: Springer-Verlag, 280–297. (2012)
4. Boyen, X.: Attribute-Based Functional Encryption on Lattices. Proceedings of the 10th theory of cryptography conference on Theory of Cryptography. Berlin: Springer-Verlag, 122-142. (2013)

5. Zhang, L, Wu, Q, Mu, Y, et al.: Privacy-Preserving and Secure Sharing of PHR in the Cloud. Journal of Medical Systems, Vol. 40, No. 12, 1-13. (2016)
6. Chase, M.: Multi-authority attribute based encryption. Proceedings of Theory of Cryptography Conf. (TCC '07), S.P. Vadhan, ed., 515-534. (2007)
7. Chase, M, Chow, S.: Improving privacy and security in multi-Authority attribute-Based encryption. Proceedings of the 16th ACM confenence of computer and communication Security. E. Al-Shaer, S. Jha, and A.D. Keromytis, eds., 121-130. (2009)
8. Lewko, A, Waters, B.: Decentralizing attribute - based encryption. Proceedings of Eurocrypt 2011: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology, K.G. Paterson, ed., 568-588 . (2011)
9. Han, J, Susilo, W, Mu, Y, et al.: Privacy-Preserving Decentralized Key-Policy Attribute-Based Encryption. IEEE Transactions on Parallel & Distributed Systems, Vol. 23, No. 11, 2150-2162. (2012)
10. Xhafa, F, Feng, J, Zhang, Y, et al.: Privacy-aware attribute-based PHR sharing with user accountability in cloud computing. Journal of Supercomputing, Vol. 71, No. 5, 1607–1619. (2015)
11. Han, J, Susilo, W, Mu, Y, et al.: Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE Transactions on Information Forensics and Security, Vol. 10, No. 3, 665-678. (2015)
12. Wang, M, Zhang, Z, Chen, C.: Security analysis of a privacy-preserving decentralized ciphertext-policy attribute-based encryption scheme. Concurrency & Computation Practice & Experience, Vol. 28, No. 4, 1237-1245. (2016)
13. Qian, H, Li, J, Zhang, Y, et al.: Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation. International Journal of Information Security, Vol. 14, No. 6, 487-497. (2015)
14. Xia, Z, Zhang, L, Liu. D.: Attribute-based access control scheme with efficient revocation in cloud computing. China Communications, Vol. 13, No. 7, 92-99. (2016)
15. Sahai, A, Seyalioglu, H, Waters, B.: Dynamic credentials and ciphertext delegation for attribute-based encryption. Advances in Cryptology – CRYPTO 2012. Springer Berlin Heidelberg, 199-217. (2012)
16. Yang, K, Jia, X, Ren, K.: Secure and verifiable policy update outsourcing for big data access control in the cloud. IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 12, 3461-3470. (2015)
17. Ying, Z, Ma, J, Cui, J.: Partially policy hidden CP-ABE supporting dynamic policy updating. Journal on Communications. Vol. 36, No. 12, 178-189. (2015)
18. Li, H, Liu, D, Alharbi, K, et al. Enabling fine-grained access control with efficient attribute revocation and policy updating in smart grid[J]. KSII Transactions on Internet and Information Systems, Vol, 9, No. 4, 1404-1423. (2015)
19. Wu, G.: Multi-Authority CP-ABE with policy update in cloud storage. Journal of Computer Research and Development. Vol. 53, No. 10, 2393-2399. (2016)
20. Ying, Z, Li, H, Ma, J, et al.: Adaptively secure ciphertext-policy attribute-based encryption with dynamic policy updating. Science China-Information Sciences, Vol. 59, No. 4, 1-16. (2016)
21. Liu ZC, Jiang ZL, Wang X, et al.:Practical attribute-based encryption: outsourcing decrtyption, attribute revocation and policy updating. Journal of Network and Computer Applications, Vol. 108, 112-123.(2018)
22. Jiang YH, Susilo W, Mu Y, et al.:Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes. International Journal of Information Security, Vol. 17, No. 5, 533-548.(2018)
23. Lai, J, Deng, H, Li, Y.: Expressive CP-ABE with partially hidden access structures. Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. ACM, 18-19. (2012)

Xixi Yan, received her Ph.D. degree from Beijing University of Posts and Telecommunications. She is currently an associate professor in the School of Computer Science and Technology, Henan Polytechnic University. She is mainly engaged in the digital content security and cloud computing.

Hao Ni, received his Bachelor of Engineering degree from Henan Polytechnic University. At present, he is a postgraduate of Henan Polytechnic University, and mainly engaged in Network and Information Security, Cryptography.

Yuan Liu, received her Bachelor of Engineering degree from Henan Polytechnic University. At present, she is a doctoral student of Beijing University of Posts and Telecommunications, and mainly engaged in Network and Information Security, Cryptography.

Dezhi Han (corresponding author), received the Ph.D. degree from Huazhong University of Science and Technology. He is currently a professor of computer science and engineering at Shanghai Maritime University. His research interests include cloud computing, mobile networking and cloud security.

Received: August 30, 2018; Accepted: May 25, 2019.

Computer Science and Information Systems 16(3):849–865 https://doi.org/10.2298/CSIS181002030C

Classification and Analysis of MOOCs Learner's State: The Study of Hidden Markov Model

Haijian Chen[1], Yonghui Dai[2], Heyu Gao[3], Dongmei Han[3], and Shan Li[2]

[1] Institute of science and technology, Shanghai Open University, Shanghai 200433, China
xochj@shtvu.edu.cn
[2] Management School, Shanghai University of International Business and Economics, Shanghai 201620, China
daiyonghui@suibe.edu.cn, 380734913@qq.com
[3] School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China
gaoheyu1993@163.com, dongmeihan@mail.shufe.edu.cn

Abstract. In MOOCs, learner's state is a key factor to learning effect. In order to study on learner's state and its change, the Hidden Markov Model was applied in our study, and some data of learner were analyzed, which includes MOOCs learner's basic information, learning behavior data, curriculum scores and data of participation in learning activities. The relationship of the learning state, the environment factors and the learner's individual conditions was found based on the data mining of the above of learning behavior data. Generally, there are three main conclusions in our research. Firstly, learners with different educational background have different learning states when they first learn from MOOCs. Secondly, the environmental factors such as curriculum quality, overall learning status and number of learners will influence the change of learners' learning status. Thirdly, the learner's behavioral expression is an observational signal of different learning states, which can be used to detect and manage the learner's learning states in different periods. From the analysis results of Hidden Markov Model, it is found that learners in different learning states can adopt appropriate methods to improve their learning efficiency. If the learner is in a negative state, the learning efficiency can be improved by improving the learning environment. If the learner is in a positive state, the positive learning status of the surrounding learners can help him or her maintain current state. Our research can help the MOOCs institutions improve the curriculum and provide reference for the development of MOOCs teaching.

Keywords: Classification and analysis, Hidden Markov model, MOOCs Learner's state, Learning behavior, Status transition.

1. Introduction

In recent years, MOOCs have developed rapidly based on information technology and mobile learning. As a new form of educational innovation, the rise of MOOCs can be traced back to 2008. Since then some MOOCs platforms have emerged such as Coursera (Li et al, 2017), edX, Udacity in US, and xuetangx, cnmooc, icourse163 in China. Although MOOCs enrich the interaction of traditional online learning and contributes to resource sharing, it still faces some problems as follows. (1) Online learning relies on

the self-participation of learners, so the course withdrawal rate is high and the resource utilization of the platform is very low. (2) The evaluation system of learning outcomes is imperfect, and the relationship between the environmental indicators and learning behavior has not been established, so the learners with poor expression cannot help them adjust the current learning mode in time. (3) As the lack of effective data analysis, it is difficult for learners to grasp the learning state. In the past, most of the relevant research explained the relationship between environment and learning behavior from the perspective of behavioral theory or education, and lacked quantitative analysis.

By summarizing the previous literatures, it is found that the learning state of platform learners in the real world is not only related to the psychological factors and life factors of the learners themselves, but also related to the overall learning atmosphere of the MOOCs platform. Because learners' participation in distance education is mainly based on autonomous learning. So, learners' awareness of the platform environment will affect their attitude towards curriculum learning. In order to study the above problems, the Hidden Markov Model was applied to our study, which was used to quantitatively explain the relationship between environmental factors, learning behaviors and learner's individual conditions.

This paper is organized as follows. In section 2, related research of learner behavior and Hidden Markov Model are introduced. In section 3, the modeling of learning state transition is shown. In section 4, verification of model is illustrated. Section 5 is the discussion and conclusions of this article.

2. Related Works

Learning behaviors includes learning motivations and all behaviors in the realization of learning objectives, and it can be regarded as the result of interaction between learners and learning environment. From previous studies, scholars mainly focus on learning behavior and motivation.

2.1. Learner Behavior

The related research on learning behavior is mainly summarized into three aspects: the influencing factors of online learning behavior, the classification of learning behavior characteristics and the emotions of learners. Venkatesh (2000) proposed the TAM3 model to analyze the usefulness and ease of use of things from the learners, and explored the differences in individual characteristics such as age and education, the differences in learning environment, the impact of social group differences and the convenience of the course on learning behavior [16]. Shen (2014) studied the relationship between the expected rate of return of courses and the level of knowledge, distinguished the expected rate of return of students with different levels of knowledge, and analyzed the impact of building learning networks on knowledge acquisition [14]. Liu (2016) studied behavioral differences in the MOOCs environment from the aspects of learner type, gender, academic level and age [10]. Dixson (2011) studied the impact of activity organization and teaching resources on learning outcomes in online learning [3]. Li (2005) pointed out that the important factors of affecting learning outcomes include learning intention, expected learning experience and learning tendency [9].

In the study of teaching based on learning behavior, learners are classified according to learning behaviors, and teaching suggestions and measures are proposed for each category. Wang (2014) analyzed the five types of MOOCs learners, namely, non-participants, passive participants, temporary participants, passive mandatory participants, active participants, and discussed the learning effect of students on the MOOCs platform [17]. Yao (2009) defined learning behavior as negative and positive, which including the four dimensions of attention, motivation, learning attitude and learning strategy [22]. In addition, in the study of learner emotions, some scholars proposed learner's emotion recognition model based on cognitive evaluation in E-learning system and OCC model, which used Mamdani fuzzy inference model to realize students' expectation of learning events, and simulated the model by constructing dynamic Bayesian network (Qiao, 2010; Wang, 2011) [13][18].

2.2. Hidden Markov Model

Hidden Markov model can be used to describe the process of randomly generating observation sequences of hidden Markov chains, which was originally applied in the field of ecology [1]. Since then many research works have focused on the application of Hidden Markov Models in signal recognition such as speech recognition and motion recognition. In the study of behavior classification, Honsel (2016) selected the software development exchange platform and scored the number of applications submitted by developers, bug fixes, and responses to related questions, and classified hidden forms by comprehensive scores [6]. In the study of motion recognition, Yamato (1992) identified the tennis action, and then transformed the picture into a sequence of features, and converted it into an observation sequence [21]. He (2016) selected netizens' characteristics, information subjects and information content completion degree to establish three-dimensional indicators, and divided four implicit states according to the changes of attention, and then established Hidden Markov Model according to the event evolution process of micro blog public opinion [5]. Pu (2014) used the Hidden Markov Model to dynamically predict the user's interest changes, and defined the interest transfer state according to the user's change of topic interest [12].

In addition, HMM has been widely applied in many fields. For example, the HMM of two states is established for the financial daily income sequence, the hidden Markov state can be interpreted as the state of the financial market, the high fluctuation state and the low state correspond respectively to the period of shock and stability [7]. In CRM, the hidden status can be used to reflect the purchasing tendency of customers and to evaluate the potential value that customers may bring [4]. Because the learning state sequence used in this study is a state time series that cannot be directly visible, this recessive state will enable learners to present different learning performances, which are reflected in the degree of course completion and activity participation. Compared with other models, such as deep learning model, the research of hidden Markov model is relatively mature and the theory is more complete. Taking into account the data quality, data volume, application scenarios and iterative efficiency studied in this paper, this study finally chooses a more suitable HMM. In the learning behavior, the learning state will be affected by subjective psychology, physiological conditions and the overall environment of the platform, and then transferred. Because it is difficult to directly evaluate the psychological and physiological conditions of learners, this paper discusses the transfer of user learning status from

the perspective of objective conditions. Since these states have hidden features and need to be reflected by the explicit behavior of learning, the learning state can be studied as the hidden state in the HMM.

3. Modeling of Learning State Transition

This article chooses the platform of www.shlll.net for study. The curriculum of this platform includes different categories such as life and health care, literature and art, higher education, vocational education and so on. Users of the platform include students, professional workers and migrant workers.

3.1. Definition of Learning State Transition

According to the application conditions of HMM, the observation behavior is determined by the hidden state, and the hidden state of the current period is affected by the hidden state of the previous period. Therefore, the following hypothesis is made in this paper. It is assumed that the learner's course learning behavior is determined only by the learning state, that is, it will exhibit certain regular behavior in the same state. And the learning state shifts to the first-order Markov process. This paper studies the influence of objective learning environment on learning state, assuming that the learner's life and work will not affect the learning of the online course, and the learner's own behavior in the current period will not affect the next learning state. The relevant symbols indicating the change of the learning state in the time series are shown in Table1.

Table 1. HMM parameter symbol description

Variable	Description
S=(s_1,s_2,...,s_T)	The learning state of the learner at each stage.
O=(o_1,o_2,...,o_T)	The observation sequence of learning behavior presented by learners in T-period.
A=[a_{ij}]$_{N*N}$	The probability that the learner is in the learning state i in the current period and in the learning state j in the next period.
B=[b_{io}]$_{N*K}$	The probability of a certain learning behavior o in the learning state i (the learning behavior contains K specific behavioral characteristics, k=2 in this paper).
π=(π_s)	t=1, the probability that the learner is in the learning states s at the beginning of the time series.

The explanation of the relationship between learning state transition and observable learning behavior in different periods is shown as Fig. 1. In each period t, the learner will be in a state of N states with a certain probability. The learning state transition probability a_{ij} indicates that the learner is currently in a certain state, the likelihood and tendency to move to another learning state or remain in the current state in the next period. b_{io} represents the probability of exhibiting observed behavior O in state i, and the observed behavior consists of two indicators: course completion scores and activity. In this paper,

we studied the influence of objective environment on learner state transition probability a_{ij}, including internal course quality, the number of students in the internal course and the learning situation of internal curriculum. In addition, the academic level will have an impact on the learning state at t=1.

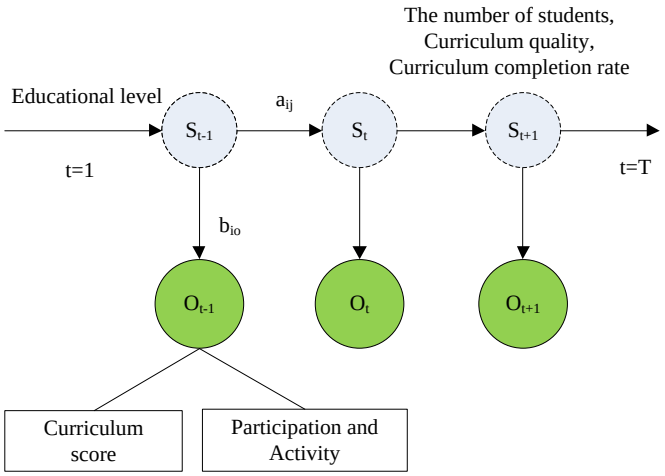


Fig. 1. Learning state transition and observable learning behavior in different periods

According to the three basic elements in the HMM, that is observation sequence, hidden state and initial state. Three basic parameters of the model is defined $\lambda = \{A, B, \pi\}$, which is defined respectively as the learning state transition probability, the learning behavior observation probability and the initial learning state probability. After that, some variables were selected according to HMM model, and it is shown in Table2.

3.2. Learning Behavior Observation Sequence Description

Two indexes of learning behavior observation sequence were selected, one is the completion score of the internal course, and the other is the activity of participating in the activity.

Average Score of Internal Courses The learner's course score is calculated based on the course completion rate and the learning interval. Referring to the calculation method of the credit rating [11], the course score was defined by a linear transformation according to formula (1) within a certain interval without changing the learner's completion status. Based on the test of the model results, the learner's score interval is defined as (1) and (2).

$$Score = 1 + 1 * Completionrate \tag{1}$$

When the learner stops learning, learner's knowledge will gradually decline. According to the Ebbinghaus memory curve [23], the knowledge memory acquired by the learner

Table 2. HMM model selection variable interpretation

Heading level	Variables	Description
Observation sequence	Score$_{mt}$	The average course completion score of learner m in t period.
	Active$_{mt}$	The activity of the learner m participating in the event during the t period.
Variable that affects the probability of state transition	Otherscore$_t$(X$_1$)	Average course completion scores of other learners during the t period.
	TotalNum$_t$(X$_2$)	The total number of learners in the t-time on the MOOCs platform.
	CourseLevel$_t$(X$_3$)	The students' average score during the t period.
Variable that affects the initial state	Edu$_m$(X$_4$)	Education information filled out by learner m by learner m.

changes in the form of a negative index. If the learner's knowledge is reduced, his or her course score will also decrease. For example, the course scores of learner m in t period are calculated as follows, where parameter $Score_{mt'}$ refers to the recent course score of learner m, parameter $Interval_{mt}$ refers to the interval at which the learner m learns the course from the t period.

$$Score_{mt} = Score_{mt'} exp(-KInterval_{mt}) \tag{2}$$

The average score of internal courses is a continuous variable. The course scores in a certain state obey the normal distribution. For each learning state, the course scores have different mean and variance. The probability of learners get course scores x in the i state is expressed as follows.

$$b_i^1(x) = \frac{1}{(2\pi)^{\frac{1}{2}}\sigma_i} e^{-\frac{1}{2\sigma^2}(x-u_i)^2}, i = 1, 2, ..., N \tag{3}$$

Participation in Activities Another measure of activity is the statistics of the number of learners participating in activities. For example, statistics on learners' participation in various learning activities over a period of time. If the number of learners participating in activities is 0, and the value in discrete expression is 1, the value of other activities were equalization and discretization. The sample of value of activity is shown in Table3.

Table 3. Discrete of activity value

Number of activities	Value
0	1
1	2
More than 6 times	3

For the activity observation sequence, the observation probability matrix is directly used for initialization and iterative estimation. The matrix form is show as follows, where b_{ia}^2 indicates the probability that the i state appears as the activity of a, and i=1,2,...,n, a= 1,2,3.

$$\begin{bmatrix} b_{11}^2 & b_{12}^2 & b_{12}^2 \\ b_{21}^2 & b_{22}^2 & b_{23}^2 \\ \vdots & \vdots & \vdots \\ b_{n1}^2 & b_{n2}^2 & b_{n3}^2 \end{bmatrix} \tag{4}$$

4. Verification of model

4.1. Data collection

The research data of this paper is collected and analysed by the MOOCs platform of www.shlll.net. The analysis shows that the quality of the course in the MOOCs platform, the online interaction of the learners, and the various interest groups on the online will influence the learner's recognition of the online course environment, and affect the learning status.

The learners who choose to study the internal curriculum are the research objects, and when analyzing the influence of the objective environment on the state transition, the average scores of other learners, the course scores and the number of learners are standardized to compare the influence of these three variables on the state transition. The sample of model is shown in Table 4.

Table 4. Sample of Model

Time Unit	Month
Selection condition	Learning duration is more than one year
Start and end time	January 2013 to December 2016
Number of students	62
Number of samples	1367

The learner behavior record data is collected from 2011 to 2016, and it has a total of 2,253,367 registered members. it involves 4,430 courses and 308 learning activities. As the number of courses has increased since 2013, the quantity of students has started to increase, so this paper count user behavior data from 2013. In the sample data, statistics were conducted once a month, and 62 course users with a learning duration of more than one year and including academic information were selected to track their behavior for two years. Since the end of 2016 the learner who study less than two years was excluded, the actual number of samples was 1367. And the descriptive statistics of variables is shown in Table 5.

Table 5. Sample of Model

Name	Number	Minimum	Maximum	Average	Standard deviation
Score	1367	0.13	2	0.99	0.41
Active	1367	1	3	1.23	0.54
Edu	1367	1	7	4.82	1.78
OtherScore	1367	0.89	2	1.46	0.15
CourseLevel	1367	3.87	5.00	4.36	0.20
TotalNum	1367	3	899	234.57	220.32

4.2. Model Parameter Estimation

The model parameters are estimated by EM algorithm (Zhang, 2014), and the hidden state as 2, 3, 4 was selected respectively, and the maximum likelihood and BIC values estimated by the model are as follows: (-1027.873, 21875.13), (-693.0693, 1674.954), (-654.5903, 1514.607). According to the BIC (Bayesian Information Criterion), the effect of the four states is not obvious. So three hidden states are selected for analysis in our study, and it was defined as negative, normal and positive states, and they were represented by numbers 1, 2, and 3 respectively. And then behavior theory was used to analyze the initial state probability parameters, observation probability parameters and state transition probability parameters in the model.

Learning Characteristics in Different States According to the model, the learner status classification result and the course score distribution can be obtained. The mean, standard deviation, and activity probability of the positive distribution of the course scores of learners with different learning states are shown in Table 6. The course scores in different states are shown in Fig. 2.

Table 6. Sample of Model

Name	Average score	Standard deviation score	Activity level (Active=1)	Activity level (Active=2)	Activity level (Active=3)
Negative	0.514	0.155	0.931	0.062	0.007
Normal	0.937	0.133	0.931	0.055	0.014
Positive	1.401	0.246	0.646	0.235	0.120

According to the classification results of Table 6 and the distribution of the course scores in Fig. 2, it can be seen that the learners in the positive state have the highest average score among the learners in the three states, and the internal standard deviation of these learners is relatively large. With the learner's activity for learning decreases, the course score will decrease. In addition, the learner's activity (activity level = 2) in the positive state is 0.235, which is higher than that in the normal and negative state. The learner's activity (activity level = 3) is also significantly higher than that in the stable and negative state, which indicates that learners in the positive state are more likely to show active state.

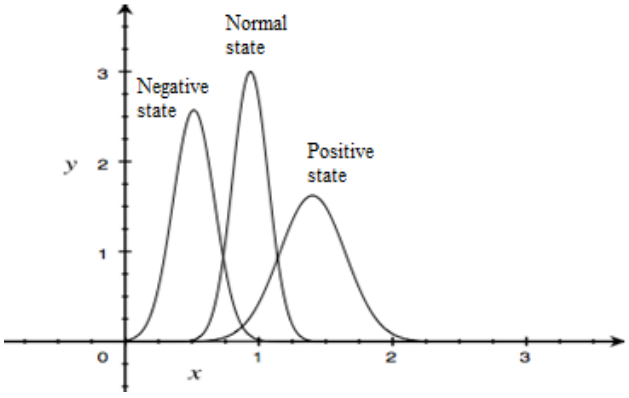


Fig. 2. Course scores in different states

Previous studies have shown that learners who participate in various activities on the distance learning platform are more likely to get higher scores in course learning [15], and learners with stronger autonomy will not be limited to course of video learning, and they learn in more diverse ways. Learners who are in a negative state do not perform well in both course completion scores and activity participation, and there is a risk of exiting the learning platform. Combining the behavioral responses in different states, the three learning states are defined as follows.

(1) The learners in the negative state are slack in both course learning and activity participation, and they belong to the bystanders and are subject to the risk of losing users.

(2) Learners have a good attitude towards learning at a normal state, he or she will take the initiative to study some courses and participate in the test or related activities in the course. However, there is no clear goal for completing the course, and he or she will learn with a relatively relaxed attitude.

(3) Positive state. The learners aim to complete the course study and will fully participate in the study of the class. The frequency of the course is high and they will actively participate in the activities on the platform.

According to the Viterbi algorithm, the learning state of learners in each period is predicted, which is shown as in Fig. 3.

It can be seen from Fig. 3, the number of active learners in 2014 is large, especially between July 2014 and January 2015 (the green area in Fig. 3), most of them are in a better learning state, and there are no negative learners (the red area in Fig. 3). After 2015, the proportion of learners who are in a negative state has increased. After July 2016, the proportion of negative learners has decreased. The number of learners who are in a positive and stable state has increased, and the learning attitude of learners on the platform has improved.

Initial Learning State By setting the educational level as a parameter in the model, the probability that the initial learning state of these learners will be negative, stable and positive can be obtained. The results are shown in Table 7.

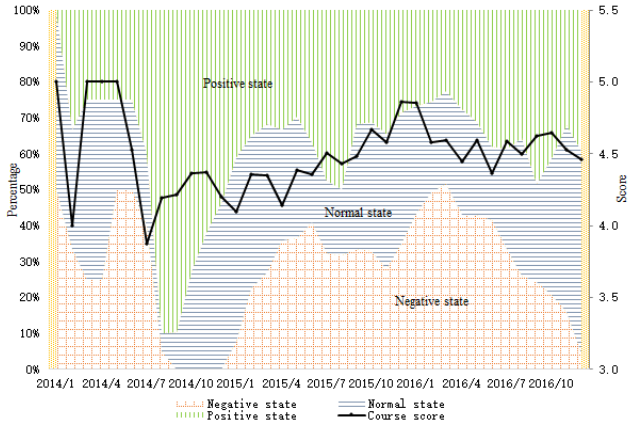


Fig. 3. The proportion of learners in different learning states in each period

Table 7. Initial state probability of different educational level

Educational level	Junior middle school	Senior middle school	Vocational college	Junior college	Bachelor	Postgraduate
Negative	0.0422	0.0238	0.0080	0.0020	0.0005	0.0001
Normal	0.2289	0.5759	0.8613	0.9660	0.9924	0.9983
Positive	0.7289	0.4003	0.1307	0.0320	0.0072	0.0016

It can be seen from Table 7 that learners of different educational levels have different probabilities of learning state at the beginning of their study. Junior middle school learners have a probability of 0.7289 and they have a very positive attitude towards learning. With the improvement of learners' educational level, the probability value of learners' negative state decreases gradually, and the probability value of learners' normal state increases gradually. Especially, the normal state probability value of learners with bachelor or above is close to 1. Because of the popularity of general courses of culture and custom on this platform, the basic knowledge requirement for learning these courses is not high. Even the learners with lower educational background have strong self-confidence to complete their studies. Some learners with higher educational background often choose courses with stronger professional knowledge for the needs of their work. Relatively speaking, on this platform, the main learners are still learners who learn from the knowledge required for their work. In addition, according to the theory of habitual domain, the learners with higher knowledge level are in a relatively stable habitual domain, the expected benefits of curriculum learning are lower, the entry and absorption of new information will be more hindered, and their enthusiasm for learning will be weaker than those with lower education level, but the possibility of negative attitude is very low.

4.3. Learning State Transfer

According to the HMM parameter estimation results, the relationship between the probability of transition to different states in the next period and the environmental factors

of different MOOCs can be getted when the current period is in a negative, normal and positive state. And the negative state is taken as an example in the current period, and the negative state is maintained in the next period. The probability distribution function of S_{t+1}=1, transition to the normal state S_{t+1}=2 and transition to the active state S_{t+1}=3 is shown as follows.

$$P\left(S_{t+1}=1|S_{\mathrm{t}}=1\right)=\frac{1}{1+\exp\left(I_1\right)+\exp\left(I_2\right)} \tag{5}$$

$$P\left(S_{t+1}=2|S_t=1\right)=\frac{\exp\left(I_1\right)}{1+\exp\left(I_1\right)+\exp\left(I_2\right)} \tag{6}$$

$$P\left(S_{t+1}=3|S_t=1\right)=\frac{\exp\left(I_2\right)}{1+\exp\left(I_1\right)+\exp\left(I_2\right)} \tag{7}$$

where, I_1=-16.76+6.16X_1+1.44X_2+2.21X_3, X_2=-1.98+0.49X_1+0.67X_2+0.01X_3. When there is no activity on the learning platform, that is, when the variables are 0, the probability that the next learner stays in the negative state is close to 1. Without the external influence, the learner's state will not directly change from the passive state to the positive state. Through derivation analysis of independent variables, the influence of each variable on state transition is analyzed. The probability of learners remaining in a negative state is negative for each variable, and the derivative of course scores is almost 0. The most obvious effect of other learners' course scores on state transition is that of course scores. And then we mainly analyze the influence of other learners'course scores on state transition, and fix the value of other variables to 0. The probability that learners move from negative state to other states varies with other learner course scores is shown in Fig. 4.

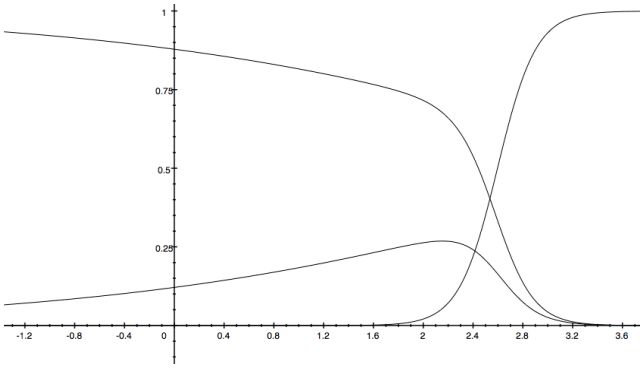


Fig. 4. The influence of other learner course scores on transition probability

The completion of other learners reflects the overall learning level of the platform. In order to better reflect the changing trend, the variables were standardized. The independent variables ranged from 0.89 to 2, corresponding to -1.2 to 3.6 in Fig. 4. When the

learner is in a negative state, the probability of keeping the state unchanged is very high. If the other learners on the platform complete the course better, the possibility of the learner in a negative state will be reduced, and the probability of turning to a positive state or a better state will be increased. When the scores of other learners are still at a low level, the improvement of learners'state in a negative state will be relatively small, but with the improvement of the overall learning state, the learner's learning state will gradually improve significantly, and finally the learner will transfer to a stable state with a probability of close to 1. In general, the overall learning status on the platform will have a significant effect on learners. That is to say, the heterogeneity in the learning environment promotes the learning attitude. When the learning effect of other people exceeds a certain value, the heterogeneity of the learner in the negative state is more obvious, and the learner in the negative state will be more actively close to excellent learners, the probability of learners improving their learning status will be significantly improved.

According to ARCS (2010) theory [8] and emotion regulation theory, good completion of other learners will enable learners to increase their confidence in curriculum learning, and increase their self-efficacy through mutual attention, and effectively improve negative learning emotions [2]. However, when other learners score far better than themselves, learners' state improvement will have certain limitations. At this time, learners with negative state have less similarities with learners with better curriculum completion, and it is not easy for them to establish a closer relationship. Therefore, learners with similar positive knowledge level should be organized to motivate negative learners to learn, so that negative learners can easily accept help, and change their state by communicating with each other.

It can also be seen from Fig. 4 that with the improvement of objective environment level, learners in a negative state are more likely to increase their interest in learning, and The course scores and the total number of learners have a more stable positive impact on the learners' transition from negative to positive. When other learners' scores are less than 1.76, the influence of course scores on improving the negative state will be weaker than that of the total number of learners. On the contrary, if other learners' scores are higher than 1.76, their influence will be significantly improved. When the score reaches 1.8, the probability of turning to the stable state is the highest. It also shows that the learner's learning state is a process of gradual improvement, and the probability of direct improvement is relatively small. By improving the environment, it can significantly encourage learners to improve their learning status.

The current period is in a normal state, and the probability of moving to a negative state, maintaining a normal state, or shifting to a positive state in the next period is show as follows:

$$P\left(S_{t+1}=1|S_t=2\right)=\frac{1}{1+\exp\left(J_1\right)+\exp\left(J_2\right)} \tag{8}$$

$$P\left(S_{t+1}=2|S_t=2\right)=\frac{1+\exp\left(J_1\right)}{1+\exp\left(J_1\right)+\exp\left(J_2\right)} \tag{9}$$

$$P\left(S_{t+1}=3|S_t=2\right)=\frac{\exp\left(J_2\right)}{1+\exp\left(J_1\right)+\exp\left(J_2\right)} \tag{10}$$

where, J_1=1.65+0.02X_1+0.33X_2-0.05X_3, J_2=-1.49+0.46X_1+0.46X_2+0.67X_3, and the current period is in a positive state, and the probability of moving to a negative state or a normal state, or keeping in a positive state in the next period is show as follows:

$$P\left(S_{t+1}=1|S_t=3\right)=\frac{1}{1+\exp\left(Q_1\right)+\exp\left(Q_2\right)} \tag{11}$$

$$P\left(S_{t+1}=2|S_t=3\right)=\frac{\exp\left(Q_1\right)}{1+\exp\left(Q_1\right)+\exp\left(Q_2\right)} \tag{12}$$

$$P\left(S_{t+1}=3|S_t=3\right)=\frac{\exp\left(Q_2\right)}{1+\exp\left(Q_1\right)+\exp\left(Q_2\right)} \tag{13}$$

where, Q_1=12.30+0.16X_1+0.22X_2-0.28X_3, Q_2=13.77+0.53X_1+0.12X_2-0.39X_3. Similarly, for the situation of normal state and positive state, the probabilistic transfer function expression is used for derivative analysis. The above three cases are summarized as shown in Table 8. The three coincidences in parentheses indicate the positive or negative effects of three learning environment factors on state transition, the order of the three variables is course scores of other learner, the total number of student of courses, the average score of the courses. The * in the upper right corner represents the variable that has the greatest impact on the probability transition, and the + or - sign indicates a positive or negative impact on the state transition probability.

Table 8. The impact of each variable in the state transition

t t+1	Negative state	Normal state	Positive state
Negative state	(-* - -)	(+* + +)	(+ +* +)
Normal state	(- - -*)	(- + -*)	(+ + +*)
Positive state	(- - -)	(- +* +)	(+* - -)
(Course scores of other learner, Total number, Average scores)			

When the learner is in a negative state, the scores of other learners' learning scores have a significant effect on the improvement of the learning state. At this time, the learning state of other student has a greater influence on the learner in a negative state, which is easy for the learner to enhance their confidence and adopt a positive solution to turn pressure into motivation. It is explained that the MOOCs learning platform can effectively promote the learner in a negative state to improve the learning state and increase the investment in learning by improving the overall learning environment.

When the learner is in a normal learning state, the MOOCs environment has the least impact on the change of the learning state. At this time, the overall number of students in the course and the improvement of the course quality are more conducive to the learner to turn to a more positive state, and the completion of other learner courses has less influence on learners who are in a stable state. According to Moore's theory [19], Curriculum design is one of the factors influencing distance interaction in distance education. A good course can effectively shorten the interaction distance between learners and teaching resources, and improve learners' interest in the curriculum, and increase interaction with

teaching resources. In addition, Moore's theory provides that different stages of learning form their own views through interactive exchanges, and in the third stage, other people's understanding of the knowledge is incorporated into their own knowledge system, and in the fourth stage, information is shared, so as to gradually realize the steady dissemination of knowledge [20]. Therefore, when the learner is in a normal state of active learning, the increase of student will promote the learner in the course discussion and learning activities, which will increase the possibility of the learner to improve the learning state and effectively prevent the learner's state sliding down.

When the learner is already in a positive learning state, the scores of other learners' completion of the course has a significant impact on the learner's positive state. According to the theory of confidence and satisfaction in ARCS, when the surrounding students complete well, it will encourage learners to increase the confidence of the completion of the course and the satisfaction of knowledge acquisition. At the same time, the learners in a positive state will be more actively involved in learning because of the competitive mentality, which will help the learners to maintain the current state of active learning.

5. Discussion and Conclusions

Hidden Markov Model was used as the theoretical framework in our study, and the hidden state refers to the learner's learning state. The observation sequence is the learner's course completion score and the activity participation. According to our study, the learner's knowledge level will have an impact on the initial learning state, and the learner's learning status will be affected by the environmental factors of the MOOCs. Therefore, the learners' basic knowledge level, learning behavior and the learning environment of the course-mourning platform are correlated and modeled. In addition, the behavioral theory is applied to explain the influence of the learning behavior characteristics under different learning conditions, the total number of course learning, the learner's course completion score, and the course score on the learner's learning state transition.

From the results of the research, the overall activity participation on the MOOCs platform is low, but the active learners are more likely to actively participate in the activity and the course completion score is higher. When learners are in different states, the influence of objective environment on learners'motivation is different, and the motivation factors are also different. When the learner is in a negative state, the change of the objective environment will have a significant impact on the improvement of the learner's state; when the learner is in a positive state, the influence of the learning environment on the learning state is not very great, and the completion of other learners' courses can help learners to stay in a positive state, but the course score and the total number of students may cause the learner's state to decline slightly; when the learner is in a normal state, the MOOCs environment has the least impact on the learning state, and the course score is the most significant effective in promoting the learner's state.

In general, according to the research results of the learner's state, the improvement of MOOCs teaching can be carried out from the following aspects.

(1) Course design optimization. According to the research results, when the learner is in a normal learning state, the course score has a significant impact on it. Learners who have a medium-level course score can be selected from the platform, and combined with their browsing records, relevant courses with a recommended score of more than

4.8 can be recommended. In addition, according to the statistical results, the number of students studying in-house courses increased from July to September. During this period, courses suitable for students can be added, such as knowledge development with school associations, Graduate Development Planning and other courses.

(2) Encourage collaborative learning. According to the research results, when the learner is in a negative learning state, the increase in the number of learners can help the learner to improve his or her state significantly. Therefore, arranging positive and negative learners to study together can help negative learners increase their interest in learning, and help those negative learners improve their learning effect.

(3) Standardized management of learner behavior. For example, we can take measures such as the sign-in system or improve the learner evaluation mechanism to supervise the learner's learning activities. Through regular testing of learners, we not only record their course completion and participation, but also track their learning status in order to promptly remind learners with poor learning and take measures to stimulate their active learning.

From the perspective of the development of MOOCs, personalized teaching based on learner's status will become the trend of MOOCs teaching. In future research, artificial intelligence, big data analysis and other technical methods can be used to analyze learning behavior data and learner status, so as to provide help for personalized teaching.

Acknowledgements. This work was supported in part by Shanghai Higher Education Society under Grant (No. GJEL1851), Shanghai Philosophy and Social Sciences Plan (No.2016BGL004, No.2018BGL023), Shanghai Open Distance Education Engineering Technology Research Center (KFKT1701). Yonghui Dai is the corresponding author. Many thanks to Dr. Guowei Li for his assistance to Haijian Chen and Yonghui Dai who are the joint first authors of this paper.

References

1. Baum, L.E., Eagon, J.A.: An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. Bulletin of the American Mathematical Society 73(3), 360–363 (1967)
2. Chen, H.J., Dai, Y.H., Feng, Y.J., Jiang, B., Xiao, J., You, B.: Construction of affective education in mobile learning: The study based on learner's interest and emotion recognition. Computer Science and Information Systems 14(3), 685–702 (2016)
3. Dixson, M.D.: Engaging the online learner: Activities and resources for creative instruction. Journal of Scholarship of Teaching & Learning 11(3), 120–121 (2012)
4. Guo, L.W. (ed.): Research on Dynamic Customer Relationship Management Based on Hidden Markov Model. Harbin Industrial University (2013)
5. He, J.M., Li, X.: A hidden markov model research in the microblog public opinion evolutionary analysis. Information Science 34(4), 7–12 (2016)
6. Honsel, V., Herbold, S., Grabowski, J.: Hidden markov models for the prediction of developer involvement dynamics and workload. In: International Conference on Predictive MODELS and Data Analytics in Software Engineering. pp. 1–10 (2016)
7. Hu, S.L. (ed.): Hidden Markov Model and Its Application in Finance. Huazhong Normal University Press (2014)
8. Keller, D.J.M.: The arcs model of motivational design. Motivational Design for Learning & Performance (10), 43–74 (2010)

9. Li, X.P., Guo, J.W.: Research on the correlation between learning attitude and learning behavior. Psychological and Behavioral Research 3(4), 265–267 (2005)
10. Liu, S.N.Y., Liu, Z., Gao, J., Sun, J.W.: Analysis of the differences of learners‘ learning behavior in the environment of mooc. Research on Electro-Education (10), 57–63, 69 (2016)
11. Pang, S.L., He, Y.Z., Wang, S.Y., Jiang, H.: Multilayer crossing credit scoring models for enterprise and application based on risk environment. Journal of management sciences in China 20(10), 57–69 (2017)
12. Pu, J.C. (ed.): Microblog user interest modeling and its dynamic research. Harbin Industrial University (2014)
13. Qiao, X.J., Wang, Z.L., Wang, W.S.: Emotional modeling of e-learning system based on occ model. Computer Science (05), 214–218 (2010)
14. Shen, C., He, X.: Research on the influence of college students' group learning behavior on learning achievements. Journal of Nanjing University of Posts and Telecommunications (Social Sciences Edition) (01), 106–112 (2014)
15. Sivapalan, S., Cregan, P.: Value of online resources for learning by distance education. International Journal of Innovation in Science and Mathematics Education (formerly CAL-laborate International) 24(3), 414–429 (2012)
16. Venkatesh, V.: Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. Information Systems Research 11(4), 342–365 (2000)
17. Wang, J.D., Feng, Y.Y., Wang, W.: Cold thinking behind mooc heat. Educational Research (09), 104–111 (2014)
18. Wang, W.S., Gong, W.: Research on emotional cognition personalized student model in e-learning. Application Research of Computers (11), 4174–4176, 4183 (2011)
19. Wang, Y.S.: An empirical study of college students‘ learning input-based on data analysis of 2012 national college students‘ learning survey. China Higher Education Research (01), 32–36 (2013)
20. Wilson, J.M., Goodman, P.S., Cronin, M.A.: Group learning. Academy of Management Review 32(4), 1041–1059 (2007)
21. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. Transactions of the Institute of Electronics Information & Communication Engineers 76(9), 379–385 (1993)
22. Yao, C.Z., Mi, J.R., Wang, H.C.: A review of research on "learning behavior" at home and abroad. Teaching and Management (10), 48–50 (2009)
23. Yu, H.T., Cui, R.F., D, Q.Q.: Microblog user interest model based on forgetting curve, computer engineering and design. Computer Engineering and Design 35(10), 3367–3372, 3379 (2014)

Haijian Chen is an associate professor at the Shanghai Engineering Research Center of Open Distance Education, Shanghai Open University, China. He received his Ph.D. in Management Science and Engineering from Shanghai University of Finance and Economics, China in 2015. His current research interests include Educational technology and cloud computing. Contact him at xochj@shtvu.edu.cn.

Yonghui Dai the corresponding author of this paper. He is currently a lecturer at the Management School, Shanghai University of International Business and Economics, China. He received his Ph.D. in Management Science and Engineering from Shanghai University of Finance and Economics, China in 2016. His current research interests include Affective Computing, Intelligence Service and Big Data Analytics. His works have appeared in international journals more than twenty papers. Contact him at daiyonghui@suibe.edu.cn.

Heyu Gao received her Master at the School of Information Management and Engineering, Shanghai University of Finance and Economics, China in 2016. Her research interests include Data mining. Contact her at gaoheyu1993@163.com.

Dongmei Han is a full Professor and Doctoral Advisor at the School of Information Management and Engineering, Shanghai University of Finance and Economics, China. She received her Ph.D. in 2002 from Quantitative Economics in Jilin University, China. Currently, her research interests include finance engineering, management science and Engineering. Contact her at dongmeihan@mail.shufe.edu.cn.

Shan Li received her Master at the Management School, Shanghai University of International Business and Economics, China in 2019. Her current research interests include Network behavior analysis. Contact her at 380734913@qq.com.

Received: October 2, 2018; Accepted: June 15, 2019.

Computer Science and Information Systems 16(3):867–890 https://doi.org/10.2298/CSIS180917031S

MR-DFM: A Multi-path Routing Algorithm Based on Data Fusion Mechanism in Sensor Networks

Zeyu Sun[1,2], Zhiguo Lv[1,3], Yue Hou[1], Chen Xu[4], and Ben Yan[1]

[1] School of Computer Science and Engineering, Luoyang Institute of Science and Technology, 471023 Luoyang, China
{lylgszy, lzg96wl, houyue0315}@163.com; yanben@lit.edu.cn
[2] Department of Computer Science and Technology, Xi'an Jiaotong University, 710049 Xi'an, China
lylgszy@163.com
[3] State Key Laboratory of Integrated Services Networks, Xidian University, 710071 Xi'an, China
lzg96wl@163.com
[4] School of Computer Science and Information Engineering, Shanghai Institute of Technology 201418, Shanghai, China
sitxuchen@163.com

Abstract. Sensor networks will always suffer from load imbalance, which causes bottlenecks to the communication links. In order to address this problem, a multi-path routing algorithm based on data-fusion-mechanism (MR-DFM) is proposed in this work. In this algorithm, the Mobile Sink controls the clustered energy consumption of the nodes in the event domain according to the delay messages relayed by the neighbor nodes. Meanwhile, the optimal neighbor node in the candidate set is obtained according to the data stream of the neighbor nodes to perform the relay of the data packets. It is shown via simulation results that the proposed MR-DFM algorithm shows obvious improvement according to the energy consumption of the network throughput of the sensing data and the throughput of the sensing data and each hop of the neighbor nodes. Therefore, it is verified that the proposed MR-DFM algorithm shows remarkable data fusion effects and optimizes the network resources.

Keywords: sensor network, data fusion mechanism, multi-path routing, energy consumption, network lifetime.

1. Introduction

The data transmission causes a major part of the network energy consumption in wireless sensor networks (WSNs) [1-3]. It depends on the quantity of the transmitted data in the network for the level of the data transmission energy consumption. The data fusion in the network based on the data compression technique could efficiently reduce the amount of the data. In addition the node clustering technique offers structural convenience for the data fusion. By effectively combining clustering

optimization and the data fusion mechanism, the new energy-saving routing technologies have attracted much interest.

The clustering routing protocols divide the WSN into several clusters according to the node energy and the node distance. The nodes within each cluster communicate with the aggregation node through the cluster head. Considering the sharing property of the WSN links, the sensing routing protocol was proposed. In the sensing routing protocol, the transmission strategy of the data packets is controlled by estimating some parameters of the links, such as the signal-to-noise ratio (SNR), delay and load [4-5]. In addition, these parameters are adjusted adaptively according to the network performance. The selecting strategy of the cluster head is modified in paper [6] based on the clustering routing protocols and the remaining energy of the nodes is considered to balance the energy between different nodes and prolong the network lifetime. In paper [7-8], with the rapid development and application of the locating technologies, the geological location based routing protocol has attracted much attention due to its excellent extensibility and adaptability to wireless networks. Based on routing protocol, the core of the geological location is to simply employ the partial topology information which is among different nodes to relay the data in a greedy method to the nearest neighbor node of the destination node. When the data packet reaches a routing void, the flooding, the back-off or the marginal recovery mechanism can be employed to continue the transmission of the data packets. As for the greedy relay mechanism, the expense of any recovery scheme is excessively large. It is an efficient method to saving energy by moving the Sink to a proper site to reduce the data transmission distance. By doing so, the one-hop neighbors of the Sink can be changed so that the hotspot problem can be solved, which no longer becomes the bottleneck of the network performance. The Mobile Sink (MS) [9] can visit every node or only several locations in the network. From the perspective of data routing, the data gathering through the movement of the Sink can be considered as the interception of the data in the routing, i.e., the MS intercepts the transmitted data through some links by proper movement to reduce the network load.

The rest of paper is organized as follows. In Section 2 related work is presented. The data fusion mechanism is given through the network model in Section 3. The multi-route control algorithm is given by the control strategy of the node in Section 4. In Section 5, MR-DFM is assessed by experimental results. Finally, this paper is concluded in Section 6.

2. Related Work

A tree-shaped data aggregation routing algorithm organizes related nodes as a routing tree with the Sink as its root. Each node on the tree has one and only one parent node and multiple child nodes. During the data gathering process, each non-leaf node on the tree gathers the data relayed from its child nodes and aggregates the relayed data with its own sensing data. Then the aggregated data are relayed to its parent node. This aggregation process is performed along with the data transmission until the data finally reaches the Sink. Sun et. al [10] first defined the link cost based on node distance, the

information entropy of the nodes, the union entropy and the data amount and then obtained High-dimensional Data Aggregation Control (HDAC) in the network through the dynamic scheduling. Nie et.al [11] proposed three different algorithms, i.e., MST-based, DMDC-based and COM algorithms to construct data aggregation trees with better energy consumption performances and delay performances with different data growth rates. The location of the one-hop neighbors or the distance information is employed by the L-PEDAPs [12] to construct the local minimum spanning tree and correlated neighbor graph. Then three different selection methods, i.e. FP, MH and SWP, are adopted for the choosing parent nodes to construct the data aggregation tree. For the WSNs with different initial node energy, Jadidoleslamy Hossein et.at [13] investigated the data aggregation tree problem in the set of shortest path trees which could maximize the network lifetime. This issue is equivalent to finding a shortest path tree with the lightest load. Then it is solved by transforming the shortest to a generalized semi-matcher. A distributed construction algorithm was proposed for the shortest path tree based on depth priority search and width priority search [14, 15]. Wang et.al [16] proposed a dual-tree routing protocol to employ the reverse link and construct two routing trees based on the MST and the DMDC, respectively. For a WSN with multiple mobile Sinks, Zhang et al [17] designed a branch and bound algorithm and an analogous back-fire algorithm to construct the minimum Wiener index spanning tree to obtain better energy efficiency and delay performance for data. It is required in the GSTEB that the Sink designates the root node according to the node energy and the data fusion type while the other nodes choose their father nodes according to their location information and neighbor information. Finally an energy-balanced routing tree is built in a distributed manner.

According to the probabilistic network model, Meng et al [18] first employed the linear relaxation and random rounding technique to construct a load-balanced maximum independent set. A connected maximum independent set was constructed and the allocation of parent nodes was then performed to balance the load. Finally a direction was assigned to each link to obtain a data aggregation tree. The tree was constructed gradually from the Sink in CASMOC [19] while was guaranteed that all the node share the same number of child nodes. Finally a globally load-balanced routing tree was constructed. Based on an established shortest routing tree, the PM MSW [20] performs iterative modification on the tree structure from the bottom to the top based on the load balancing factor. In iteration, only the child nodes, parent nodes and the grandparent nodes are involved. The transmission cost and data fusion cost are considered in the AFST algorithm to obtained a more energy-efficient data fusion routing tree based on the routing tree constructed by MFST algorithm [21, 22]. The benefits brought by the data fusion was evaluated and employed to adaptively adjust the data fusion behavior. Chen et al [23] design a centralized algorithm with polynomial time complexity and one distributed algorithm based on local information to design the optimal transmission strategy for an arbitrary tree. Meanwhile, each node was allocated a corresponding slot for data transmission to optimize the data fusion. For duty-cycled WSNs, Xing et al [24] constructed the routing tree based on connected support set and proposed the centralized Greedy Aggregation Scheduling (GAS) and Partial Aggregation Scheduling (PAS) based on partitioning.

According to the properties of the routing in WSNs, we perform our study from the following four perspectives Greedy relay strategy based.

(1) The greedy relay strategy could further reduce the energy consumption and reduce the quantity of the hops for the data packets which are from the source node to the sink node.

(2) Predict the routing void. We try to reduce the probability of reaching routing void for the data packets, which could efficiently reduce the routing hops for the data packets.

(3) Avoid the congested links. Nodes in the congested area have to perform many retransmissions to finish transmitting the data packet, which increases the average energy consumption and delay.

(4) Balance the remaining node energy. Excessive use of a certain node would lead to the fast exhaustion of its energy, which further increases the transmission path length for subsequent data packets

3. Data Fusion Mechanism

3.1. Construction of the Neighbor Nodes

For description convenience, we first give some necessary definitions here. For an arbitrary node *i*, *N(i)* is the series of nodes which are neighbor and also can directly communicate with node i and *d(i,j)* denotes the distance of Euclidean between node *i* and node *j* while R is the wireless transmission radius of the nodes. In the geological routing, the nodes could acquire specific information about neighbor nodes through exchanging the Hello message such as the node ID, geological location, working state and remaining energy. While the node *i* gets the data packet from the Sink node *D*, the node *i* calculates the propelling degree of each neighbor node to the Sink node *D* as follows:

$$F(i,j,D)=d(i,D)-d(j,D). \quad (1)$$

The neighbor node *j* with the largest positive $F(i,j,D)$ is then chosen as the relay node of the next hop. If no node exists which satisfies $F(i,j,D)>0$, then the data packet suffers from the routing void at node *i*. We can adopt any recovery mechanism to guarantee the transmission of the data packet. However, no matter which recovery mechanism is chosen, the energy consumption caused by data packet transmission can be rather large compared with greedy relaying. The greedy geological routing algorithm based on the geological information of the multi-hop neighbor nodes requires each node to obtain the information of two or more hops. Then based on this information, the path can be optimized for choosing the next neighbor node. As a result, we can greatly reduce the probability of routing void and the average consumption of the energy for the transmission of the data packets.

3.2. Neighbors Quantification Based Relay Strategy

While the node *i* gets a data packet which is *p* whose destination address is *D*, a node from *N*(*i*) is chosen as the relay node for the next hop according to the relay strategy of the MR-DFM algorithm, which is elaborated as follows.

(1) If node *i* is exactly the destination of the data packet *p*, the data packet transmission is successful.

(2) If node *j* from *N*(*i*) is the destination of the data packet *p*, then that data packet is relayed to node j from node *i*.

(3) If the destination *D* lies within the wireless transmission area of node *j* from *N*(*i*), the data packet *p* is relayed to node *j* from node *i*. If multiple nodes from *N*(*i*) satisfies the above condition, *F*(*i*,*j*,*D*) is calculated according to equation (1) for those nodes. Then the node *j* with the maximal *F*(*i*,*j*,*D*) is chosen as the relay node.

(4) If none of the conditions are satisfied, then for an arbitrary node from *N*(*i*), the potential propulsion degree from this node to *D* is calculated according to the corresponding neighbors quantification value.

$$F_t(i,j,D)=d(i,D)-d(j,D). \tag{2}$$

Where $F_t(j,k,D)$ is the approximate distance from the nodes in the *k*-th sector of node *j* to *D*, which can be calculated as follows.

$$\Gamma_t(j,k,D)=\left(\left(\Gamma_\alpha(j,k)\sin\frac{2\pi k}{m}\right)^2+\left(d(j,D)-\Gamma_\alpha(j,k)\cos\frac{2\pi k}{m}\right)^2\right)^{\frac{1}{2}}. \tag{3}$$

Construct the set of candidate relay nodes $\Phi_{cs}(i)$. If $\Phi_{cs}(i)$ is not empty, then chose the node *j* with the maximal $F_{tn}(i,j,D)$ in $\Phi_{cs}(i)$ as the relay node. Otherwise, data packet *p* is likely to suffer from routing void at any neighbor of node *i*. To reduce unnecessary energy consumption, the edge recovery scheme can be employed in advance at node *i*. $\Phi_{cs}(i)$ is expressed as follows:

$$\Phi_{cs}(i)=\{j|F_{tn}(i,j,D)>F(i,j,D)\},\ j\in N(i). \tag{4}$$

In comparison with the conventional greedy relay strategy, routing void can be avoided by the nodes utilizing the MR-DFM routing data packet. Therefore, the average energy consumption for data packet transmission can be reduced.

3.3. Data Flow Congestion Avoidance

The data packet is propelled towards the destination node along the shortest path utilizing the greedy relay strategy. Normally, one data flow would consistently transmit data packets within a certain period. Traffic congestion can be caused if all the data packets are transmitted along the shortest path, as a result of which the transmission delay is increased. If the congestion degree further increases, the probability of data packet collision grows higher. Meanwhile, the packet loss rate increases due to the

buffering overflow. Furthermore, employing the retransmission recovery scheme would increase the average energy consumption for transmission.

In order to solve this problem, the greedy relay strategy can be modified by introducing a greediness factor g, which indicates the expected minimal propulsion degree. The factor g is real-valued and ranges from 0 to 1. A small g would increase the path length and the energy consumption for transmission. On the contrary, if g is too large, the number of candidate nodes would decrease and congestion avoidance cannot be achieved. Since the propulsion degree for the candidate node is expected to be at least 80% of the maximal propulsion degree here, we take the value g=0.8. To avoid data flow congestion, we construct the set of candidate relay nodes which satisfies the greediness factor based on the set $\Phi_{cs}(i)$.

$$\Phi_{Gcs}(i)=\{j|F_{tn}(i,j,D)>g\max F_{tn}\},\ j\in\Phi_{cs}(i). \tag{5}$$

Where $\max F_{tn}$ is the maximal $F_{tn}(i,j,D)$ in the set $\Phi_{cs}(i)$ for node i. A node with the smallest data flow congestion degree can be chosen as the relay node for the next hop as long as the number of nodes in $\Phi_{Gcs}(i)$ is larger than one.

Considering the sharing property of the wireless links between the nodes in the sensor network, the nodes will have more chances to relay data packets when the surrounding links are idle. Otherwise, data packets will be buffered in the node array. The congestion degree of a node within a period of T_{cwin} is defined as:

$$\Psi_{cwin}(i)=P_{in}(i)/P_{tr}(i). \tag{6}$$

Where $P_{tr}(i)$ is the number of successfully relayed data packets of node i and $P_{in}(i)$ is the average array length for node i within T_{cwin}.

$\Psi_{cwin}(i)$ reflects the node congestion degree within T_{cwin}. However, during the transmission of data packets, it is the future congestion degree after the arrival of data packets that we are always concerned about. In order to predict the future congestion degree for the nodes, we employ the exponentially weighted moving average method and define the node congestion degree as:

$$\Psi_c(i)=(1-\alpha)\,\Psi_c(i)+\alpha\,\Psi_{cwin}(i). \tag{7}$$

α is the weight factor which reflects the real-time varying extent for the node congestion degree. Experiments have shown that when the weight factor takes the value of 0.2, the mean squared error between the predicted value and the realistic value is the smallest. Therefore, we assume α=0.2 here.

Considering the fact that the MR-DFM algorithm employs the greedy relay strategy based on two-hop neighbors, the congestion degree of neighbors should also be taken into consideration for the calculation of node congestion degree. The neighbors' congestion degree is defined as the data flow congestion degree $\Psi_{dc}(i)$.

$$\Psi_{dc}(i)=(1-\beta)\,\Psi_c(i)+\beta A(j),\ j\in N(i). \tag{8}$$

Where $A(j)$ is the average congestion degree for the neighbor j of node i and β is the weight factor for the neighbors.

In order to obtain the data flow congestion degree of the neighbors, $\Psi_{dc}(i)$ of node i can be included in the Hello message. Therefore, we assume T_{cwin} to be equal to the exchange period of Hello message here.

3.4. Node Energy Consumption Balance

As is mentioned above, the choice of the relay node for the next hop would influence the network performance to some extent when there are multiple data flows. The reason is that when one node lies on the greedy relay path of multiple data flows, the node energy would be exhausted quickly. If one node is the only greedy relay path for one of the data flows, the failure of this node will cause routing void to this data flow, which will result in sharply increased average energy consumption for this data flow.

For solving this problem, the remained node energy and the number of data flows being relayed have to be taken into consideration to choose the relay node for the next hop. The data flows corresponding to each data packets have to be identified for the counting of data flows. Here, the data flows are distinguished by the source and destination of the data packets. The information on the source node and destination node is extracted from the received data packets by the nodes. If the data flow passes the node for the first time, it will be recorded in the data flow table at the nodes. Otherwise, corresponding records will be updated in the data flow table. Considering the fact that the memory of sensor nodes is limited, only the data flow within the latest period T_{dwin} is remained. The number of data flows is:

$$\gamma_n(i)=(1-\alpha)\gamma_n(i)+\alpha\gamma_{win}(i). \tag{9}$$

Where $\gamma_{win}(i)$ is the number of data flows within the latest period T_{dwin}. Just like the data flow congestion degree $\Psi_{dc}(i)$ of nodes, $\gamma_n(i)$ is also included in the Hello message. Furthermore, $T_{dwin}=T_{cwin}$.

$\gamma_n(i)$ reflects the importance of node i to the transmission of data packet. In order to avoid the increase of energy consumption for data packet transmission caused by the failure of important nodes, more energy has to be reserved for the nodes with larger $\gamma_n(i)$, which is referred to as the node energy consumption balance scheme.

The energy balance degree of node i is defined as:

$$B_e(i)=E(i)/\gamma_n(i). \tag{10}$$

Where $E(i)$ is the remained energy of node i. A larger $E(i)$ means that node i hopes to relay more data flows or data packets. After the introduction of the node energy balance degree, the relay node weight of each node j in $\Phi_{Gcs}(i)$ is calculated as follows for node i when the relay node for the next hop is chosen based on the greedy relay strategy.

$$W(j)=B_e(j)/\Psi_{dc}(j) \quad j\in\Phi_{Gcs}(i). \tag{11}$$

Then the node with the largest $W(j)$ is chosen as the relay node for the next hop. It is shown from the definition of $W(j)$ that node i usually choose the neighbor with smaller congestion degree and higher energy balance degree as the relay for the next hop.

We consider three cases based on the sequential order of the two data flow.

Case 1. The data flow (B,D) comes before flow (A,E). There will be two candidate relay nodes for flow (B,D), node C, node F and i.e.. The data packet that is the first can choose an arbitrary candidate node (which is assumed to be F). But for the second data packet, since $W(C)>W(F)$, only C is left as the candidate relay node for the next hop.

So, when the 10 data packets are relayed for the data flow (*B*,*D*), 5 packets are relayed by each one of node *C* and node *F*. The first 5 data packets are relayed through node *F* while the last 5 packets through node *B*. The average energy consumption is 2.5, which is normal.

Case 2. The data flow (*B*,*D*) comes after flow (*A*,*E*). Since there is only one greedy path for flow (*A*,*E*), all the 10 data packets are relayed through node *F*. The energy consumption is 2 on average, which is optimal.

Case 3. When the two flows come simultaneously, at most one data packet of flow (*B*,*D*) chooses F as the relay for the next hop. Therefore, $\gamma_n(F)$=2while $\gamma_n(C)$=1. The remaining 9 data packets of flow (*B*,*D*) choose *C* as the relay for the next hop, while at most one data packet of flow (*A*,*E*) chooses *B*. The average energy consumption is 2.06, which is near optimal.

Judging from what is analyzed above, the average energy consumption lies between the optimal value and the normal value, regardless of the sequential order of the two flows. The target set for the MR-DFM algorithm is therefore achieved.

4. Multi-Routing Control Algorithm

4.1. Choice of Cluster Head

When the event occurs, the nodes in the event domain are distributed into clusters while the cluster heads take charge of the node management and data aggregation. The choices of cluster heads are crucial to the formation of clusters and we can adopt different strategies to maximize the node degree or the remaining node energy, or minimize the identifiers (ID). For the convenience of comparison, we adopt the same choosing strategies for cluster head as in HDAC. The node with the smallest ID in the event domain is chosen as the cluster head. In this stage, the nodes acquire the event monitoring state of neighbors through exchanging the Detecting Message (DM) and further employ the Cluster-Head Announcement (CA) message to contend for the position of cluster head. Both the CA messages and DM are 3-tuples, respectively denoted as <Type, CH_ID, S_ID> and <Type, ID, E_ID>. In the messages, while *ID* is the node identifier ,Type indicates the message type, *E_ID* is the identifier of the event, *CH_ID* is the identifier of the cluster head and *S_ID* is the ID of the node which relays this CA message. Correspondingly, each node maintains 4 domains, i.e., Role (node character indicator, *CH* for cluster head or CM for cluster member), *CH_ID1* (official cluster head), *NH_C* (the next hop in the cluster) and *CH_ID2*(temporary cluster head). After the detection of the event, the node employs the DM message to acquire the event monitoring status and ID of the neighbor node. If the ID of the node is the smallest, then it is chosen as the cluster head candidate. Otherwise, it can only be the cluster member whiles the node with the smallest ID in the event which is monitored by its neighbors is chosen as the temporary cluster head. Next, the nodes with the Role of CH obtain the transmission delay of its CA message according to equation (12)

$$Delay=(T_{CH}-\tau)\times(X/X_{max}). \quad (12)$$

Where T_{CH} is the duration of the contention sub-period for the cluster head, τ is the time required for the CA message to be transmitted all through the event domain which is given empirically, X is the ID of the node with the Role of CH while X_{max} is the largest ID of the nodes.

Through the delayed relay of the CA message, the node which is with the smallest ID will be chosen as the head of the cluster while the corresponding routing structure within the cluster will also be determined. The transmission delay is introduced to guarantee that the cluster head candidate with smaller ID could send the CA message earlier while the amount of CA message sent by cluster head with larger ID can be reduced. Therefore, the control signaling can be efficiently saved.

The choice of cluster head among 10 nodes is illustrated in Fig.1 and the initial distribution of the nodes is shown in Fig.1(a). The nodes first ensure whether they are the cluster head through exchanging the DM message, as shown in Fig.1(b). Next, through a series of relaying of the CA message, node 1 is chosen as the cluster head, which is shown in Fig.1(c)-(f).

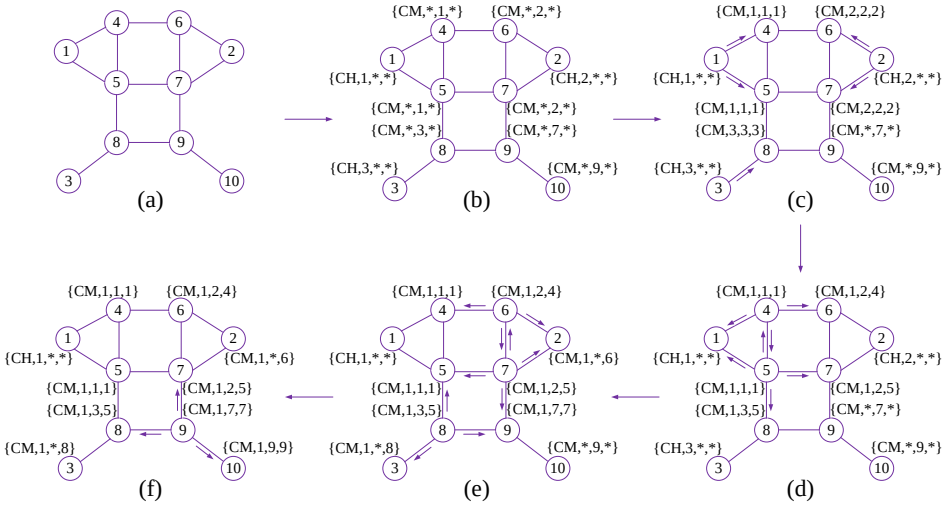


Fig. 1. cluster head routing structure

4.2. Communication Correlation among the Nodes

Theorem 1: During the dialogue between terminal *a* and terminal *b*, when no node synchronization is performed, the newly-built dialogue of the exposed terminal *c* will cause no interference on the existing dialogue if $f(Cov(a,b),Cov(c,d))\leq 1/(N+1)$, where terminal *d* is the destination of the exposed terminal *c* and $N=10^{(\gamma/10)}$.

Proof: If the exposed terminal *c* is a neighbor of terminal *b* and terminal *a* is silent, then according to equation (2), the critical condition for terminal *b* to correctly receive the data from terminal *c* is:

$$Pr_{(c\to b)}=(\phi+N_0)\times N. \tag{13}$$

When terminal *a* and terminal *c* transmit data simultaneously, the condition for terminal *b* to correctly receive the data from terminal *a* is：

$$Pr_{(a\to b)}\geq(\phi+Pr_{(c\to b)}+N_0)\times N. \tag{14}$$

Substituting equation (13) into equation (14), the condition for terminal *b* to correctly receive the data from terminal *a* is:

$$Pr_{(a\to b)}/\ Pr_{(c\to b)}\geq N+1. \tag{15}$$

Similarly, when terminal *b* and terminal *c* transmit data simultaneously, the condition for terminal *a* to correctly receive the data from terminal *b* is:

$$Pr_{(b\to a)}/\ Pr_{(c\to a)}\geq N+1. \tag{16}$$

Generalizing equation (15), equation (16) and Assumption 2, the condition for transmission of terminal *c* to pose no influence on *Cov*(*a*,*b*) is:

$$Pr_{(b\to a)}/\ \max(Pr_{(c\to a)},Pr_{(c\to b)})\geq N+1. \tag{17}$$

Similarly, we can derive the condition for transmission of terminal *d* to pose no influence on *Cov*(*a*,*b*). Therefore, the condition for *Cov*(*c*,*d*) to pose no influence on *Cov*(*a*,*b*) is:

$$Pr_{(b\to a)}/\ \max(Pr_{(d\to a)},Pr_{(d\to b)},Pr_{(c\to a)},Pr_{(c\to b)})\geq N+1. \tag{18}$$

Finally, we can derive the condition for the case that *Cov*(*a*,*b*) and *Cov*(*c*,*d*) do not influence each other, i.e.,

$$\min(Pr_{(b\to a)},Pr_{(c\to d)})/\ \max(Pr_{(d\to a)},Pr_{(d\to b)},Pr_{(c\to a)},Pr_{(c\to b)})\geq N+1 \tag{19}$$

The proof is completed.

Corollary 1: When no node synchronization is performed, the condition for the exposed terminal c to build a new dialogue during the dialogue between terminal a and terminal b is:

$$f'=DX/DM\geq N' \tag{20}$$

Where *d* is the destination terminal of terminal *c*, $DX=min\{d(a,c),d(b,c),d(a,d),d(b,d)\}$.

Proof: because $DM=max\{d(a,b),d(c,d)\}$ and $N'=(N+1)^{1/\lambda}$.

$$\frac{Pr_{(a\to b)}}{Pr_{(c\to b)}}=\frac{GPt_a/d(a,b)^{\lambda}}{GPt_c/d(c,d)^{\lambda}}=\frac{d(a,b)^{\lambda}}{d(c,d)^{\lambda}}\geq N+1\Rightarrow\frac{d(a,b)}{d(c,d)}\geq(N+1)^{\frac{1}{\lambda}} \tag{21}$$

.

According to the deriving process which is similar to that of Theorem 1, we have:

$$\frac{\min\left(d\left(a,c\right),d\left(b,c\right),d\left(a,d\right),d\left(b,d\right)\right)}{\max\left(d\left(a,b\right),d\left(c,d\right)\right)}=\frac{DX}{DM}\geq N' \tag{22}$$

The proof is completed.

4.3. Establishing the algorithm

When the cluster is formed based on the event domain or the event is finished, the cluster head has to transmit its location information to the PS through the existing hop tree. According to the information of the cluster heads, the PS calculates the geometric center among the PS and the cluster heads according to equation (21). This center is then chosen as the RAC (Routing Aggregation Center) of the network and further broadcasted all across the network.

$$x_{RAC}=\begin{cases}\dfrac{x_{PS}+\sum\limits_{v\in CHSet}x_v}{1+\left|CHSet\right|} & \left|CHSet\right|>1\\ x_1 & \left|CHSet\right|=1\end{cases} \tag{23}$$

Where *CHSet* is the set of cluster heads, x_{RAC}、x_{PS}、$x_v \in CHSet$ are the locations of the RAC, PS and the cluster head, respectively. After the acquisition of the RAC location on the nodes in the network, the nodes is chosen for the next hop which needs the least hops to the PS and closet to the RAC. The algorithms for establishing the MR-DFM routing and updating the routing are explained in details in Table 1 and Table 2, respectively.

Table 1 Algorithm for choosing the cluster head in the MR-DFM

1. FOR each node *u* which detected the event
2. a DM is sent from *u* to its neighbors and *u* waits for a proper time to receive DMs , indent and adjust these things in whole code below. Better use another font for algorithm.;
3. IF ID(*v*) is bigger than any ID(*u*)
4. Role(*u*)=CH; CH_ID1(*u*)=ID(*u*); CH_ID2(*u*)=NULL;
5. ELSE
6. Role(u)=CM; CH_ID1(u)=NULL; CH_ID2(u)=w;
7. IF Role(u)= =CH
8. *u* calculates the delay of CA according to Formula (20) and sets a timer;

```
9.      u sends a CA out within the event scope when the timeout happens;
10.   WHILE u receives a CA
11.         IF CH_ID1(u)= =NULL
12.             IF CH_ID2(u)<CH_ID(CA)
13.                 u abandons the CA;
14.             ELSE
15.                 CH_ID1(u)=CH_ID(CA);NH_C(u)=S_ID(CA);
16.                  S_ID(CA)=u,u retransmits the CA;
17.         ELSE
18.                  IF CH_ID(CA)<CH_ID1(u)
19.                 Do the identical operations that is shown in Lines 15-18;
20.                 ELSE
21.                 u abandons the CA;
```

Table 2 Algorithm for updating the MR-DFM routing

```
1. IF an event finishes or occurs
2.    The case to the PS was reported by the cluster head of the event;
3.    IF CHSet≠Φ
4.      The primary Sink calculates the Routing Aggregation Center (RAC)
according to Formula (21), and broadcasts it to the whole network;
5.          Each node u finds a neighbor v that satisfies: a)the HTS level is
lower than that of u, b) the Euclidean distance to the RAC is the smallest.
6.          NH(u)=v;
```

According to the algorithm for choosing the MR-DFM cluster head, the node is required to make one decision every time it receives one DM message so that it determines whether it should be involved in the contention for the cluster head. The time complexity for this algorithm is $O(N_N)$ while the spatial complexity is $O(N_N)$ where N_N is the average number of neighbor nodes. Next the nodes involved in the contention calculate the delay of its CA message and transmits its TA message after the delay. The time complexity of this algorithm is $O(1)$ while its spatial complexity is $O(1)$. Every time the node receives a CA message, it is required to make a decision for assuring the cluster head to finally finish the distributed clustering. The time complexity of this algorithm is $O(N_{CH})$ while its spatial complexity is only $O(1)$, where N_{CH} is the number of cluster head candidates. Therefore, we can know that the time

complexity for choosing the cluster head in the MR-DFM is $O(\max(N_N, N_{CH}))$ while its spatial complexity is $O(N_N)$.

5. System Evaluation and Simulations

In this paper, we run the simulations to verify the performances of the MR-DFM algorithm on NS-2 (Version NS2.33). We further make comparisons with the conventional greedy routing GSPR algorithm [5] and the two-hop neighbor based Greedy-2 algorithm. The evaluation index is the average hop number, end-to-end delay for transmitting data packets and average energy consumption. I am employed in the simulations while the total number of sensor nodes deployed in each topology is 200. The 802.11 protocol is employed in the MAC layer and the queue length is 50 data packets. The transmission rate of data packets is 11Mb/s and the node socket type is hybrid. The communication radius is 250m. For each type of network topology, 12 different node deployment scenarios are generated uniformly and randomly. The running duration for each simulation is 500s. 10 data streams are randomly chosen among the nodes. The data packet is directly abandoned when it suffers from a routing void. Finally, the results are provided by averaging over the 12 experiments.

The average quantity of hops for transmission of the data packet is illustrated in Fig. 2. It is shown that when the number of neighbor nodes is less than 15, the quantity of hops increases with the quantity of neighbors. Especially when there are 5 neighbor nodes, the GPSR algorithm requires fewer hops than the MR-DFM algorithm and the Greedy-2 algorithm. This is due to the fact that when there are fewer neighbors, the number of routing voids is large. Since both the MR-DFM algorithm and the Greedy-2 algorithm adopt the two-hop based relaying strategy, the routing void can be efficiently predicted and thus avoided. The data transmission can also be finished when the source of the data flow is relatively far from the Sink for these two algorithms. However, the GPSR algorithm works only when the source is close to the Sink node. On the other hand, when the number of neighbor nodes is larger than 15, the average number of hops tends to decrease with the number of neighbors. They can be explained by the fact that the number of routing voids are reduced with increasing number of neighbors and the average distance from the source to the Sink tends shorter.

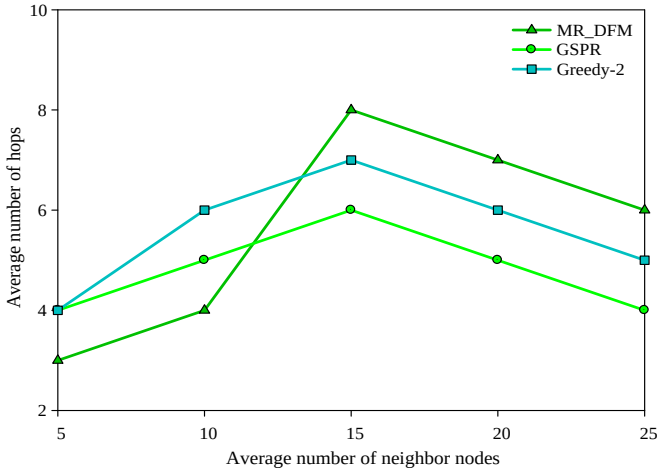


Fig. 2. The average quantity of hops for transmission of the data packet

The average energy consumption over time for successfully transmitting one data packet is illustrated in Fig. 3 when the number of neighbor nodes is 15. The average energy consumption is shown to increase with time. For example, in the first 300s of this experiment, the TNEB algorithm and the Greedy-2 algorithm require more energy consumption than the GPSR algorithm. As for the reason, the MR-DFM algorithm and the Greedy-2 algorithm need to exchange the information of the two-hop neighbors through the Hello message, which causes additional energy consumption. Since the greedy relaying strategies of the Greedy-2 algorithm and the algorithm of GPSR do not consider the node energy balance, the transmission of subsequent data packets would experience obviously longer paths when the nodes on the shortest path runs out of energy. As a result, the average energy consumption increases drastically for the GPSR algorithm and the Greedy-2 algorithm. However, the TNEB algorithm considers the energy balance as well as the node data stream in the greedy relaying process, the nodes can be appropriately assigned for relaying data packets. Therefore, the energy consumption is almost the same for different periods. At the end of the experiment, compared with the Greedy-2 algorithm and the GPSR algorithm, the MR-DFM algorithm reduces the average energy consumption by 13.5% and 23.6%, respectively.

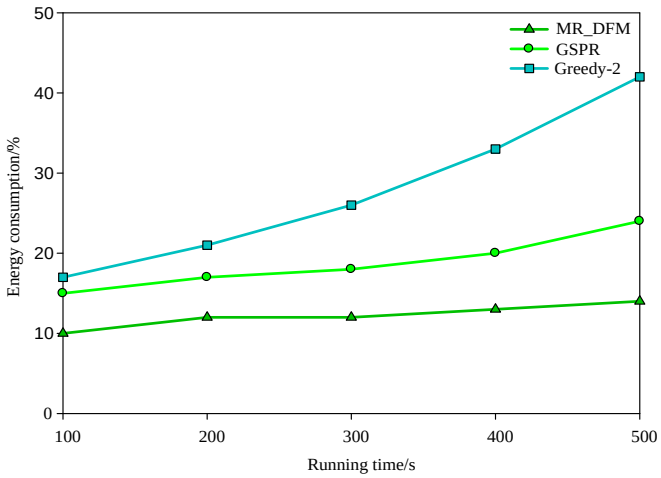


Fig. 3. The average energy consumption for data packet transmission over time

Since the MR-DFM algorithm chooses better nodes for the next hop in the routing updating phase for the sake of data aggregation, the amount of transmitted data in the network for the MR-DFM algorithm is lower than the amount of the HDAC algorithm, which is illustrated in Fig.4. The flooding strategy inside the network is adopted by the HDAC to update the sum of the distance from the nodes to each cluster heads. As a result, far more controlling signaling is required by the HDAC algorithm than the MR-DFM algorithm, the PM_MSW algorithm and the DMDC algorithm. It is won in Fig.5 that with the increasing communication radius, the clustering controlling signaling is increased for the MR-DFM algorithm and the HDAC algorithm, while the MR-DFM algorithm shows a smaller increase. As for the reason, the amount of transmitted CA message can be efficiently reduced by the delay mechanism of the CA message in the MR-DFM algorithm. Since the complete aggregation is adopted, the routing efficiency of all the algorithms increases with the radius of the event domain. Herein, as shown in Fig.6, the MR-DFM algorithm exhibits the highest routing efficiency while the HDAC comes second. It is shown in Fig.7 that the MR-DFM algorithm exhibits the smallest total energy consumption since the MR-DFM algorithm could effectively aggregate data and requires less controlling signaling.

When the radius of communication (35m) and the range of the network (1000m×1000m) are fixed, the quantity of nodes changes to 6400 from 3575 and we can obtain the performance with different node density for the MR-DFM algorithm in Fig. 8- Fig. 11. It is shown that the node density increases with more nodes in the event domain, which causes larger amount of data transmission and more controlling signaling in Fig. 8- Fig. 9. It can be observed from Fig.10 that the routing efficiency increases with the node density. The routing efficiency of the MR-DFM algorithm is better than the other three algorithms since it requires the least amount of data transmission. Therefore, the MR-DFM algorithm is still the best in terms of the total energy consumed, as shown in Fig.11.

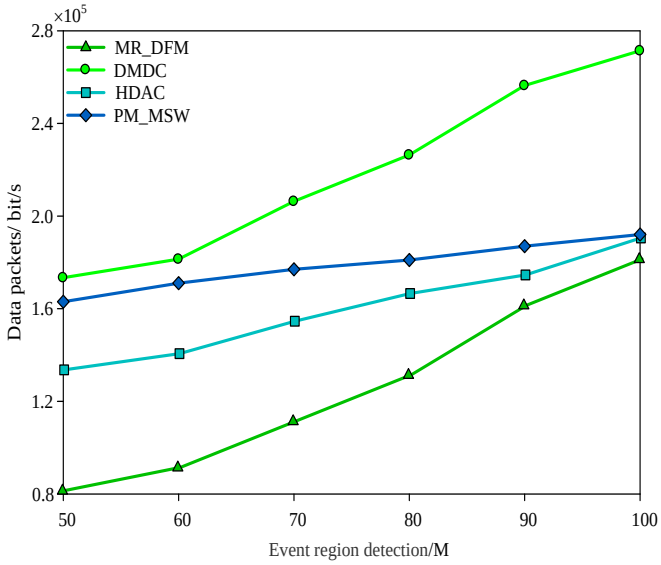


Fig. 4. Performance comparisons with data clustering

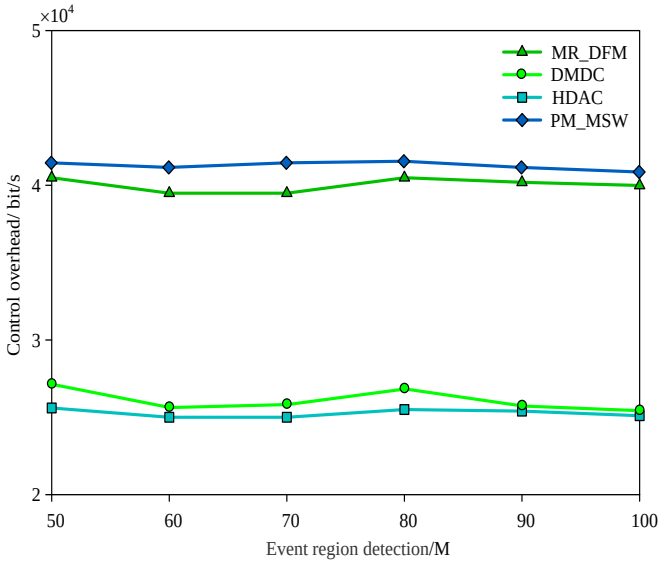


Fig. 5. Performance comparisons with controlling signaling

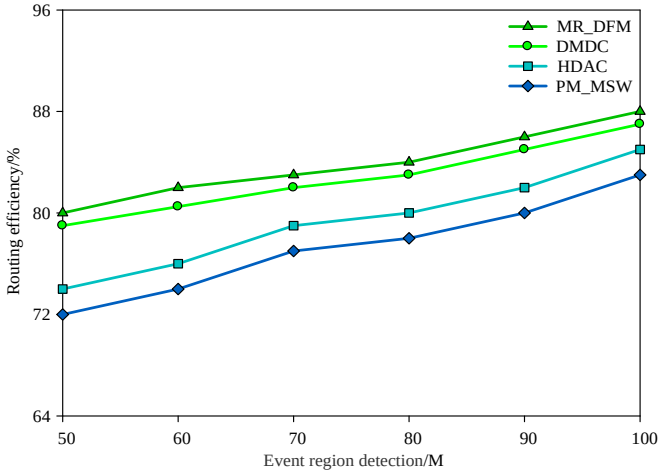


Fig. 6. Performance comparisons with routing efficiency

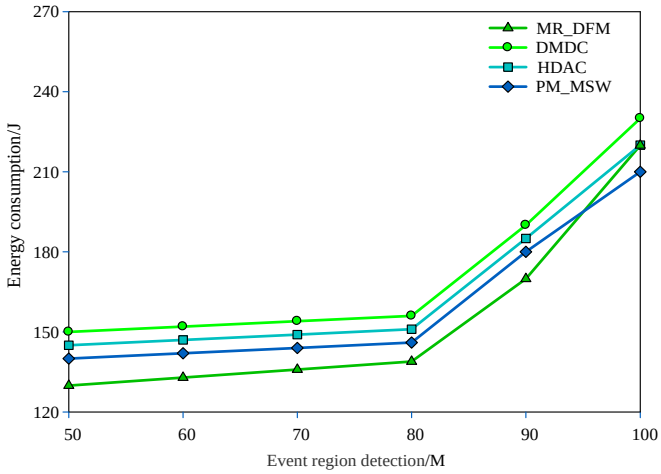


Fig. 7. Performance comparisons with total energy consumed

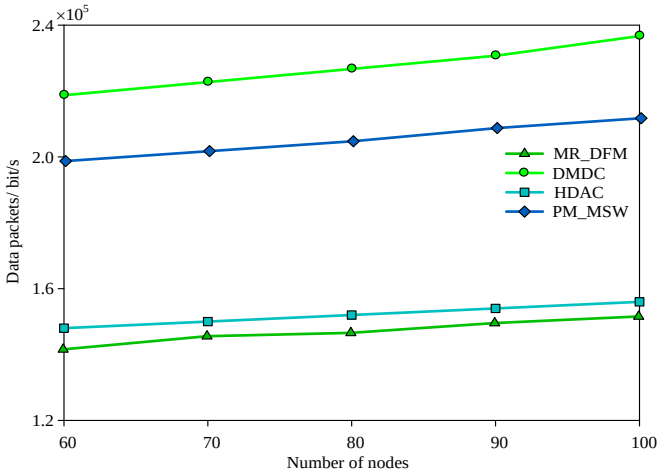


Fig. 8. Performance comparisons with data clustering

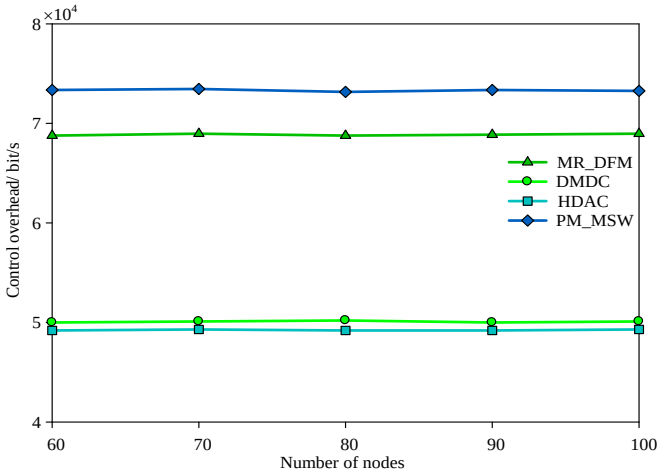


Fig. 9. Performance comparisons with controlling signaling

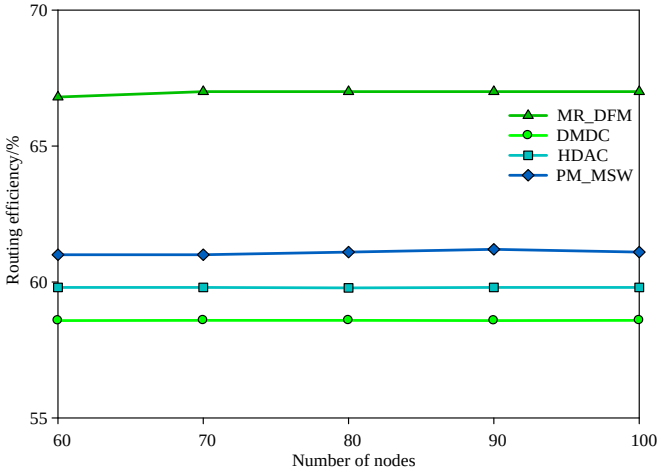


Fig. 10. Performance comparisons with routing efficiency

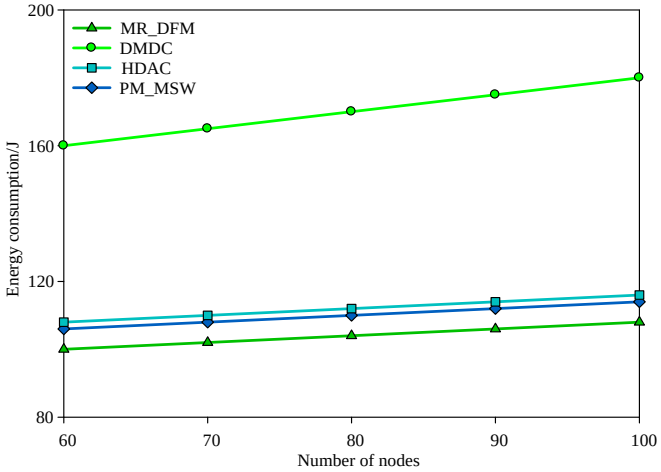


Fig. 11. Performance comparisons with total energy consumed

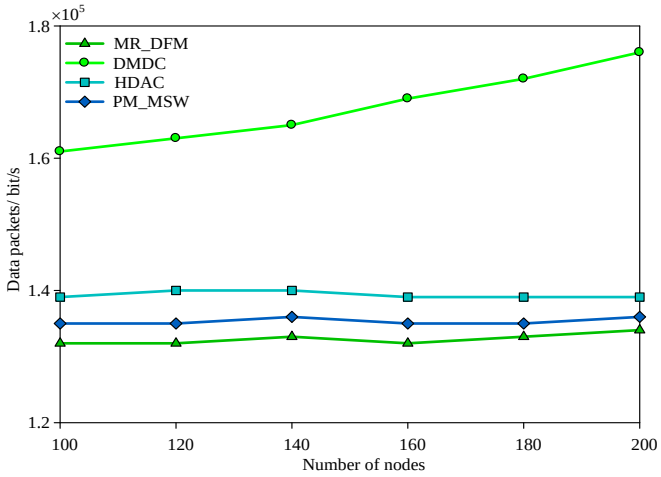


Fig. 12. Performance comparisons with data clustering

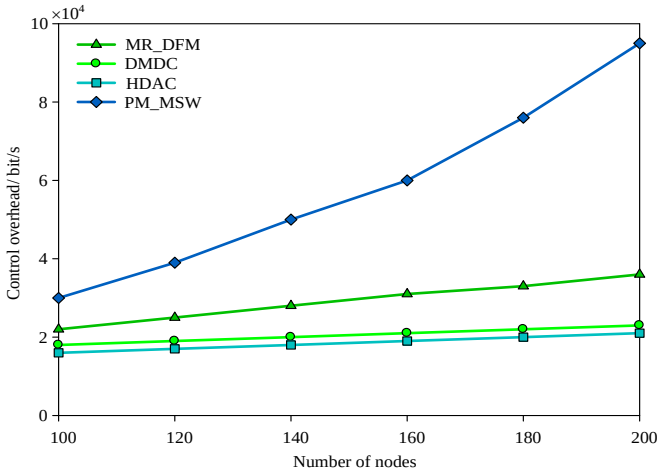


Fig. 13. Performance comparisons with controlling signaling

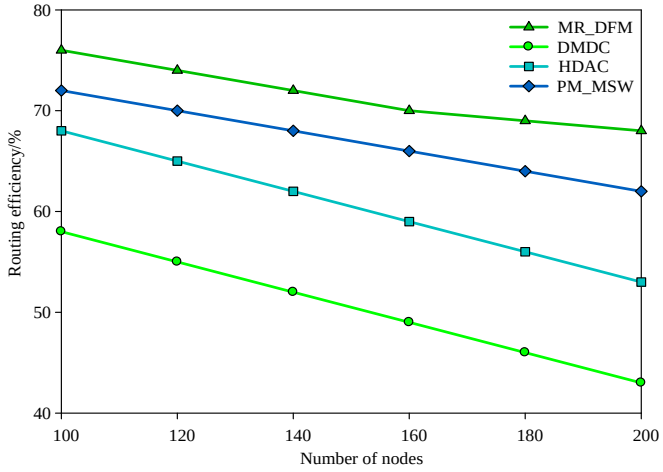


Fig. 14. Performance comparisons with routing efficiency

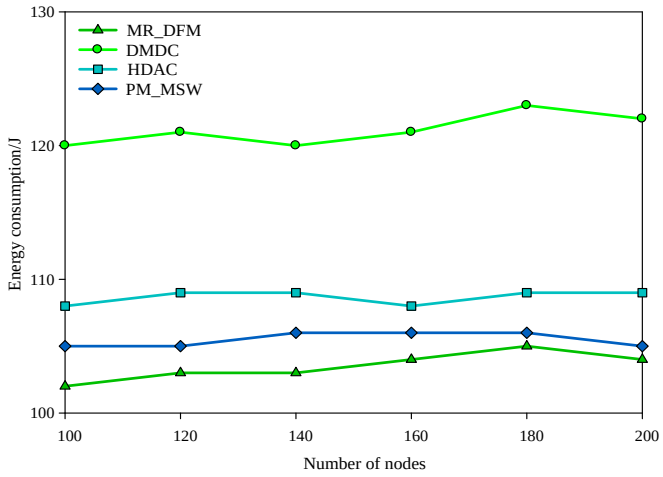


Fig. 15. Performance comparisons with total energy consumed

When the area of the network becomes 500m×500m and the radius of the communication becomes 10m, even the number of the nodes changed from 100 to 200, the situation of the algorithm performance is shown in Fig.12- Fig.15. The performance of the algorithm in this paper is higher than other 3 algorithms. With the increase of the node density, the neighbor of the first hop of other 3 kinds of algorithm also increases. Therefore, the average energy consumption of the first hop neighbor in other 3 algorithms are increased. Compared with the 3 algorithms, the energy-efficient result of the algorithm in this paper is higher than 31.26%, 12.26%, 9.68% and 6.31%, respectively. In the aspect of the loss of the data package, the algorithm in this paper is higher than other 3 algorithms as 41.39%, 17.56%, 12.52% and 9.16%, respectively.

In the aspect of the efficiency of routing, the algorithm in this paper is higher than other 3 algorithms as 37.88%, 26.15%, 18.62% and 13.71%, respectively.

6. Conclusions

A multi-path routing algorithm based on data-fusion-mechanism (MR-DFM) was proposed. After the event occurs, this algorithm could achieve the distributed clustering of the nodes in the event domain with less controlling signaling. After that, the information about the cluster head is sent to the main base station PS by the CH node. The PS calculates the geometric center among itself and all the cluster heads and broadcast the information of center as the routing aggregation center (RAC) across the network. When any node obtains the information of the RAC, it updates the next hop according to its partial state information and finally establishes an approximate Steiner tree. The auxiliary station (AS) moves to the RAC after it receives the information of the RAC and adjusts its location according to the data transmission status of the surrounding neighbors until it can successfully intercept the data. The MR-DFM algorithm could optimize the data aggregation efficiency. Thus, it requires less controlling signaling for the construction and maintenance of the routing. Meanwhile, through the movement of the AS, the information in the network is intercepted to further reduce and balance the data transmission amount in the network as well as alleviate the “hotspot” issue. As a result, an efficient data routing and gathering can be achieved and the network performance is improved. The analyses and simulation results verified the performances of the MR-DFM algorithm.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No.U1604149); Henan Education Department Young Key Teachers Foundation of China (No.2016GGJS-158); Henan Education Department Natural Science Foundation of China (No.19A520006); High-level Research Start Foundation of Luoyang Institute of Science and Technology(No.2017BZ07); Key Science and Technology Program of Henan Province under Grant 182102210428.

References

1. Meena, U., Sharma, A.: Secure Key Agreement with Rekeying Using FLSO Routing Protocol in Wireless Sensor Network. Wireless Personal Communication, Vol.101, No.2, 1177-1199. (2018)
2. Mohamed, L.M., Hamida, S.: EAHKM+: Energy-aware Secure Clustering Scheme in Wireless Sensor Networks. International Journal of High Performance Computing and Networking, Vol.11, No.2, 145-155. (2018)
3. Zhou, Y. Z., Peng, D. P.: A New Model of Vehicular Ad Hoc Networks Based on Artificial Immune Theory. International Journal of Computational Science and Engineering, Vol.16, No.2, 132-140, (2018)
4. Wang, T., Zhou, J. Y., Huang, M. Z.: Fog-based Storage Technology to Fight with Cyber Threat. Future Generation Computer Systems, No. 83, 208-218. (2018)

5. Sasirekha, S., Swamynathan, S.: Cluster-chain Mobile Agent Routing Algorithm for Efficient Data Aggregation in Wireless Sensor Network. Journal of Communication and Networks, Vol.19, No.4, 392-401. (2017)
6. Gao, X. F., Fan, J. H., Wu, F.: Approximation Algorithms for Sweep Coverage Problem with Multiple Mobile Sensors. IEEE/ACM Transactions on Networking, Vol.26, No.2, 990-1003. (2018)
7. Liu, Z. M., Jia, W. J., Wang, G. J.: Area Coverage Estimation Model for Directional Sensor Networks. International Journal of Embedded Systems, Vol.10, No.1, 13-21. (2018)
8. Putra, E. H., Hidayat, R., Widyawan, N.: Energy-efficient Routing Based on Dynamic Programming for Wireless Multimedia Sensor Networks. International Journal of Electronics and Telecommunications, Vol.63, No.3, 279-283. (2017)
9. Guo, P., Liu, X. F., Cao, J. N.: Lossless In-network Processing and Its Routing Design in Wireless Sensor Networks. IEEE Transactional on Wireless Communications, Vol.16, No.10, 6528-6542. (2017)
10. Sun, Z. Y., Ji, X. H.: HDAC: High-dimensional Data Aggregation Control Algorithm for Big Data in Wireless Sensor Networks. International Journal of Information Technology and Web Engineering, Vol.12, No.4, 72-86. (2017)
11. Nie, Y. L., Wang, H. J, Qin, Y. J.: Distributed and Morphological Operation-based Data Collection Algorithm. International Journal of Distributed Sensor Networks, Vol.13, No.7, 1-15. (2017)
12. Zhao, L. Q., Chen, N.: An improved Zone-based Routing Protocol for Heterogeneous Wireless Sensor Networks. Journal of Information Processing Systems, Vol.13, No.3, 500-517. (2017)
13. Jadidoleslamy, H.: A Hierarchical Multipath Routing Protocol in clustered Wireless Sensor Networks. Wireless Personal Communications, Vol.96, No.3, 4217-4236. (2017)
14. Naidja, M., Bilami, A.: A Dynamic Self-organizing Heterogeneous Routing Protocol for Clustered WSNs. International Journal of Wireless and Mobile Computing, Vol.12, No.2, 131-141. (2017)
15. Das, H. S., Bhattacharjee, S.: A congestion aware routing for lifetime improving in grid-based sensor networks. Journal of High Speed Networks, Vol. 23, No.1, 1-14.(2017)
16. Wang, T., Peng, Z., Liang, J. B.: Follow targets for mobile tracking in wireless sensor networks. ACM Transactions on Sensor Networks, Vol.12, No.4, 31. (2016)
17. Zhang, Y. Q., Li, Y. R.: A Routing Protocol for Wireless Sensor Networks Using K-means and Dijkstra Algorithm. International Journal of Advanced Media and Communication, Vol.6, No.2-4, 109-121. (2016)
18. Meng, X. L., Shi, X. C, Wang, Z.: A Grid-based Reliable Routing Protocol for Wireless Sensor Networks with Randomly Distributed Clusters. Ad Hoc Networks, No.51, 47-61. (2016)
19. Sun, Z. Y., Zhang, Y. S., Nie, Y. L.: CASMOC: A Novel Complex Alliance Strategy with Multi-objective Optimization of Coverage in Wireless Sensor Networks. Wireless Networks, Vol.23, No.4, 1201-1222. (2017)
20. Wang, T., Peng, Z., Wu, W. Z: Cascading Target Tracking Control in wireless Camera Sensor and Actuator Networks, Asian Journal of Control, 2017,19(4):1350-1364.
21. Ghosh, P., Ren, H., Banirazi, R.: Empirical evaluation of the heat-diffusion collection protocol for wireless sensor networks. Computer Networks, No.127, 217–232. (2016)
22. Wang, S., Yi, H., Wu, L. N.: Mining Probabilistic Representative Gathering Patterns for Mobile Sensor Data. Journal of Internet Technology, Vol.18, No.2, 321-332. (2017)
23. Chen, Z. S., Shen, H.: A Grid-based Reliable Multi-hop Routing Protocol for Energy-efficient Wireless Sensor Networks. International Journal of Distributed Sensor Networks, Vol.14, No.3, 1-13. (2018)

24. Xing, G. L., Li, M. M., Wang, T.: Efficient Rendezvous algorithms for mobility- enabled wireless sensor networks. IEEE Transactions on Mobile Computing, Vol.11, No. 1, 47-60. (2012)

Zeyu Sun received the B.S. degree in computer science and technology from the Henan University of Science and Technology, in 2003, and the M.Sc. Degree from Lanzhou University, in 2010, and the Ph.D degree form Xian Jiaotong Unviersity, in 2017. He is currently an Associate Professor with the School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, Henan, China. His research interests include wireless sensor networks, mobile computing and Internet of Things.

Zhiguo Lv received the B.S. degree in applied electronic technology form Henan Normal University, in 2000, and the M.Sc. degree in communication and information system from the Guilin University of Electronic Technology, in 2008. He is currently pursuing the Ph.D. degree with Xidian University. He is a Lecturer with the Luoyang Institute of Science and Technology. His research interests include compressive sensing, wireless sensor networks and MIMO system.

Yue Hou is currently pursuing the M.S. degree with Luoyang Institute of Science and Technology, Her research interests include compressive sensing, wireless sensor networks.

Chen Xu (Corresponding Author) received the M.Sc. degree from Lanzhou Jiaotong University, in 2010. He is currently pursuing the Ph.D. degree with Tongji University. He is a Lecturer with Shanghai Institute of Technology. His research interests include mobile computing, Intelligent Transportation System and wireless sensor networks.

Ben Yan received the M.Sc. degree in Department of Information Science Okayama University of Science, Japan, in 2006, and Ph.D. degree in the Graduate School of Information Science at Nara Institute of Science and Technology, Japan, in 2008. His main research interests include the service-oriented architecture, mobile computing and safety critical systems.

Received: September 17, 2018; Accepted: August 21, 2019.

Computer Science and Information Systems 16(3):891–914 https://doi.org/10.2298/CSIS180828032W

Research of MDCOP Mining Based on Time Aggregated Graph for Large Spatio-temproal Data Sets

Zhanquan Wang[1,2], Taoli Han[1], and Huiqun Yu[1,2]

[1] Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
zhqwang@ecust.edu.cn, 2730023187@qq.com, yhq@ecust.edu.cn
[2] Shanghai Engineering Research Center of Smart Energy, Shanghai, 200237, China

Abstract. Discovering mixed-drove spatiotemporal co-occurrence patterns (MDCOPs) is important for network security such as distributed denial of service (DDoS) attack. There are usually many features when we are suffering from a DDoS attacks such as the server CPU is heavily occupied for a long time, bandwidth is hoovered and so on. In distributed cooperative intrusion, the feature information from multiple intrusion detection sources should be analyzed simultaneously to find the spatial correlation among the feature information.In addition to spatial correlation, intrusion also has temporal correlation. Some invasions are gradually penetrating, and attacks are the result of cumulative effects over a period of time. So it is necessary to discover mixed-drove spatiotemporal co-occurrence patterns (MDCOPs) in network security. However, it is difficult to mine MDCOPs from large attack event data sets because mining MDCOPs is computationally very expensive. In information security, the set of candidate co-occurrence attack event data sets is exponential in the number of object-types and the spatiotemporal data sets are too large to be managed in memory. To reduce the number of candidate co-occurrence instances, we present a computationally efficient MDCOP Graph Miner algorithm by using Time Aggregated Graph. which can deal with large attack event data sets by means of file index. The correctness, completeness and efficiency of the proposed methods are analyzed.

Keywords: Network spatiotemporal co-occurrence pattern intrusion detection, mixed-drove spatiotemporal co-occurrence pattern, large spatiotemporal data set, Time Aggregated Graph (TAG), file index.

1. Introduction

In network security area, intrusion detection is the detection of any intrusion that attempts to compromise the integrity, confidentiality, or availability of computer resources. Intrusion detection collects and analyzes the information of key nodes in a computer network or system to detect the behaviors and signs of security policy violations and attacks in the network or system, and to respond accordingly.

The first element of intrusion detection is data source, which usually includes host-based data source and network-based data source. The current intrusion behavior is gradually changing from the messy, unorganized and single invasion behavior to the distributed, organized and cooperative invasion behavior. In distributed cooperative intrusion, multiple attackers and multiple IP addresses can spy on and attack a common target simultaneously. In this collaborative intrusion behavior, the invasion of the attack from multiple

points to invasion by the invasion of the source of the information is very fragmented, from a single intrusion detection information source may can't see anything suspicious, cannot determine whether have intrusion behavior, only at the same time analysis of characteristic information from multiple source of intrusion detection, is likely to determine the intrusion behavior. The correlation between the feature information of this intrusion is called spatial co-occurrence.

In addition to spatial co-occurrence, intrusion also has temporal co-occurrence. Some invasions are gradually penetrating, and attacks are the result of cumulative effects over a period of time. Analysis from some intrusion behavior, for example, the characteristics of the information can be found that Δ t period of network packet with the same source IP address, the address is trying to establish multiple connections with the same target host system. For this kind of intrusion behavior, if any time just thinks of this time period in isolation of characteristic information can reflect the attack, must consider all the feature information Δ t period.

Intrusion detection which takes spatial and temporal into account is mixed-drove spatiotemporal co-occurrence patterns detection.

As the volume of spatiotemporal data continues to increase significantly due to both the growth of database archives and the increasing number of spatiotemporal sensors, automatic and semi-automatic pattern analysis becomes more essential. It is meaningful and challenging for us to extract interesting patterns from these large spatiotemporal attack event data sets. Co-occurrence Pattern represents subsets of different object-types whose instances are co-located together for a significant fraction of time.

Given a large spatiotemporal attack event database, a neighbor relationship and mixed-drove interest measure thresholds, our aim is to discover mixed-drove spatiotemporal co-occurrence patterns (MDCOPs). To mine co-occurrence patterns, Celik et al. proposed MDCOP-Miner and fast MDCOP-Miner[14]. The two methods are based on the join-based collocation algorithm proposed by Huang et al.[18]. The basic co-occurrence pattern mining procedure involves four steps, as shown in Fig.1. First, candidate co-occurrence instances are gathered from the spatiotemporal data sets. Then, prevalent co-occurrence pattern sets satisfying the given prevalence thresholds are filtered. Finally, co-occurrence patterns satisfy the given prevalence thresholds are generated. Most of the computational time of co-occurrence pattern mining is devoted to finding co-occurrence instances. The approach is Apriori-like algorithm, which is costly as it enumerates all possible co-occurrence instances over all time instances. Thus, we propose adding a step for materializing the neighbor relationships to increase the efficiency of co-occurrence mining.

While the volume of the spatiotemporal data set is large, we find discovering the MDCOPs in a relatively small computer memory are difficult by using the existed methods. Mining the co-occurrence patterns is meaningful to network security.

1.1. Contributions

This study makes the following contributions:

We propose a novel method for materializing the neighbor relationships in order to perform efficient mixed-drove spatiotemporal co-occurrence patterns detection.

This work provides a new storage method for mining MDCOPs from large spatiotemporal attack data sets. The experimental results show that the proposed storage method and the pattern mining algorithms are correct, complete and efficient.

The framework of experimental design and the comparisons of existing approaches are provided in the paper.

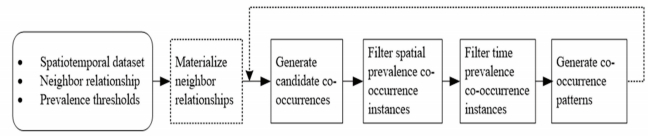


Fig. 1. Basic procedure of co-occurrence pattern mining

1.2. Scope and Outline

This work is aim to addresses the co-occurrence pattern mining for large volume of spatiotemporal attack data sets. Our system achieves superior performance of computation efficiency to existing work like fast MDCOP-Miner [3]. Determining thresholds for MDCOP interest measures and inserting object-types at arbitrary time intervals are not the scope of this paper.

The rest part of this paper is organized as follows: Section 2 reviews related works. Section 3 presents basic concepts to provide a formal model of MDCOP Graph and the problem statement of mining MDCOPs. In Section 4, we presented our proposed MDCOP mining algorithms. Analysis of the algorithms is given in Section 5. Section 6 presents the experimental evaluation and section 7 presents conclusions and future work.

2. Related Works

Previous studies for mining spatiotemporal co-occurrence patterns can be classified into two categories: the approaches of mining uniform groups on large attack event data sets and the approaches of mining mixed groups on large attack event data sets. MDCOP belongs to the latter. Audit Data Analysis and Mining is one of the known data mining projects in an intrusion detection, which uses a module to classify the abnormal event into false alarm or real attack proposed by Barbara′ et al. [4]. It is an online network-based ID, which used two data mining techniques, association rule and classification. Shi-Jinn Horng et al. [8] investigate using both a linear and a non-linear measure linear correlation coefficient and mutual information, for the feature selection. Jau Tsang et al. [19] proposed fuzzy rule-based system is evolved from an agent-based evolutionary framework and multi-objective optimization. Gudmundsson et al. [10] proposed an algorithm for detecting flock patterns in spatiotemporal datasets. Kalnis et al. [13] defined the problem of discovering moving clusters and proposed clustering-based methods to mine such patterns. If there are many common objects between clusters in consecutive time slots, such clusters are called moving clusters. The moving cluster patterns can be unified or a

mixed object group [14]. However if there are no overlaps between the clusters in consecutive time slots, their proposed algorithms for mining moving clusters will fail to discover MDCOPs.

The MDCOPs problem differs from the co-location pattern. Previous approaches of MDCOP mining can use a spatial co-location mining algorithm for each time slot to find spatial prevalent co-locations, and then apply a post-processing step to discover MDCOPs by check their time prevalence. To mine co-locations, Huang et al. firstly proposed the mining of co-location patterns(different subset of instances of object types are adjacent to each other) from spatial datasets, then they proposed a join-based approach [20]; Cao et al. proposed a co-location approach from some period in spatiotemporal datasets, which can find out the co-location instances from continuous time slots [2]; Yoo et al. proposed a partial join-based and a join-less approach [18][11][12] which used nearby star and instance methods. Celik et al. [14]formalizes the problem, propose a new monotonic mixed-drove interest measure to discover and mine MDCOPs, and also propose an efficient algorithm (MDCOP-Miner).

MDCOPs represent object types co-located over space and time forming a spatial network (edges between objects in the network indicate existence of a neighborhood relationship) that dynamically changes over time. A common and naive approach to model such a network is to use time expanded graph, as described by K*ö*hler et al. [6] where the network is replicated across discrete time instants. Ding et al. [5] proposed an alternative approach where attributes at each node and edge of the graph are used to model state and topology changes respectively. A more efficient method of modeling temporal spatial networks was proposed by George et al. [7], by incorporating the properties of nodes and edges in the graph as a time series. This paper also proposed efficient algorithms for computing the shortest path and connectivity in time dependent networks modeled using time aggregated graphs. The problem of mining MDCOPs with high spatial and time prevalence is described by Celik et al. [14]. However the approach is similar to Apriori like and involves candidate generation, which is costly as it enumerates all possible cliques over all time instances. A. Garaeva, M. Tang, Z. Wang, etc. [15][1] developed a framework implemented in Apache Spark environment for co-location patterns mining in big spatio-temporal data, which make evaluation of applied algorithms from the point of their efficiency and scalability.

Considerable achievements in MDCOP mining have been obtained in the last few years. However, dealing with large amount of spatiotemporal datasets is still challenging due to the accuracy and efficiency issues. In this paper, we materialize the neighbor relationships for efficient co-occurrence pattern mining, and solve the problem of efficient storage of MDCOPs by using the Time Aggregated Graph model and create our own storage model MDCOP Graph for mining MDCOPs. Finally, we provide an accurate and efficient co-occurrence pattern mining algorithm in large spatiotemporal datasets.

3. Co-occurrence Pattern Mining

In this section, we present basic concepts to provide a formal model of MDCOP Graph and the problem statement of mining MDCOPs. In the paper, we use multiple prevalence measures, which use threshold-spatial prevalence and temporal prevalence to determine

whether one pattern is co-occurrence pattern. Meanwhile, we use time aggregate graph as data storage structure.

3.1. Mixed-drove Prevalence Measure

The focus of this study is to mine MDCOPs with multiple prevalence measures from large spatiotemporal data sets. Co-occurrence pattern represents subsets of different object-types whose instance are co-located together for a significant fraction time. Mining co-occurrence patterns from spatiotemporal datasets is based on the mining of spatial co-location patterns from spatial datasets. Co-location pattern is the dataset of the frequent and closely adjacent spatial characteristics in the geographical space. Spatiotemporal co-occurrence patterns are the co-location patterns which satisfy the time prevalence measure.

The basic MDCOP algorithm [14] defines two interest measures namely spatial prevalence θ_p and a time prevalence measure θ_{time} .

The spatial prevalence measure is used to determine if the pattern is spatially prevalent in a specific time slot. The time prevalence measure is used to determine if the pattern is frequent in all time slots. These threshold values are mostly domain-specific. It is difficult to determine suitable interest measure thresholds to identify MDCOPs without domain knowledge. If the values of thresholds are too small, there are many unnecessary patterns will be generated. Otherwise, there are too few patterns. It's possible to miss some significant candidate MDCOPs if they are too large.

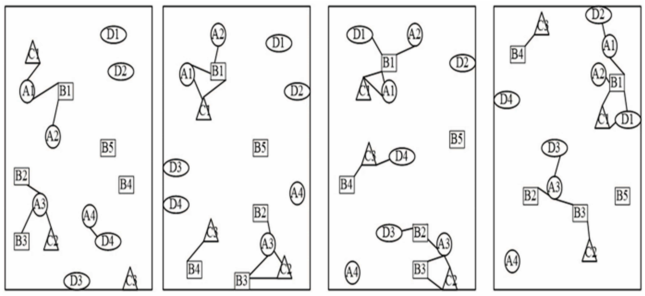
If a candidate pattern satisfies the following inequality, we can take this pattern as a spatial co-location pattern.

$$Prob_{t_m \epsilon TF}(s\_prev(P_i, time\_slot \quad t_m)) \geq \theta_p \tag{1}$$

Where, $Prob(.)$ is the probability of overall prevalence time slots; s_prev stands for spatial prevalence measure; θ_p is the spatial prevalence measure. Hence, a pattern is defined as an MDCOP if it satisfies the following inequality.

$$Prob_{t_m \epsilon TF}[s\_prev(P_i, time\_slot \quad t_m)) \geq \theta_p] \geq \theta_{time} \tag{2}$$

Where, $Prob(.)$ is the probability of overall prevalence time slots; θ_p stands for spatial prevalence measure; θ_{time} is the time prevalence measure. For example, in Fig. 2a, AB is an MDCOP because it is spatial prevalent in time slots 0, 1, 2, and 3 since its participation indices are no less than the given threshold 0.4 in these time slots, and is time prevalent since its time prevalence index of 1 is above the threshold 0.5. In contrast, BD is not a MDCOP. Although it is spatial prevalent in time slot 2, it is not time prevalent since its time prevalence index is no more than the given time prevalence threshold 0.5.



(a)

Mixed-Drove Co-occurrence Patterns	Spatial prevalence index values				Time prevalence index values
	time slot()	time slot1	time slot2	time slot3	
AB	3/5	3/5	3/5	3/5	4/4
AC	2/4	2/4	2/4	0	3/4
BC	0	3/5	3/5	3/5	3/4
ABC	0	2/5	2/5	0	2/4

(b)

Fig. 2. (a) An input spatiotemporal data set. (b) A set of output MDCOPs

Time Aggregated Graph (TAG) Spatiotemporal network is a spatial network whose topological relations and attribute information change with the passage of time. Spatiotemporal network model has many applications, such as real-time traffic planning, optimal shortest path selection and so on. These applications need to create a spatiotemporal network model which can be queried and calculated efficiently.Kohler et al. proposes a spatiotemporal network model-time expanded graph. Spatial network for each time slot is established in this model, which results in a large duplication of node information. With the increase of time slots, the time-consuming of establishing time expanded graph also significantly increased. To solve this problem, George et al. proposes a more efficient method-time aggregate graph, which takes time series as the attribute of node and edge. The duplication problem is solved by using this method, while it is difficult to efficiently obtain the location relationships between target object and other objects in all time slots. Currently, modeling spatiotemporal objects by time aggregate graph is an ideal spatiotemporal network model, which has low storage consumption, high query efficiency.

To take the advantage of time aggregate graph, we propose a graph based data structure to capture the information required to mine MDCOPs from the spatio-temoral data set. This data structure is motivated by Time Aggregated Graphs (TAG) [7] which models time varying road conditions as time series on the edges of a road network.[7] defines the time aggregated graph as follows.

$$TAG = (N, E, TF, f_1 ... f_k, g_1 ... g_m, w_1 ... w_p | f_i : N \rightarrow R^{TF}; \\ g_i : E \rightarrow R^{TF}; w_i : E \rightarrow R^{TF}) \tag{3}$$

Where N is the set of nodes, E is the set of edges, TF is the length of the entire time interval, $f_1...f_k$ are the mappings from nodes to nodes, $g_1...g_m$ are mapping from edges to edges, and $w_1...w_p$ indicate the dependent weights (eg.travel times) on the edges.

Each edge has an attribute, called an edge time series that represents the time instants for which the edge is present. This enables TAG to model the topological changes of the network with time.

Fig. 3a, 3b, 3c shows a network at three time instants. The network topology and parameters change over time. For example, the edge N2-N1 is present at time slot 0, time slot 1 and disappears at time slot 2, and its weight changes from 1 at time slot 0 to 5 at time slot 1. The time aggregated graph that represents this dynamic network is shown in Fig. 3d. In this figure, edge N2-N1 has two attributes, each being a series. The attribute (0, 1) represents the time instants at which the edge is present and [1, 1, -] is the weight time series, indicating the weights at various instants of time.

TAG models the network which has topology change with time passage, it can stores and combines spatial and temporal information efficiently. Thus it can save a lot of memory space and easily get all approaching slot information when querying examples of spatial relationships. Using the above-mentioned advantages of TAG model, we expand on this model and propose spatiotemporal co-occurrence storage model MDCOP Graph.

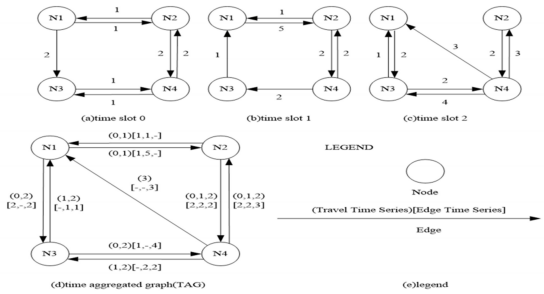


Fig. 3. (a), (b), (c) Network at various time instants. (d) Time Aggregated Graph (TAG). (e) Legend

3.2. Modeling MDCOP Graph

In this paper, we propose MDCOP Graph which based on TAG (Time Aggregated Graph) network model to store instances of close relationships, and obtain information for the process of mining co-occurrence patterns through visiting MDCOP Graph. Thus it can avoid a large number of join operations and complex space region split, which can improve the computational efficiency.

Given a set of spatiotemporal mixed object-types E, a neighborhood relation R, a set of time slots TF, a threshold pair (θ_p, θ_{time}), MDCOP Graph can be represented as a neighbor graph in which a node is an object type and edge between two nodes represents the neighbor relationship over all time slots. We use MDCOP Graph to materialize neighbor

relationships. As we know that most of the computational time of co-occurrence pattern mining is devoted to finding co-occurrence instances. By means of MDCOP Graph, we don′ t need to generate all possible candidate co-occurrence instances. We just generate real co-occurrence instances through visiting the MDCOP Graph. Thus, we can increase the efficiency of co-occurrence mining.

Definition 3.1 Given a set of co-occurrence instances CI, instance type level graph (IG) is used to captures the existence of co-location instances between two instance types over time. We define instance type level graph as follows.

$$IG = (I, CI, TF, f_0...f_{k-1}, e_0...e_{n-1}, |f_i : I \rightarrow R^{TF}; e_i : CI \rightarrow R^{TF}) \quad (4)$$

Where I is the set of instances of all object-types, CI is the set of co-location instances, TF is the length of the entire time interval, such that TF = [T_0 ,...,T_{n-1}], $f_0...f_{k-1}$ are the mappings from object-types to object-types , $e_0...e_{n-1}$ indicate the existence of co-occurrence instances between two instance types over time on the edges.

For example, we generate instance type level graph (Fig.4) using the data set given in Fig. 2a. In Fig.4, we use time-series [1 1 1 0] to show that A1 and C1 are co-located at time slot 0, time slot 1, time slot 2 and disappear at time slot 3. Therefore, we can easily capture the existence of co-occurrence instances over time by traversing instance type level graph.

Definition 3.2 Given a set of candidate co-occurrence patterns (CP), object type level graph (OG) is used to indicate the participation count of particular object-types contributing to particular co-occurrence patterns. We define object type level graph as follows.

$$OG = (E, CP, TF, f_0...f_{k-1}, p_0...p_{n-1}, |f_i : E \epsilon CP^{TF}; p_i : E \epsilon CP^{TF}) \quad (5)$$

Where E is the set of spatiotemporal mixed object-types, CP is the set of co-occurrence patterns, TF is the length of the entire time interval, $f_0...f_{k-1}$ are the mappings from particular object-types to the particular co-occurrence patterns , $p_0...p_{n-1}$ indicate the participation count of particular object-types contributing to particular co-occurrence patterns over time on the edges.

For example, we generate object type level graph (Fig.5) using the data set given in Fig. 2a. In Fig.2a, the co-occurrence pattern AC has co-occurrence instances sets A1, C1, A3, C2 at time slot 0, time slot 1 and time slot 2. At time slot 3, AC has no co-location instances. In Fig.5, since A1 and A3 are different instances of A, we used time-series [2 2 2 0] to show the participation count of object-type A contributing to co-location pattern AC. By using object type level graph, we can get the spatial prevalence index values and time prevalence index values.

Definition 3.3 Given instance type level graph and object type level graph, MDCOP Graph is composed of two parts: instance type level graph and object type graph. The instance type level and object type graphs are connected through links. We define MDCOP Graph as follows.

$$MDCOPGragh = (IG, OG, TF, E, I, l_0...l_{k-1}, |l_i : I \rightarrow E^{TF}) \quad (6)$$

Where IG is instance type level graph, OG is object type level graph. TF is the length of the entire time interval, E is the set of spatiotemporal mixed object-types, I is the set of instances of all object-types. $l_0 ... l_{k-1}$ are the mappings from object-types to their own instances.

For example, we generate MDCOP graph (Fig.6) using the data set given in Fig. 2a. In Fig.6, both the instance type level and object type graphs are connected through links for easy traversal.

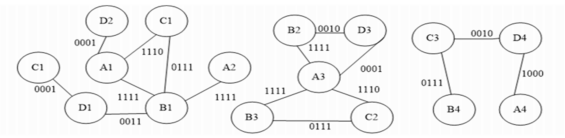


Fig. 4. Instance type level graph

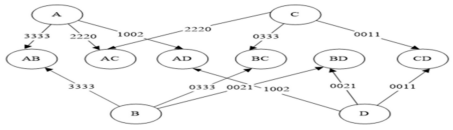


Fig. 5. Object type level graph

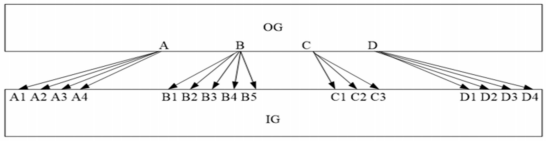


Fig. 6. Object type level graph

3.3. Problem Statement

Given:

A set E of spatiotemporal object types over a common spatiotemporal framework.
A neighbor relation R over locations.
A multiple set of user defined spatial and temporal prevalence thresholds.
Find:

Minimize cost of generate the MDCOP graph.

A set of frequent MDCOPs whose mixed-drove prevalence satisfies spatial and temporal prevalence thresholds.

Objective:

Capture and represent the temporal variation of MDCOPs.

Minimize cost of candidate generation.

Minimize cost of generate the MDCOP graph.

Constraints:

To build the MDCOP graph that stores all the MDCOPs.

To find the correct and complete set of MDCOPs for any threshold value specified.

4. Mining MDCOPS

In this section, we discuss fastMDCOP-Miner and then propose two novel MDCOP mining algorithms: MDCOP Graph Miner and LDMDCOP Graph Miner to mine MDCOPs. We also give the execution trace of these algorithms.

4.1. Fast MDCOP-Miner

FastMDCOP-Miner [3] uses a spatial co-location mining algorithm for each time slots to find spatial prevalent co-locations and prune time non-prevalent patterns as early as possible between the time slots to discover MDCOPs. To mine co-locations, Huang et al. proposed a join-based approach, Yoo et al. proposed a partial join-based approach and a join-less approach [16][18][11], this approach is based on the join-based collocation algorithm proposed by Huang et al., but it is also possible to use other approaches. Fast MDCOP-Miner [3] will first discover all size k spatial prevalent MDCOPs and prune time non-prevalent patterns as early as possible between the time slots to discover MDCOPs. Then the algorithm will generate size k +1 candidate MDCOPs using size k MDCOPs until there are no more candidates. However, this approach is Apriori like and involves candidate generation which is costly as it enumerates all possible cliques over all time instants.

4.2. MDCOP Graph Miner

To eliminate the drawbacks of fast MDCOP-Miner, we propose a MDCOP mining algorithm (MDCOP Graph Miner) to discover MDCOPs by storing all the MDCOPs in the MDCOP Graph.

This data structure is motivated by Time Aggregated Graphs (TAG) [7], which models time varying road conditions as time series on the edges of a road network. In our case, we use two different types of series over the edges. One of the series captures the existence of co-occurrence patterns between two instances over time. Based on the existence time series of co-occurrence patterns between pairs of instances, we aggregate the information to object types. At the object type graph, each time series contains the participation count of a particular object contributing to a particular co-location pattern. Both the instance type level and object type graphs are connected through links for efficient traversal.

Firstly, MDCOP Graph Miner will establish spatial temporal network MDCOP Graph, initialize the spatial relationship between each time slot; then it generates k+1 candidate spatiotemporal co-occurrence patterns; It obtains instance sets of candidate spatiotemporal co-occurrence patterns through querying MDCOP Graph and calculates time and space frequent degrees, generate k+1 spatiotemporal co-occurrence patterns; Iteratively, it repeatedly generating candidate co-occurrence patterns until there is no new candidate co-occurrence pattern and algorithm terminate. Eventually, we get all the co-occurrence patterns meeting temporal and spatial thresholds.

We give the pseudo code of the algorithm and provide an execution trace of it using the data set in Fig. 2a. Algorithm1 shows the pseudo code of the MDCOP Graph Miner algorithm. This pseudo code is used to explain two algorithms: MDCOP Graph Miner and LDMDCOP Graph Miner which will be discussed in the next section. The choice of the algorithm is provided by the user. In the algorithm, steps 1-14 create the MDCOP Graph. Steps 15-21 give an iterative process to mine MDCOPs, steps 15-21 continue until there is no candidate MDCOP to be generated. Step 22 gives a union of the results. The execution traces of MDCOP Graph Miner are explained below.

```
Algorithm 1 pseudo code for the MDCOP Graph Miner
Inputs:
E: a set of spatial object types
ST: a spatiotemporal data set < object_type, object_id,
 x, y, timeslot >
R: spatial neighborhood relationship
TF: a time slot frame t_0,...,t_{n-1}
θ p:a spatial prevalence
θ time: a time prevalence threshold
Output: MDCOPs whose spatial prevalence indices, i.e.,
participation indices, are no less than θ p,
for time prevalence indices are no less than θ time.
Variables:
t : time slots(t_0,...,t_{n-1})
k:co-occurrences size
Tk: set of instances of size k co-occurrences
SPk: set of spatial prevalent size k co-occurrences
TPk: set of time prevalent size k co-occurrences
Ck: set of candidate size k co-occurrences
MDPk: set of mixed-drove size k co-occurrences
MDG: graph stores all the MDCOPs
Address: address for storing time series to the file
Method:
   k = 2
   Ck (0) = gen_candidate_co_occ(E)
   for each time slot t in TF
       Tk (t) = gen_co_occ_inst(Ck (t),ST, R)
       set timeSeries [t] =1 for Tk (t)
       SPk (t) = find_spatial_prev_co_occ(Tk (t), θ p)
       TPk (t) = find_time_index(SPk (t))
      MDPk (t) = find_time_prev_co_occ(TPk (t), θ  time)
       Ck (t) = MDPk (t)
 if(alg_choice  "LDMDCOP Graph Miner")
        Address = gen_co_occ_address (Tk (t) )
       access the MDG file by the addresses
       set timeSeries [t] =1 for Tk (t) if required
   if(alg_choice == "MDCOP Graph Miner")
       MDG = gen_MDCOP_Graph(MDPk)
```

```
while (not empty MDPk)
   Ck +1 = gen _candidate_co_occ ( MDPk)
    Tk +1= gen _instancesTree (Ck +1,MDG)
   SPk+1 = find_spatial_prev_co_occ(Tk+1, θ p)
   TPk +1 = find_time_index(SPk+1 )
    MDPk+1 = find_time_prev_co_occ(TPk+1, θ time)
       k = k+1
 return union( MDP2 ,..., MDPk)
```

The execution trace of the MDCOP Graph Miner is given in Fig.7. This data set in Fig.2a. Contains four object-types A, B, C, and D and their instances in four time slots (i.e., A has four instances). The instances of each object-type have a unique identifier, such as A1. To discover MDCOPs, we use a monotonic composite interest measure which is a combination of the spatial prevalence and time prevalence measure. The spatial prevalence measure shows the strength of the spatial co-location when the index is greater than or equal to a given threshold [3][18].The time prevalence measure shows the frequency of the pattern over time.

In step1, by dividing each entry in Fig. 7a with the corresponding number of instances for an object, we get the participation ratio of an object type in co-location. For example, the participation index of collocation AB is [3/5 3/5 3/5 3/5], which is the minimum participation ratio of type A and B in all time slots. We prune time non-prevalent patterns whose participation indices are less than a given threshold as early as possible. For example, there are four time slots and the time prevalence threshold is 0.5.In this case, a size k pattern should be present for at least two time slots to satisfy the threshold. If the time prevalence index of a pattern is 0 for the first (or any) three time slots, there is no need to generate it and check its prevalence for the rest of the time slots even if it is time persistent for the remaining time slots. Spatial prevalent patterns AB, AC, and BC are selected as MDCOPs since they are time prevalent (their time prevalence indices satisfy the given time prevalence threshold 0.5). In contrast, spatial prevalent patterns AD, BD and CD are pruned since they are time non-prevalent.

In step2, three sub-graphs on the bottom of Fig. 7b are created. It also creates links from the instance types to the object type, for example, Link between A3 and A if A3 is part of at least one co-occurrence. The links between the object and the instance type help in traversing the data set efficiently to calculate the spatial prevalence index values. Connections between the object type graph and the instance type graph are missing to reduce clutter and the series in the object graph has not been represented for the same reason. After the algorithm has been executed, the series on edge A and AB would be [3/4 3/4 3/4 3/4] because of co-locations A1B1, A2B1, A3B2 andA3B3. Note that A3 is counted only once at each time interval though it appears in two co-locations at every time instant.

In step3, the candidate MDCOP ABC is generated through AB, AC and BC. Generally, the number of candidate patterns is large. Then if we generate temporal time prevalence index for every candidate pattern, candidate pattern whose temporal time prevalence index is less than a given threshold can be pruned. For example, ABC is a candidate pattern, the temporal time series of ABC is [0 1 1 0] equals to time series of AB [1 1 1 1] & AC [1 1 1 0] & BC [0 1 1 1], the time prevalence index is 0.5 which is no less than the given threshold. Thus, we generate instances of candidate pattern ABC. By modeling

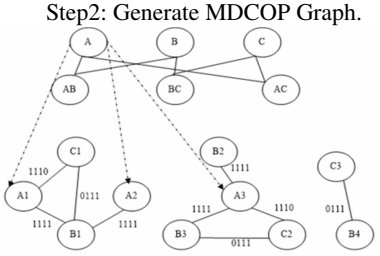
the instances-trees for candidate patterns, the instances of candidate pattern ABC can be generated.

Step1: Generate size 2 co-occurrence patterns.

Object Type	Candidate MDCOP	Paticipation Count	Paticipation Rations	Paticipation Index	Time Series	Time prevalence index
A	AB	3 3 3 3	3/4 3/4 3/4 3/4	3/5 3/5 3/5 3/5	1 1 1 1	4/4
B	AB	3 3 3 3	3/5 3/5 3/5 3/5			
A	AC	2 2 2 0	2/4 2/4 2/4 0	2/4 2/4 2/4 0	1 1 1 0	3/4
C	AC	2 2 2 0	2/3 2/3 2/3 0			
A	AD	1 0 0 2	1/4 0 0 2/4	1/4 0 0 2/4	0 0 0 1	1/4(prune)
D	AD	1 0 0 2	1/4 0 0 2/4			
B	BC	0 3 3 3	0 3/5 3/5 3/5	0 3/5 3/5 3/5	0 1 1 1	3/4
C	BC	0 3 3 3	0 3/3 3/3 3/3			
B	BD	0 0 2 1	0 0 2/5 1/5	0 0 2/5 1/5	0 0 1 0	1/4(prune)
D	BD	0 0 2 1	0 0 2/4 1/4			
C	CD	0 0 1 -	0 0 1/3 -	0 0 1/4 -	0 0 0 -	(prune)
D	CD	0 0 1 -	0 0 1/4 -			

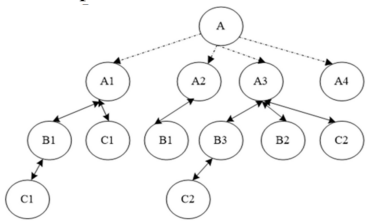
(a)

Step2: Generate MDCOP Graph.



(b)

Step3: Generate instances-tree.



(c)

Step 4: Generate size 3 co-occurrence patterns

Object Type	Candidate MDCOP	Paticipation Count	Paticipation Rations	Paticipation Index	Time Series	Time prevalence index
A	ABC	0 2 2 0	0 2/4 2/4 0	0 2/5 2/5 0	0 1 1	2/4
B	ABC	0 2 2 0	0 2/5 2/5 0			
C	ABC	0 2 2 0	0 2/3 2/3 0		0	

(d)

Fig. 7. Execution trace of the MDCOP Graph Miner algorithm. (a) Step 1. (b) Steps 2. (c) Steps 3. (d) Steps 4

In step4, the participation indices of pattern ABC are 2/5 in time slots 1 and 2 and its time prevalence index 0.5 equals to the threshold. Since there are not enough subsets to generate the next superset patterns, the algorithm stops at this stage and outputs the union of all size MDCOPs, i.e., A B, AC, BC, and ABC.

4.3. LDMDCOP Graph Miner

Most of existing spatiotemporal co-occurrence patterns mining algorithms are only work on memory. Due to the advance of geo-sensor technology, spatiotemporal data increased dramatically. The demand of mining spatiotemporal co-occurrence patterns in large datasets is now gradually urgent.

In practical applications, such as now have 300MB armed vehicles date, when a vehicle moves, its co-occurrence pattern also present movement in a similar trajectory. By mining spatiotemporal co-occurrence patterns of moving vehicle data, we can infer military strategy through co-occurrence patterns. As another example, the public security bureau of Zhejiang province has 2GB of crime data, we can identify many interesting co-occurrence patterns from the data, such as fraud and gambling is a co-occurrence pattern, then the police can infer the offender's whereabouts. However, in the above practical application, the amount of spatiotemporal data is huge, existing mining algorithms cannot obtain useful co-occurrence patterns efficiently.

For solving the above problems, we use MOCOP Graph to store the data sets of instances; the instance set is stored in the hard disk file. In the calculation process, we only load the necessary data to memory in order to save the space. Meanwhile, we establish index for the instances in document to improve data query efficiency.

In this section, we propose a new algorithm, called LDMDCOP Graph Miner, which can handle large data sets by using file index. MDCOP Graph is an efficient storage method, which can capture the information required to mine MDCOPs from the data sets. When the volume of the data sets is large, it consumes a lot of space to store MDCOP Graph. Since the capacity of memory is limited, there may be no enough space to store large amount of data sets or big MDCOP Graph. As a result, we use file index to store big MDCOP Graph. This approach will still have two problems. One is how to store the big MDCOP Graph in the file; the other is how to capture the information required for mining MDCOPs from the MDCOP Graph in the file.

In order to solve these problems, we use adjacency matrix to store the MDCOP Graph. The advantage of adjacency matrix is the ability to determine the existence of a particular edge in constant time, and access the storage media only once. According to this method, we can calculate the address for a particular edge to store its time series in the file and also access the time series by the same address, there is no need to store the address of the time-series, we calculate the address according to the same expression which used to calculate the address for storing. The expression is as follows.

$$address(R_i, C_j) = R_i \times N + C_j - E(R_i, C_j) \tag{7}$$

Where R_i is the row number, C_j is the column number, N is the total number of instance, E (R_i , C_j) is the number of patterns whose instances are of the same type.

Physical address can be obtained according to this formula. So long as we can get the required time sequence through visiting the file only once. By indexing methods to

achieve the storage of MDCOP Graph in file, while efficiently obtaining the required information.

Firstly, LDMDCOP Graph Miner will establish spatial temporal network MDCOP Graph, initialize the spatial relationship between each time slot; then generate k+1 candidate spatiotemporal co-occurrence patterns, calculate Pi and TPi, prune the patterns which don't meet spatial and temporal thresholds, then generate k+1 spatiotemporal co-occurrence patterns; It repeats the above operate until there is no new candidate co-occurrence pattern; Eventually we get all the co-occurrence patterns meeting temporal and spatial thresholds.

The difference between LDMDCOP Graph Miner and MDCOP Graph Miner is that MDCOP Graph Miner has the different storage structure and storage media. LDMDCOP Graph Miner storage MDCOP Graph as adjacency matrix in document, while MDCOP Graph Miner storage MDCOP Graph as adjacent table in memory. Other steps are basically the same.

The pseudo code of the LDMDCOP Graph Miner is given in Algorithm 1. When the LDMDCOP Graph Miner is chosen, the algorithm will activate steps 10, 11, 12 and deactivate steps 13 and 14. We use adjacency matrix to store the MDCOP Graph. At the same time, the addresses for storing time series of co-occurrence instances could be calculated and the addresses also are used to access the file for getting the information required to mine MDCOPs.

The execution trace of gen_MDCOP_Graph in LDMDCOP Graph Miner is explained as algorithm 2.

```
Algorithm 2 pseudo for gen_MDCOP_ Graph in
 LDMDCOP Graph Miner
    Inputs:
ST: a spatiotemporal data set < object_type, object_id,
x, y, timeslot >
TF: a time slot frame t0,...,tn−1
t:time slot number
MDPk: set of mixed-drove size k co-occurrences
Output:
MDG File: MDCOP Graph file
Variables:
Address:index table
eofAdress:end address of the current file
logAddress:logical address
phyAddress:physical address
Method:
BEGIN
  FOR each co-occurrence patterns Pi of MDP2(t)
                FOR each co-located instances (Ai,Bj)
 of Pi
                     logAddress = gen_Address ( Ai , Bj )
                  IF (Address[logAddress] == 0)
                 Address[logAddress] = eofAddress
                        eofAddress += TF
                    END IF
                        phyAddress = Address[logAddress]
                    set timeSeries[t] = 1 for AiBj
                      update timeSeries in MDG File
       END FOR
```

```
  END FOR
END
```

The method to store MDCOP Graph in LDMDCOP Graph Miner is described as algorithm 2. It stores the nearest relationships between instances of co-occurrence patterns in each timeslot. It generates the logical address logAddress between time series of instances through gen_Address and lookup index table. Then, it obtain physical address phyAddress according to logical address logAddress; finally the time series of instances are instored in the files.

As described in algorithm 2, for each co-location pattern of each co-occurrence pattern, firstly generate its logical address (algorithm 3); if this co-location pattern doesn't have physical address, then generate physical address; update the end address of current file; once this co-location pattern has physical address, then obtain its physical address through index table and update its time series; finally store the updated time series in MDG file.

The difference between LDMDCOP Graph Miner and MDCOP Miner is the storage structure for MDCOP Graph. This chapter below will focus on the description of the process storing MDCOP Graph by adjacency matrix and index storage, as to mine spatial-temporal co-occurrence patterns from data sets in figure 2. (a).

According to table 1, we can create index for the adjacency matrix, which has been shown in table 2. The logical address is generated by the former formula, while physical address is the storage address of time series. We can efficiently obtain physical address by indexing table, so as to achieve quick indexing for required information.

```
Algorithm 3 pseudo for gen_Address
Inputs:
ST: a spatiotemporal data set < object_type, object_id,
 x, y, timeslot >
TF: a time slot frame t_0,...,t_{n-1}
type1:type1
type2:type2
ins1:instance 1
ins2:instance 2
Output:
Address: storage address between instance 1 and 2
Variables:
N:types of different instances
typeIns:instance set of each type
typeInsNum: instance number of each type
insNum:instance number
deleteIns:the number of instances of patterns
with same type
BEGIN
WHILE(!ST.eof())
  typeIns[typeId].push(insId)
END WHILE
FOR each type i in E
                N += typeIns[i].size()
END FOR
FOR each type i in E
                FOR each instance j of i
                        IF( i== 0)
                         insNum[i][j] = j
```

```
         ELSE
                                insNum[i][j] = insNum [i-1]
               [ typeIns[i-1].size() -1] +j+1
         END IF
                END FOR
END FOR
FOR each type i in E
                FOR each instance j of i
                        deleteIns[i][j] += insNum[i+1][0]
     END FOR
END FOR
Address = insNum[type1][ins1] * N
+ insNum[type2][ins2]- deleteIns[type1][ ins1]
END
```

Table 1. MDCOP Graph adjacency matrix

	B1	B2	B3	B4	B5	C1	C2	C3	D1	D2	D3	D4
A1	1111	0000	0000	0000	0000	1110	0000	0000	0000	0000	0000	0000
A2	1111	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
A3	0000	1111	1111	0000	0000	0000	1110	0000	0000	0000	0000	0000
A4	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
B1	×	×	×	×	×	0111	0000	0000	0000	0000	0000	0000
B2	×	×	×	×	×	0000	0000	0000	0000	0000	0000	0000
B3	×	×	×	×	×	0000	0111	0000	0000	0000	0000	0000
B4	×	×	×	×	×	0000	0000	0111	0000	0000	0000	0000
B5	×	×	×	×	×	0000	0000	0000	0000	0000	0000	0000
C1	×	×	×	×	×	×	×	×	0000	0000	0000	0000
C2	×	×	×	×	×	×	×	×	0000	0000	0000	0000
C3	×	×	×	×	×	×	×	×	0000	0000	0000	0000

Table 2. Index Table

Instances	A1B1	A1C1	A2B1	A3B2	A3B3	A3C2	B1C1	B3C2	B4C3
Logical Address	0	5	12	25	26	30	48	63	71
Physical Address	0	4	8	12	16	20	24	28	32

5. Analytical Evaluation

This section gives the analysis of correctness and completeness for the MDCOP mining algorithms.

5.1. Correctness

LEMMA 1: MDCOP Graph Miner and LDMDCOP Graph Miner are complete.

Proof: The MDCOP Graph Miner and LDMDCOP Graph Miner are complete if they find all MDCOPs that satisfy a given participation index threshold and time prevalence threshold. We can show this by proving that none of the functions in the algorithm miss any patterns.

The $gen\_candidate\_co\_occ$ function does not miss any patterns. The input of this function is size k MDCOPs, and the output is candidate size $k+1$ MDCOPs. The gen_InstancesTree function does not miss any patterns. This function generates instances of candidate size $k+1$ MDCOPs if they are in the neighborhood distance and forming a clique.We can prove this with the completeness of the Breadth First Search algorithm which enumerates all connected edges of a given instance level sub-graph. Further, the links from the object type nodes to the instance level nodes are visited. This guarantees that no instance level sub-graph is missed by the algorithm. Further, using the time series on the instance level sub-graphs, the participation ratios of object types in their respective co-occurrences are updated for every object instance visited by the BFS.

5.2. Completeness

LEMMA 2: MDCOP Graph Miner and LDMDCOP Graph Miner are correct.

Proof: MDCOP graph is correct if the participation indices of each co-occurrence is calculated according to [17][9][14]for all time instances. First, all edges in the instance level sub-graph are traversed only once by the BFS and hence are accounted only once for the participation index of the co-location. Second, if there is a case where an instance B_i co-occurrences with A_j and A_k where A_j and A_k are of the same object type A, then the existence series of B in AB has to be considered only once for calculating participation ratio of B in AB. We use these participation indices weed out candidates not meeting the given thresholds.

Hence, the MDCOP Graph Miner and LDMDCOP Graph Miner are correct.

5.3. Complexity

LEMMA 3: MDCOP Graph Miner Graph Miner is more effective than fastMDCOP-Miner.

Proof: For fastMDCOP-Miner, we just assume that T_f is the cost time of fastMDCOP-Miner. Just as shown in the following formula.

$$T_f = T_{neighbor\_pairs}(ST) + T_f(2) + \sum_{k>2} T_f(k) \tag{8}$$

Where ST is spatiotemporal data set, $T_f(2)$ is the time for calculating co-occurrence patterns under k=2; $T_f(k)$ is the time for calculating these patterns when k >2, we can calculate $T_f(k)$ by the following formula.

$$\begin{aligned} T_f(k) &= T_{gen\_candidate}(P_{k-1}) + T_{gen\_co\_occ\_inst}(C_k) \\ &\quad + T_{find\_spatial\_prev}(T_k) + T_{find\_spatial\_prev}(SP_k) \\ &\approx T_{gen\_co\_occ\_inst}(C_k) \end{aligned} \tag{9}$$

Where P_{k-1} represents co-occurrence patterns under $(k-1)$ size; C_k represents candidate co-occurrence patterns under size; T_k represents candidate co-occurrences pattern instances under k size; SP_k indicates co-location patterns under k size.

For MDCOP Graph Miner, as the same, we assume T_g indicates the total cost time. T_g can be calculated by the following formula.

$$T_g = T_{MDCOP\_Graph}(ST) + T_g(2) + \sum_{k>2} T_g(k) \tag{10}$$

Where ST is spatiotemporal data set, $T_g(2)$ is the time for calculating co-occurrence patterns under k=2; $T_g(k)$ is the time for calculating these patterns when k>2.As the same, $T_g(k)$ just can be calculated by this formula.

$$\begin{aligned} T_g(k) &= T_{gen\_candidate}(P_{k-1}) + T_{gen\_instanceTree}(C_k) \\ &\quad + T_{find\_spatial\_prev}(T_k) + T_{find\_spatial\_prev}(SP_k) \\ &\quad \approx T_{gen\_instancesTree}(C_k) \end{aligned} \tag{11}$$

P_{k-1} represents co-occurrence patterns under $(k-1)$ size; C_k represents candidate co-occurrence patterns under k size; T_k represents candidate co-occurrences pattern instances under k size; SP_k indicates co-location patterns under k size.

To compare the total cost time advantage between fastMDCOP-Miner and MDCOP Graph Miner, firstly, we can aim at the cost time for initializing the neighboring relationships and generating co-occurrence patterns when k=2. Obviously, MDCOP Graph Miner is more costly than fastMDCOP-Miner as the former needs more extra time to generate MDCOP Graph. The results appear as shown below.

$$T_{neighbor\_pairs}(ST) + T_f(2) < T_{MDCOP\_Graph}(ST) + T_g(2) \tag{12}$$

It's worth nothing that Tgen_candidate, Tfind_spatial_prev and Tfin_time_prev are equal in formulas Tf(k) and Tg(k). And these three costs just can be ignored comparing to Tgen_co_occ_inst.

Then, the comparison of the cost for generating co-occurrence patterns is under k>2. The ratio is as follows.

$$\frac{T_f(k)}{T_g(k)} \approx \frac{T_{gen\_co\_occ\_inst}(C_k)}{T_{gen\_instancesTree}(C_k)} \approx \frac{|C_k| \times t_join}{|C_k| \times t_scan} \approx \frac{t_join}{t_scan} \tag{13}$$

Where t_{join} indicates the cost of join operation to generate candidate co-occurrence patterns; t_{scan} indicates the cost of generating candidate co-occurrence patterns through searching MDCOP Graph.

If the spatial temporal data sets we input is intensive, $t_{scan} \ll t_{join}$. And with the growth of spatial frequency threshold or temporal frequency threshold, this ratio tends to be bigger, there are more spatialtemporal co-occurrence patterns required to be generated.

Through the above analysis, assuming the average count of instances in each time slot is N, the total number of co-location patterns under k=2 is Nco-lo; average number of co-occurrence patterns is Nins; the maximum length of candidate co-occurrence patterns is I; TF represents the total number of time slots.

For fast MDCOP-Miner, the time complexity of $Tneibor\_pairs(ST)$ is $O(N * N * TF)$; and time complexity respectively of $T_f(2)$ and $T_f(k)$ are $O(Nco-lo)$

(which can be ignored) and $O(NinsI)$. Therefore, the total time complexity of fast-MDCOP-Miner is [$O(N*N*TF)$+ $O(Nco-lo)$+$O(NinsI)$].

For MDCOP Graph Miner, the time complexity of $TMDCOP\_Graph(ST)$ is $O(N*N*TF+Nco-lo)$. And time complexity of $T_g(2)$ and $T_g(k)$ respectively are $O(Nco-lo)$ and $O(NinsI-1)$. The total time complexity of MDCOP Graph Miner is [$O(N*N*TF+Nco-lo)$+$O(Nco-lo)$+$O(NinsI-1)$].

By comparing these two total time complexity, we can come to a conclusion that the latter is smaller than the former. In other word, MDCOP Graph Miner is more effective than fastMDCOP-Miner.

As shown in the methods, we can get conclusion that LDMDCOP Graph Miner Graph Miner is effective in mining MDCOP patterns under big data sets. For this miner method, let T_l represents the total time LDMPCOP Graph Miner costs. So T_l can be described by this formula.

$$T_l = T_{MDCOP\_Graph}(ST) + T_l(2) + \sum_{k>2} T_l(k) \tag{14}$$

Where ST indicates the data set; $T_l(2)$ is the cost time for calculating spatialtemporal co-occurrence patterns when k=2; $T_l(k)$ just represents the total time for generating co-occurrence patterns when $k > 2$.

$$\begin{aligned} T_l(k) &= T_{gen\_candidate}(P_{k-1}) + T_{gen\_co\_occ\_inst}(C_k) \\ &\quad + T_{find\_spatial\_prev}(T_k) + T_{find\_time\_prev}(SP_k) \\ &\approx T_{gen\_co\_occ\_inst}(C_k) \end{aligned} \tag{15}$$

Also, P_{k-1} represents co-occurrence patterns under $(k-1)$ size; C_k represents candidate co-occurrence patterns under k size; T_k represents candidate co-occurrences pattern instances under k size; SP_k indicates co-location patterns under k size. And comparing to $T_{gen\_co\_occ\_inst}$, we can just ignore $T_{gen\_candidate}$, $T_{find\_spatial\_prev}$ and $T_{fin\_time\_prev}$.

Assume that hard disk I/O time-consuming is M-fold to memory I/O time-consuming. The time complexity of TMDCOP_Graph(ST) is $O(N*N*M*TF)$; for $T_l(2)$ is $O(Nco-lo)$; for $T_l(k)$ is $O(Ninsk-l*M)$. The total time complexity for LDMD-COP Graph Miner is $O(N*N*M*TF)$+ $O(Nco-lo)$+ $O(Ninsk-l*M)$.

LDMDCOP Graph Miner can effectively mine spatialtemporal co-occurrence patterns from big spatial and temporal data sets, which can't be reached by MDCOP Graph Miner and na?ve method. Meanwhile, we can get correct and complete sets of spatial-temporal co-occurrence patterns.

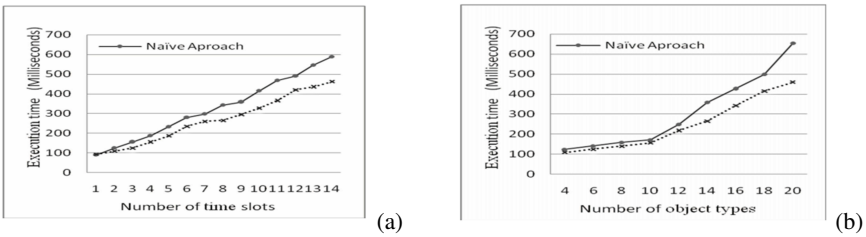
6. Experimental Results

We use Real Data Sets and Synthetic to evaluate the proposed algorithm. The real data includes 15 time snapshots and 21 distinct vehicle types and their instances. The minimum instance number is 2, the maximum instance number is 78, and the average number of instances is 19. To evaluate the performance of the algorithms, spatiotemporal data sets were generated based on the spatial data generator proposed by Huang et al. [3]. Synthetic data sets were generated for spatial frame size $D \times D$.For simplicity,the data sets were divided into regular grids whose side lengths had neighborhood relationship R.

6.1. Experimental Results for Real Data Sets

We evaluated the effect of the number of time slots on the execution time of the MDCOP algorithms using the real data set. The participation index, time prevalence index, and distance were set at 0.2, 0.8, and 100 m respectively. Experiments were run for a minimum of 1 time slot and a maximum of 14 time slots. Results show that the MDCOP Graph Miner requires less execution time than the fast MDCOP-Miner (Fig. 8a). As the number of time slots increases, the ratio of the increase in execution time is smaller for MDCOP Graph Miner than for the fast MDCOP-Miner. It shows that MDCOP Graph Miner has time cost advantage to fast MDCOP-Miner.

The participation index, time prevalence index, number of time slots, and distance were set at 0.2, 0.8,15, and 100 m, respectively. Results show that the MDCOP Graph Miner outperforms the fast MDCOP-Miner when the number of object-types increases (Fig. 8b). It is observed that the increase in execution time for the fast MDCOP-Miner is bigger than that of the MDCOP Graph Miner as the number of object-types increases for the real data set.



(a) (b)

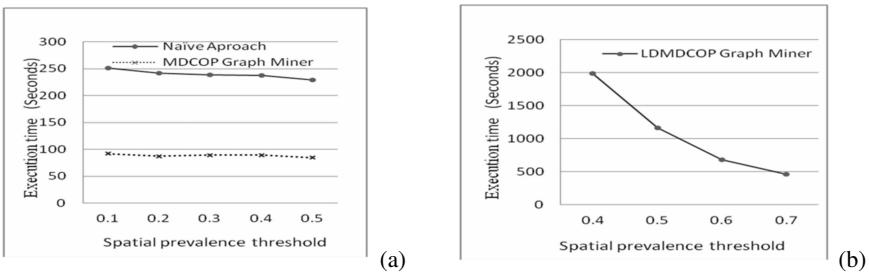
Fig. 8. (a) Effect of number of time slots and object types using real data set

6.2. Experimental Results for Synthetic Datasets

Effect of the Spatial Prevalence Threshold. We evaluated the effect of the spatial prevalence threshold on the execution times of MDCOP mining algorithms. The fixed parameters were time participation index, distance, and number of time slots, and their values were 0.8, 20 m, and 100, respectively. Experimental results show that the MDCOP Graph Miner this paper proposed is more computationally efficient than the fast MDCOP-Miner (Fig. 9a). The execution time of the MDCOP Graph Miner decreases as the spatial prevalence threshold increases, and obviously MDCOP Graph Miner is running less time than fast MDCOP-Miner with all spatial prevalence.

We also evaluated the effect of the spatial prevalence threshold on the execution time of the LDMDCOP Graph Miner using synthetic data sets. The participation index, distance and number of time slots, were set at 0.8, 20 m, 100, respectively. The results showed that the execution time of the LDMDCOP Graph Miner decreases as the time prevalence threshold increases (Fig. 9b).

Effect of the Time Prevalence Threshold We evaluated the effect of the time prevalence threshold on the execution times of MDCOP mining algorithms. The fixed parameters

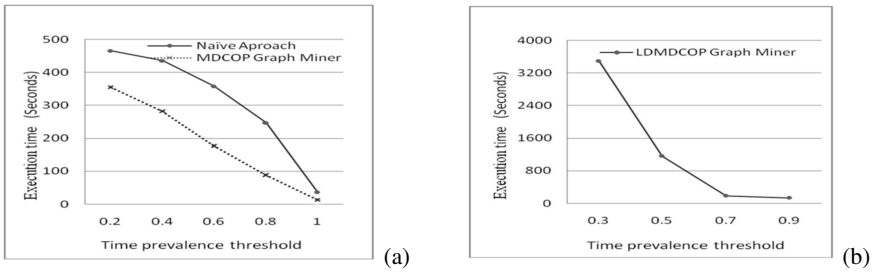


(a) (b)

Fig. 9. (a), (b) Effect of the spatial prevalence threshold using synthetic data sets

were spatial participation index, distance, and number of time slots, and their values were 0.4, 20 m, and 100, respectively. Experimental results show that the MDCOP Graph Miner this paper proposed is more computationally efficient than the fast MDCOP-Miner (Fig. 10a). The execution time of the MDCOP Graph Miner decreases more than fast MDCOP-Miner as the spatial prevalence threshold increases.

We also evaluated the effect of the time prevalence threshold on the execution time of the LDMDCOP Graph Miner using synthetic data sets. The participation index, distance and number of time slots, were set at 0.5, 20 m, 100, respectively. The results showed that the execution time of the LDMDCOP Graph Miner decreases as the time prevalence threshold increases (Fig. 10b).



(a) (b)

Fig. 10. ((a), (b) Effect of the time prevalence threshold using synthetic data sets

7. Conclusions and Future Work

In the exception detection, there are many data attributes in the data packet captured in the network. There are a lot of useless and redundant attributes, As the dimension of data object increases, the cost of intrusion detection system increases exponentially. It will not only consume a large amount of storage space, but also greatly increase the time and space complexity of the intrusion detection program. Excessive dimensions will not only increase the resource consumption of the system, but also reduce the accuracy of detection.

In order to improve the traditional spatiotemporal co-occurrence pattern mining techniques in mixed-drove spatiotemporal co-occurrence patterns detection. We designed the

method to improve calculation efficiency. In this paper, we presented a novel and computationally efficient algorithm (the MDCOP Graph Miner) for mining MDCOPs with large attack dataset. MDCOP Graph Miner represents the close relationship between the instances with MDCOP Graph structure; obtain information for mining co-occurrence patterns efficiently through accessing MDCOP Graph and design an effective pruning strategy simultaneously. Experiments show that MDCOP Graph Miner algorithm can not only get the complete and correct spatiotemporal co-occurrence patterns, but also improve computational efficiency.

Although MDCOP Graph Miner algorithm improves the efficiency in mixed-drove spatiotemporal co-occurrence patterns detection, but for large spatial and temporal attack data processing and analysis capabilities, or lack of. In order to solve this problem, on the basic of MDCOP Graph Miner, we presented an improved MDCOP Graph Miner algorithm (the LDMDCOP Graph Miner) which can deal with large spatiotemporal attack data sets. LDMDCOP Graph Miner takes MDCOP Graph as the storage structure for close relationships between instances, and storage MDCOP Graph on the hard disk. In the process of calculation, only the necessary attack data is loaded into memory, so as to solve attack data storage problems. Meanwhile, LDMDCOP Graph Miner makes index for MDCOP Graph, which improves the data query efficiency. Experiments show that LDMDCOP Graph Miner algorithm can not only get the complete and correct mixed-drove spatiotemporal co-occurrence patterns detection, as well as have high computational efficiency.

In the future, we will extend the MDCOP graph and the subsequent mining algorithm for insertions of object attack types at arbitrary time interval. We hope to investigate the idea of multi-scale relationship for different co-occurrence patterns detection.

References

1. A.Garaeva, Makhmutova, F., Anikin, I., Sattler, K.: A framework for co-location patterns mining in big spatial data. In: Petersbur, S. (ed.) 2017 XX IEEE International Conference on Soft Computing and Measurements (SCM) (2017)
2. Cao, H., Mamoulis, N., W.Cheung, D.: Discovery of collocation episodes in spatiotemporal data. ICDM pp. 823–827 (2006)
3. Celik, M., Shekhar, S., Rogers, J.P., Shine, J.A.: Mixed-drove spatio-temporal co-occurence pattern mining. IEEE Trans. Knowledge and Data Eng. 20(10), 85–99 (2006)
4. D, Barbara′.and J, C., S, Jajodia.and N, W.: A testbed for exploring the use of data mining in intrusion detection acm sigmod record. Special section on data mining for intrusion detection and threat analysis (2001) pp. 15–24 (2001)
5. Ding, Z., Güting, R.H.: Modeling temporally variable transportation networks. DASFAA pp. 154–168 (2004)
6. Ekkehard, K., Katharina, L., Martin, S.: Time-expanded graphs for flow-dependent transit times. In: Algorithms—ESA 2002, Lecture Notes in Comput. Sci., vol. 2461, pp. 599–611. Springer, Berlin (2002)
7. George, B., Shekhar, S.: Time-aggregated graphs for modeling spatiotemporal networks. ER (Workshops) pp. 85–99 (2006)
8. Horng, S.J., Su, M.Y., Chen, Y.H., Kao, T.W., Chen, R.J., Lai, J.L., Perkasa, C.D.: A novel intrusion detection system based on hierarchical clustering and support vector machines. Expert Systems with Applications 38(1), 306 – 313 (2011)

9. Huang, Y., Shekhar, S., Xiong, H.: Discovering co-location patterns from spatial data sets: A general approach. IEEE Trans,Data Eng 16(12), 1472–1485 (2004)
10. J.Gudmundsson, V.Kreveld, M., Speckmann, B.: Efficient detection of motion patterns in spatio-temporal data sets. pp. 250–257. ACM-GIS (2004)
11. J.S.Yoo, Shekhar, S.: A partial join approach for mining co-location patterns. Proc. 12th Ann. ACM Int′l Workshop Geographic Information Systems (ACM-GIS) (2005)
12. J.S.Yoo, Shekhar, S., M.Celik: A join-less approach for co-location pattern mining: A summary of results. Proc. Fifth IEEE Int′l Conf. Data Mining (ICDM) (2005)
13. Kalnis, P., Mamoulis, N., Bakiras, S.: n discovering moving clusters in spatio-temporal data. 9th Int′l Symp. on Spatial and Temporal Database(SSTD) (2005)
14. Mete, C., Shashi, S., James, R., James, A.S., Jin, S.Y.: Mixed-drove spatio-temporal co-occurence pattern mining: A summary of results. Proceedings - IEEE International Conference on Data Mining, ICDM pp. 119–128 (12 2006)
15. Minwei, T., Zhanquan, W.: Research of spatial co-location pattern mining based on segmentation threshold weight for big dataset. ICSDM 2015 - Proceedings 2015 2nd IEEE International Conference on Spatial Data Mining and Geographical Knowledge Services pp. 49–54 (2015)
16. S.Banerjee, Carlin, B., Gelfrand, A.: Hierarchical modeling and analysis for spatial data. CRC Press 22(2), 115–170 (2003)
17. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: A summary of results (2001)
18. S.Shekhar, Huang, Y., Xiong, H.: Discovering spatial co-location patterns: A summary of results. pp. 236–256. Proc. Seventh Int′l Symp. Spatial and Temporal Databases (SSTD ′01) (2001)
19. Tsang, C.H., Kwong, S., Wang, H.: Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition 40(9), 2373 – 2391 (2007)
20. Y, H., S, S., H, X.: Discovering co-location patterns from spatial datasets: a general approach. IEEE Transactions on Knowledge and Data Engineering (TKDE) 16(12), 1472–1485 (2004)

Zhanquan Wang is an associate professor at Department of Computer Science and Engineering of East China University of Science and Technology in Shanghai, China. He received his Ph.D. in computer science from Zhejiang University in China in 2005. His current research interests include Spatial Datamining, GIS, Educational technology. Contact him at zhqwang@ecust.edu.cn.

Taoli Han is currently a Master candidate at East China University of Science and Technology in Shanghai, China. He received his bachelor's degree in Computer science and technology from Hainan University in Haikou, China in 2017. Her current research interests include datamining, spatial datamining. Contact him at 2730023187@qq.com.

Huiqun Yu (Corresponding Author) is professor and PhD supervisor, received the PhD degree from the department of computer science, ShangHai JiaoTong University, in 1995. He is the senior member of IEEE and China Computer Federation and ACM members. His main research interests include trusted computing, cloud computing, software engineering and service oriented software in cloud. Contact him at yhq@ecust.edu.cn

Received: August 28, 2018; Accepted: September 10, 2019.

Computer Science and Information Systems 16(3):915–938 https://doi.org/10.2298/CSIS180901033L

A Novel Self-adaptive Grid-partitioning Noise Optimization Algorithm Based on Differential Privacy

Zhaobin Liu[1], Haoze Lv[1], Minghui Li[1], Zhiyang Li[1], and Zhiyi Huang[2]

[1] School of Information Science and Technology, Dalian Maritime University
China
Correspondence: lizy0205@gmail.com
[2] Department of Computer Science, University of Otago
hzy@cs.otago.ac.nz

Abstract. As the development of the big data and Internet, the data sharing of users that contains lots of useful information are needed more frequently. In particular, with the widespread of smart devices, a great deal of location-based data information has been generated. To ensure that service providers can supply a completely optimal quality of service, users must provide exact location information. However, in that case, privacy disclosure accident is endless. As a result, people are paying attention to how to protect private data with location information. Of all the solutions of this problem, the differential privacy theory is based on strict mathematics and provides precise definition and quantitative assessed methods for privacy protection, it is widely used in location-based application. In this paper, we propose a self-adaptive grid-partitioning algorithm based on differential privacy for noise enhancement, providing more rigorous protection for location information. The algorithm first partitions into a uniform grid for spatial two dimensions data and adds Laplace noise with uniform scale parameter in each grid, then select the grid set to be optimized and recursively adaptively add noise to reduce the relative error of each grid, and make a second level of partition for each optimized grid in the end. Firstly, this algorithm can adaptively add noise according to the calculated count values in the grid. On the other hand, the query error is reduced, as a result, the accuracy of partition count query (the query accuracy of the differential private two-dimensional publication data) can be improved. And it is proved that the adaptive algorithm proposed in this paper has a significant increase in data availability through experiments.

Keywords: Data Publication, Privacy Protection, Differential Privacy, Noise Optimization.

1. Introduction

With the increasing development of location technology, location-based service has emerged for a long time [1]. User's geographic location information can be collected by spatial data set through mobile devices and then used for server analysis to allocate tasks and increase the efficiency of human works [2] [3]. For example, the takeaway service can select the optimal worker who is the best for this task (the distance is relatively short and the overhead is relatively small) by analyzing the location of takeaway task and the worker and calculating the distance between them. Online car-hiring service should determine the best match of a driver and a passenger. The weather application can infer the

climate in our city by using our location. Hence, the service providers can supply lots of services to help people better control their lives and make important decisions with user's location information collected by devices and platform. However, for many applications, people are told to contribute their exact location information, which may cause serious privacy disclosure problem. For example, when you submit your location to the application, the attacker knows your information such as religion, profession, age by analyzing the frequency of position where you appeared. In recent years, more and more attention are paid to the privacy protection that makes a challenge for us to protect user's private information when publishing data. Obviously, it is possible to achieve when the data is completely masked such as the function $f(x) = 0$ where lost its privacy. When the query is submitted by the analysis, the answer returns a value of 0 regardless of the count operation of any part of the data set. Meanwhile, although we can regard that we completely protect the privacy, we lose the availability of query data. As a result, it has become a hot issue for researchers to maximize the availability of published data. Because it's important to protect the user's two dimensions spatial data in privacy.

We consider the differential privacy (DP) [5] as a proper method of location-based privacy for issues we described. Differential privacy is a perfect model for privacy-preserving query and analyzes under the protection of user's privacy. Intuitively, it can ensure that for a single individual data included in a data set, the result of statistical query won't be significantly changed regardless of whether this individual is in the data set or not, which means the attacker cannot infer any individual data by the statistical result. DP focuses on two drawbacks compared with the traditional privacy protection model (such as encryption algorithm, k-anonymity [25] [5]). Firstly, DP proposes the definition of privacy protection based on strict mathematics that can provide privacy for the data sets in different levels. Secondly, DP model makes the maximum assumption for the background knowledge of attacker that it assumes the attacker knows all other records expect the target record, thus the differential privacy does not need to pay attention to the attacker's background knowledge. Above all, we argue the differential privacy model is the most suitable privacy-preserving method for location-based information.

For spatial data, such as individual location information in a certain area, we need to use a data publishing algorithm based on partition, which means Private Spatial Decomposition (PSD) [6]. Partition distribution is a form of differentially private data spatial publishing which basic idea is: firstly transform the original data, and then divide the data according to certain index construction rules, and publish data according to the index structure. Each index area is marked by a calculated count value, and noise is added for privacy protection.

There are two kinds of errors in the query result. The first type of error is called noise error [7]. In order to make area D to satisfy the differential privacy, we add noise to the area which makes the original area become D', and the error relative to D is counted as:

$$Error(P) = |Count(D) - Count(D')| \tag{1}$$

The second type is called uniform assumption error [7]. In the two-dimensional spatial data publication, it is often impossible to subdivide the two-dimensional space due to space limitations, and it is necessary to estimate the statistical value in one area. The commonly used estimation method is the assumption that the points within the area are uniformly distributed. In that case, the estimated value is returned according to the ratio

of the query area. The error due to uniform distribution estimation becomes a uniform assumption error.

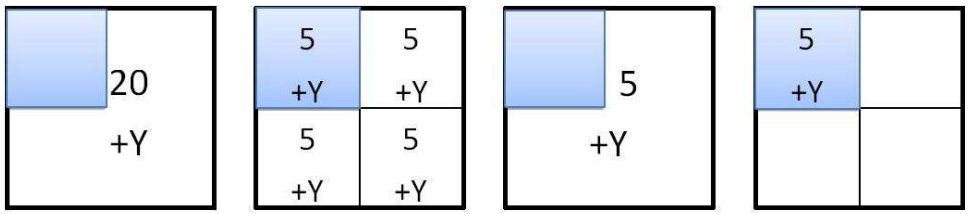


Fig. 1. Noise-added domain. (a) uniform non-partition. (b) uniform partition. (c) non-uniform non-partition. (d) non-uniform partition.

As shown in Fig 1, assume that (a) and (b) are uniformly distributed, (a) the noise is added to the area,(b) the noise is added after the area is partitioned. The numbers 5, 20 represent real statistics count for the area, +Y denotes the noise added to make this area satisfy differential privacy, and the blue district means the query area. Since the privacy budget is the same, the expected noise about two graphs (a) and (b) is the same. For the query result, (a) is (20+Y) / 4, and (b) is (5+Y). It is easy to see that the query error in the graph (b) is larger than in graph (a). However, for Figures (c) and (d), the query results are (5+Y) / 4 and (5+Y). Although Figure (c) reduces the noise error, it also increases the uniform assumption error. So under normal circumstances, the noise error and the uniform assumption error are opposite to each other. Utilizing the uniform assumption error estimate can reduce the noise error of the query result, but it will produce additional uniform assumption errors. However, subdivided regions can reduce the uniform assumption error of query results but increase the noise error. So how to balance the two kinds of errors in order to minimize the publishing error is a problem we need to discuss.

The methods described in the third section of this article all use tree structures to index sensitive data. However, the tree structure does not involve the issue of minimizing publishing errors. Literature [8] proposed uniform grid (UG) and adaptive grid (AG) to minimize publishing errors. The publishing errors mainly include the added Laplace noise error and the uniform assumption error. UG and AG quantify the two parts of the error and get the grid partition density. The UG method partitions the spatial data set equally. However,we argue that the same partition method for sparse data and dense data will bring greater uniform assumption error. On that basis, the author proposes an AG method to adaptively partition the grid according to the data density. The disadvantage of the UG and AG methods is that the same privacy budget is allocated for sparse data and dense data whose effect on the noise error of them are different. Further consideration should be given to the adaptive allocation of privacy budgets based on data point density.

We propose SGNO (Self-adaptive Grid-partitioning Noise Optimization algorithm) that focuses on geo-spatial data summarizing related domestic and foreign research status and existing algorithms. On one hand, the released data satisfies privacy protection, on

the other hand, the availability of published data(the similarity between original data and published data) is improved. We prove that the effectiveness of the algorithm is obtained in the experimental part compared with the relevant algorithm.

The rest of this paper is organized as follows: In section 2 we introduce the preliminaries and the related work. We propose our optimized algorithm for spatial decomposition and evaluate the algorithm compared with some of the previous ones in section 3 and 4. We conclude the paper in section 5.

2. Related Work

Differential privacy was originally proposed in [4] to protect the results of queries. To preserve the privacy of individuals, Mechanisms must guarantee that the contribution of each individual for a query result cannot disclose data. We introduce the related technology in this paper in detail, including the notion of differential privacy and the data publishing method based on partition.

2.1. Differential Privacy Model

The main idea of the differential privacy protection model is to perturb the data by random noise before it is published. Hence, even if an attacker knows all the other record information except the target record, the individual user's private information cannot be inferred through data mining and data analysis. In addition, differential privacy has a strict mathematical model, which can provide different degrees of privacy protection for data sets according to user requirements. This can protect users' privacy information in various types of background knowledge attacks. Hence, the differential privacy protection model has been studied and perfected by scholars since its proposed, and gradually has a rigorous theoretical system [9]. Since differential privacy is defined on neighbor data sets, it is necessary for us to introduce the definition of neighboring data sets first.

Definition 1 (Neighboring data set) For two data sets D_1 and D_2 with the same attribute structure, D_1 and D_2 are called neighboring data set, if and only if there is one different data record in D_1 and D_2.

We give the definition of differential privacy based on the neighboring data set.

Definition 2 (ε-differential privacy) A randomized algorithm A gives ε-differential privacy, for any pair of neighboring data sets D_1 and D_2 and for every set of outcomes O (O $\in$ Range(A)), A satisfies:

$$Pr[A(D_1) = O] \leq e^{\varepsilon} \cdot Pr[A(D_2) = O] \tag{2}$$

The ε in formula 2 is called privacy budget which represents the level of the privacy protection. The smaller the ε is, the higher the protection level is. In order to make the results of $A(D_1)$ and $A(D_2)$ have higher similarity, the data needs to be disturbed to a higher level, so the privacy protection level is improved and the availability of data is reduced [10]. On the contrary, the larger the value of ε, the less level of data disturbance will lead to better data availability, which may cause lower privacy protection. Nowadays, there is no good standard for the value of ε. Generally, the optimal value is constantly adjusted according to the level of privacy protection. It is always between $\ln 2$ and $\ln 5$ based on empirical evaluation.

The main idea of the differential privacy protection model is to hide the impact of a single data record on overall published data. It uses the idea of data perturbation to transform the single record by guaranteeing the invariance of the overall data on the probability to achieve the purpose of protecting the user's private data. Figure 2 shows a statistical model of differential privacy, which illustrates the results of the privacy protection model implemented on a neighboring data set.

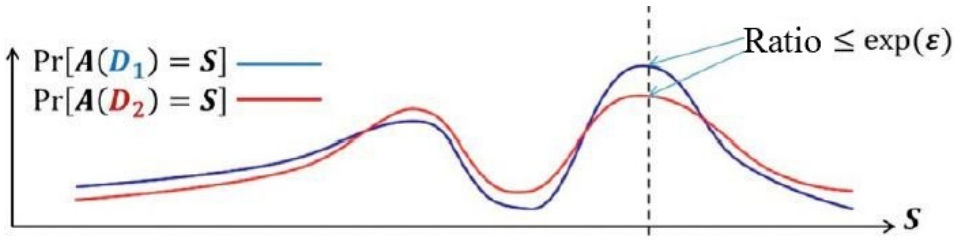


Fig. 2. Statistical Model of Differential Privacy

According to the definition of privacy we can know that it is used in data distribution mechanisms rather than directly applied to the data set. Intuitively, it can show that the behavior of differential privacy protection model on any two neighboring data sets are roughly the same in addition that the presence or absence of an individual does not affect the output of the algorithm. For example, there is a simplified medical record data set D_1 in which each record represents (name, diabetes), the second line is a boolean, 1 represents illness, and 0 represents no disease, as shown in Table 1. We assume that the opponent wants to know if Jack is sick or not and he also knows the number of rows in Jack's database. Suppose the opponent's query form is Q_i, which represents the sum of the first I rows of the $Diabetes$. In order to know Jack's prevalence, opponents perform queries Q_5 (D_1) and Q_4 (D_1) and then calculate their difference to know that Jack's disease status corresponds to 1. This example shows that personal information may be affected even if no specific personal information is queried. If we construct the data set D_2 by changing the last record to (Jack, 0) of table 1, the opponent can distinguish two neighboring data sets D_1 and D_2 by calculating Q_5-Q_4. If the opponent gets the query value Q_i through ε-differential privacy, he cannot distinguish between two neighboring data sets after selecting the appropriate value of ε.

Achieving differential privacy generally adopts two mechanisms: the Laplace mechanism and the exponential mechanism. These mechanisms contains the definition of sensitivity. To make these definition understood easily, we give an introduction of global sensitivity [11].

Definition 3 (Global sensitivity) For a function $f : D \to R^d$, for any neighboring data sets, the global sensitivity $\triangle f$ of function f is defined:

$$\triangle f = \max_{D_1 D_2} ||f(D_1) - f(D_2)||_1 \tag{3}$$

Where D refers to the data set, R_d refers to d-dimensional vector, d is a positive integer, $||f(D_1) - f(D_2)||_1$ represents the 1-order distance between $f(D_1)$ and $f(D_2)$.

Table 1. Medical Dataset of Diabetic Patients

Name	Diabetes
Ross	1
Bob	1
Phoebe	0
Carol	0
Jack	1

Global sensitivity represents the change in the output of the algorithm when changing any record in the data set.

Laplace Mechanism The Laplace mechanism achieves differential privacy by adding noise that obeys the Laplace distribution to the original real query value. As shown in Figure 3, the probability density function of the Laplace distribution is:

$$f(x|\mu, b) = \frac{1}{2b} \cdot e^{-\frac{|x-\mu|}{b}} \tag{4}$$

Where μ is position parameter, b refers to scale parameter, whose value is up to 0. When the parameter changes, the Laplace probability distribution function changes as the Figure shows.

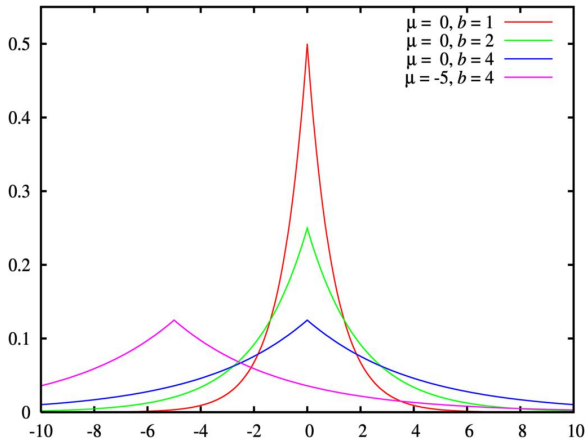


Fig. 3. Probability Density Function of Laplace Distribution

Definition 4 (Laplace mechanism) Given a function $f : D \to R^d$ over a data set D, a privacy budget ε and the global sensitivity $\triangle f$, f satisfies Laplace mechanism when:

$$L(D) = f(D) + Lap(\frac{\triangle f}{\varepsilon}) \tag{5}$$

Where $Lap(\lambda)$ means the position parameter of Laplace distribution is 0, and the scale parameter is λ, same as $\frac{\triangle f}{\varepsilon}$.

A Laplace mechanism is used for the numeric query result [12]. We can add noise to the output value of original data set to affect the result of query to protect the data set. We can know that the smaller ε corresponding larger noise value according to taking different parameter values. The noise size of the Laplace mechanism needs to achieve a good balance between the protection level and practical application according to actual requirements [13].

Composition Theorems We need to consider the privacy budget of the entire process to be controlled within ε, because a complex privacy protection scenario often requires multiple applications of the differential privacy protection model. As a result, we need to apply the two composition theorem of the differential privacy protection model [14,15]

Definition 5 (Sequential composition)

Assume a set of algorithms $A_1(D), A_2(D), ..., A_n(D)$ whose privacy budgets are $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ respectively that satisfy differential privacy. For the data set D, a new algorithm A composed of the above algorithms $A(A_1(D), A_2(D), ..., A_n(D))$, gives $\sum_{i=1}^{n} \varepsilon_i$-differential privacy model. This theorem indicates that the privacy protection level of combined algorithms is the sum of all budgets in a differential privacy protection model sequences [12].

Definition 6 (Parallel composition)

Assume a set of algorithms $A_1(D), A_2(D), ..., A_n(D)$ whose privacy budgets are $\varepsilon_1, \varepsilon_2, ..., \varepsilon_n$ respectively that satisfy differential privacy. For the data set D which can be partitioned into disjoint subcell data sets $D_1, D_2, ..., D_n$, a new algorithm A composed of the above algorithms $A(A_1(D), A_2(D), ..., A_n(D))$, gives $\max(\varepsilon_i)$-differential privacy model.

In a differential privacy protection model sequences, the privacy protection level of combined algorithms is determined by the algorithm with the largest privacy budget, where the data sets input by each algorithm are disjoint. These two kinds of composition theorems play an important role in the proof of the differential privacy protection model algorithm. At the same time, the algorithm that satisfies differential privacy can also be partitioned into sub-algorithms, and then the sub-algorithm gets the corresponding privacy budget. For example, algorithm A's privacy budget is ε. If A is divided into two processes, A_1 and A_2, then privacy budget ε can be divided into ε_1 and ε_2 and then assigned to A_1 and A_2 respectively. We only need to make A_1 and A_2 meet the privacy protection levels of ε_1 and ε_2 to satisfies differential privacy.

Metrics We usually use the degree of difference between the original and the noise-added data set to evaluate the algorithm. The commonly used error metrics are: relative error [16], absolute error [17], error variance [18], and Euclidean distance [19]. We use relative error for evaluations in this paper.

In addition, ε represents the degree of privacy protection, so it is also important to choose appropriate allocate strategies. Common allocation strategies include linear allocation, uniform distribution, exponential allocation, adaptive allocation, and mixed strategy allocation [20].

The previous partition method is mainly divided into two categories: tree-based spatial decomposition and grid-based spatial decomposition. We will introduce these two methods in detail as follows [21].

2.2. Tree-Based Spatial Decomposition

The tree-based data distribution method is a method for hierarchically decomposing spatial data. The data points are partitioned into leaf nodes so that the leaf nodes can contain a small number of data points or a small data range. Non-leaf nodes represent the sum of the counts of children's nodes. Unless otherwise specified, we assume that the tree structure is a complete binary tree that all root-to-leaf paths have the same length, and all internal nodes have the same output.

The tree-based data distribution methods can be roughly divided into two categories: data-independent partitioning and data-dependent partitioning. We define the data independent partitioning if the area is divided regardless of involving the basic data. On the contrary, when it is divided on the basis of the data, it is called the data-dependent partitioning, which may reveal data privacy [21].

Data-independent Tree Partitioning For the 2D spatial data, a representative example of the data-independent tree division is quadtree, which can be extended to high dimensions named octree and other structures to represent data. Their feature is to set the partitioning method in advance and based on the attribute domain.

Data-dependent Tree Partitioning The representative structures of data-dependent tree structure are kd-tree and Hilbert R-trees which mainly depends on the input. We focus on the construction of kd-tree and the process of combining it with differential privacy.

The main construction process of kd-tree is divided into two steps: (1) select the dimension k with the largest variance in the k-dimensional data set, and then select the median m in this dimension as the pivot to partition the data set to obtain two Subsets, while creating a tree node for storing the total calculated count values. (2) Repeat step (1) for the two subsets until all subsets can no longer be divided. We store the data of the subset to the leaf node until it can't be partitioned anymore [22].

The algorithm of using differential privacy based on kd-tree is called kd-standard [23]. Kd-standard's privacy budget is divided into two parts: First, we need to determine the median value. The segmentation line may leak the true value of the median value if you do not use differential privacy to protect the segmentation process. Then we add Laplace noise to each level calculated count value of the kd-tree. Kd-standard may also have two problems with the above-mentioned quadtree: inconsistent query results and non-uniform allocation of privacy budgets. The solution is the same as a quadtree.

Mixed Tree Partitioning The mixed tree combines the data-independent and data-dependent tree partitioning methods, in which kd-hybrid is representative. We use the tree-

dependent partitioning method in the previous l layer (l is set in advance). Then select the median with the maximum variance dimension as the pivot to divide recursively each time [24]. The rest of the tree uses a tree-independent partitioning method that sets up the partitioning process in advance. This algorithm makes the advantages and disadvantages of the two tree partitioning methods complementary, so the query results are more accurate.

2.3. Grid-Based Spatial Decomposition

In this section, we introduce the previous research achievements on the spatial data release and then gradually improve the method.

Uniform Grid Partition (UG) UG partitioning is a relatively simple way to partition the space. This method divides the data domain into $m * m$ equally sized grids, and then adds noise to a calculated count value for each grid, where m is obtained by minimizing the sum of the noise error and the uniform assumption error. The disadvantage of UG partitioning is that all areas in the data set are treated equally where dense and sparse areas are partitioned in the same way. If there are fewer data points in a region, this method will result in the region being divided too much, which increases the noise error and hardly reduces the uniform assumption error. In addition, if an area is dense, uniform grid partition can make this area too rough, which in turn leads to large uniform assumption errors. Therefore, when the area is dense, a fine-grained partitioning method should be used because the uniform assumption error in this area far exceeds the noise error. Similarly, if a region is sparse, coarse-grained partitioning is used. To overcome this problem, researchers proposed Adaptive Grids (AG) partitioning based on the uniform grid method.

Adaptive Grid Partition (AG) The AG first performs uniform grid partition in a smaller granularity which is set as $\max(10, \frac{1}{4}\lceil\frac{N\cdot\varepsilon}{c}\rceil)$ since there is a second grid division, and the privacy budget of the first layer is $\varepsilon_1 = \varepsilon \cdot \alpha$. Then, on the basis of the noise-added calculated count value N of the first layer of the grid, the AG further adaptively selects the division granularity of the second layer grid. We find that AG improves the accuracy compared with UG obviously on the second layer.

The advantage of AG partition method is that it can quantify two error sources on the basis of differential privacy to calculate the partition granularity. However, we find the drawback that AG does not adaptively allocate the privacy budget. According to the method proposed by Dwork [4], AG ignore the size of the query answer and add Laplace noise to each answer with the same scale parameters. The proposed method is more susceptible to the amount of noise which may lead to a large relative errors especially when the calculated count value of the query is small. For example, Figure 4 shows a part area of the AG partitioning results: The two numbers in each grid are the original calculated count value and the added noise value, where the noise follows the Laplace distribution with the same scale parameters. For the dashed line area which represents the user's query area, the corresponding noise added query result is 21.1 when the original query result is 11, so the query error can be calculated as $(21.1 - 11)/10 = 1.11$. Therefore, due to the accumulation of noise in the query area, the availability of published results is very low. We will focus on the shortcomings of AG and provide the corresponding solutions in Section 3.

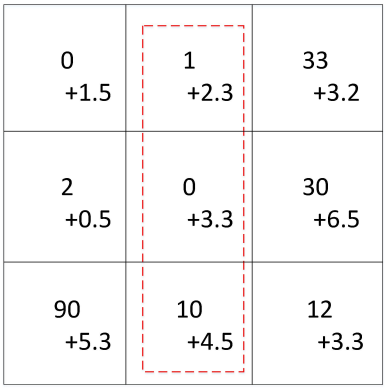


Fig. 4. Example of Injecting Noise into Grid Partition

3. Adaptive Grid-partitioning Noise Optimization Algorithm

In this section, we will introduce the improved algorithm in detail for the adaptive grid-partitioning. We introduced the quantification formulas for UG and AG, and illustrated their existing problems in related work. In that case, we proposed an optimized algorithm based on AG to solve these problems.

3.1. The Problem and Notions

The adaptive grid noise added publishing algorithm we proposed mainly addresses two aspects: (1) each grid adds Laplace noise with the same scale parameters (2) after the second-level grid is generated, the first level no longer provides useful information for the data set that may waste privacy budget. In response to the above problems, the corresponding solutions are presented as follows: First, to avoid receiving a larger relative error when querying a very small value. we propose a method to reduce the relative error by adjusting the noise scale parameter of the counter value. Second, in order to save the first-level grid privacy budget, we generate the value in second-level grid by sampling from first-level grid noise calculated count value.

Before proposing the overall steps, we first introduce some parameters and the calculation formula adjusted according to the differential privacy definition. $G = [N_1, ..., N_{m_1 * m_1}]$ is the set of count values for the first-level grid partitioning. $\Lambda = [\lambda_1, ..., \lambda_{m_1 \cdot m_1}]$ is a set of positive real numbers, corresponding to the noise scale parameter λ_i of each N_i,$(i \in [1, m_1 * m_1])$. $Y = [y_1, ..., y_{m_1 * m_1}]$ is a set of count values after adding noise to G in same scale parameter, where $y_i = g_i(G) + \eta_i$ and η_i is sampled from the Laplace distribution with scale parameter $i(i \in [1, m_1 * m_1])$. $G' = [Y'_1, ..., Y'_{m_1 * m_1}]$ is the result of optimized noise added method after the use of adaptive noise adding algorithm.

The steps of the algorithm are as follows: In first step we perform a two-dimensional data set into first-level uniform grid partition with partition granularity $m_i = \max(10, \frac{1}{4}\sqrt{\frac{N \cdot \varepsilon}{c}})$. The choice of the value c mainly depends on the uniformity of the data set. Then we add the Laplace noise in a same uniform scale parameter to

the grid set G generated in the first step to obtain the noise-added grid calculated count value set Y. Next the grid sets to be optimized from Y are selected recursively and optimized, where we use a noise optimization algorithm to satisfy the difference privacy guarantee in the meanwhile. It construct an optimized noise calculated count value set G'. Finally we perform adaptive grid partition on the basis of G' with the granularity of $m_2 = \lceil \sqrt{\frac{Y' \cdot \varepsilon_2}{c_2}} \rceil$, where $c_2 = \frac{c}{2}$. The calculation process of m_2 and c_2 are as follows: When the grid in G' is further divided into second $m_2 * m_2$ layer grids, only those queries whose query boundaries pass through the first layer grids are affected. These queries may contain 0, 1, 2... $m_2 - 1$ rows (or columns) of the second-level grids, and therefore correspond to 0, m_2, $2m_2$...$(m_2 - 1)m_2$ grids in second-level. When the query contains more than half of the second-level grid, the query is answered using Constraint Reasoning, which uses the calculated count value of the first layer minus the calculated count value in the second-level grid that is not included in the query. Therefore, the query uses an average of $\frac{1}{m_2}(\sum_{i=0}^{m_2-1} \min(i, m_2 - 1)) * m_2 \approx \frac{m_2^2}{4}$ second-level grids, which means that the average noise error is approximate $\sqrt{\frac{m_2^2}{4}} * \frac{\sqrt{2}}{\varepsilon_2}$. In addition, the mean value of the uniform assumption error is approximately $\frac{Y'}{c_0 * m_2}$. Finally, we minimize the sum of the average noise error and the uniform hypothesis error to get the minimum value of m_2 is $\lceil \sqrt{\frac{Y' \cdot \varepsilon_2}{c_2}} \rceil$, where $c_2 = \frac{c}{2}$.

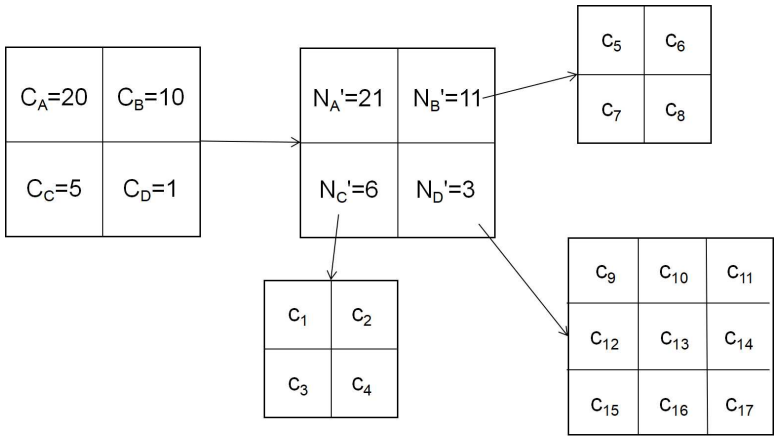


Fig. 5. Example of SGNO Partition (ε=0.5,α=0.5)

Figure 5 shows the partitioning process of SGNO, where A, B, C, and D represent the four first-level grids. Constructing differential privacy SGNO requires three steps: First, calculate the partition granularity of the first-level grid, where $c = 10$, and then add the Laplace noise with the scale parameter to the original values of A, B, C, and D to obtain the noise calculated count value N'. Secondly, we select the grid sets to be optimized from N' recursively using a noise optimization algorithm satisfying the differential privacy guarantee. The selected grid sets construct an optimized noise calculated count value set G'. In figure 5, after the second step, the N'_A is not included in the grid sets G', Next,

calculate the granularity of the second-level grid on the basis of G', where $c_2 = 5$. We use the above formulas to obtain different partition granularity for different noise-added count values N', and add Laplace noise with scale parameter to each second-level grid. Finally, SGNO structure is published with noise count.

3.2. Noise Optimized Partitioning

The principle of the NOP (Noise Optimization Partitioning) [16] is to add a uniform scale parameter $\lambda_i (1 \leq i \leq m_i * m_i)$ to the second layer of the grid to obtain a set of noise-added counts Y, and then adjust the noise scale parameter set $\Lambda = [\lambda_1, ..., \lambda_{m_1 * m_1}]$ continuously under the constraint of differential privacy to generate updated set of noise count Y'. The key question of this algorithm is whether Y' is to add noise to the original value $N_i (1 \leq i \leq m_i * m_i)$ through a new set of scale parameters, or to adjust the noise value set to generate Y' on the basis of Y, so next we consider for both cases separately.

If the first and the second level of the grid noise adding process are independent, then the total privacy budget is $2/\lambda + 2/\lambda'$, where the privacy budget of the second layer grid is $2/\lambda'$. However, the result of the first level grid partitioning is useless after the generating of the second layer grid, which means the privacy budget used by the first level grid is wasted. To solve the above problems, we argue that the noise-added value of the second level grid obeys Laplace distribution and is sampled from the distribution that depends on the first level grid, which can save part of the privacy budget.

We use formula derivation to explain the process of reducing the privacy budget: First, for the data set T_1, the first layer of partitioning produces a result of Y, and the second produces a result of Y'. Y's result is useless after Y' is generated which can quantify as the following formula:

$$Pr[T = T_1 | Y = y, Y' = y'] = Pr[T = T_1 | Y' = y'] \tag{6}$$

Formula 6 allows us to use the privacy budget more efficiently. When Equation 6 is satisfied, we can apply the Bayesian equation twice to derive the following derivation for two neighbor data sets T_1 and T_2:

$$\begin{aligned} & Pr[Y' = y', Y = y | T = T_i] \\ =& Pr[T = T_i | Y' = y', Y = y] \cdot \frac{Pr[Y' = y', Y = y]}{Pr[T = T_i]} \\ =& Pr[T = T_i, | Y' = y'] \cdot \frac{Pr[Y' = y', Y = y]}{Pr[T = T_i]} \\ =& \frac{Pr[Y' = y' | T = T_i] Pr[T = T_i]}{Pr[Y' = y']} \cdot \frac{Pr[Y' = y', Y = y]}{Pr[T = T_i]} \\ =& Pr[Y' = y' | T = T_i] \cdot Pr[Y = y, Y' = y'] \end{aligned} \tag{7}$$

Next, we consider the case where the Y and the Y' generated by the first-level and second-level grid with noise added are dependent. For a count query $q, q(T_1) - q(T_2) \in \{-1, 0, 1\}$. Y' is a random variable and satisfies two conditions: First, it obeys the Laplace

distribution with the positional parameter $q(T)$ and the scale parameter λ'. Second, it is dependent on Y. Equation 6 and 7 get the following derivation process:

$$\begin{aligned}&\frac{Pr[Y'=y', Y=y|T=T_1]}{Pr[Y'=y', Y=y|T=T_2]}\\ =&\frac{Pr[Y'=y'|T=T_1]\cdot Pr[Y=y|Y=y']}{Pr[Y'=y'|T=T_2]\cdot Pr[Y=y|Y=y']}\\ =&\frac{Pr[Y'=y'|T=T_1]}{Pr[Y'=y'|T=T_2]} \leq \exp(2/\lambda')\end{aligned} \tag{8}$$

From equation 8 we can get an upper bound of the privacy budget. which means we can achieve Y' on the basis of Y. The total privacy budget is only $2/\lambda'$ to ensure no privacy budget is wasted on Y.

From the above process, we learned that if Y obeys the Laplace distribution and samples from the Y-dependent distribution, it can save some privacy overhead. So we define the conditional probability distribution function for Y as follows:

$$\begin{aligned}&f_{\mu,\lambda,\lambda'}(y'|Y=y)=\frac{\lambda}{\lambda'}\cdot\frac{\exp(-\frac{|y'-\mu|}{\lambda'})}{\exp(-\frac{|y-\mu|}{\lambda})}\cdot\gamma(\lambda',\lambda,y',y)\\ &\gamma(\lambda',\lambda,y',y)=\frac{1}{4\lambda}\cdot\frac{1}{\cosh(\frac{1}{\lambda'})-1}\cdot(2\cosh(\frac{1}{\lambda'}\cdot\exp(-\frac{|y-y'|}{\lambda})-\exp(-\frac{|y-y'-1|}{\lambda})-\exp(-\frac{|y-y'+1|}{\lambda}))\end{aligned} \tag{9}$$

The probability density function of Y' can be further obtained from the conditional probability density function. When $\mu \leq y$, $\xi = \min\{\mu, y-1\}$. We can obtain the following formula according to the conditional probability density function for $y' \leq \xi$.

$$\begin{aligned}&f(y')=e^{y'/\lambda'}\cdot\gamma(\lambda',\lambda,y',y)\cdot\frac{\lambda}{\lambda'}\cdot\exp(\frac{-\mu}{\lambda'}+\frac{y-\mu}{\lambda})\\ &\gamma(\lambda',\lambda,y',y)=e^{y'/\lambda}\cdot\frac{1}{4\lambda}\cdot\frac{1}{\cosh(\frac{1}{\lambda'})-1}\cdot(2\cosh(\frac{1}{\lambda'}\cdot\exp(-\frac{-y}{\lambda})-\exp(-\frac{1-y}{\lambda})-\exp(-\frac{-1-y}{\lambda}))\end{aligned} \tag{10}$$

Formula 7 can be simplified as $f(y') \propto \exp(y'/\lambda + y'/\lambda')$, meanwhile, we can obtain $y' \in (\xi, y-1]$, $f(y') \propto \exp(y'/\lambda - y'/\lambda')$ and $y' \in (y+1, +\infty]$, $f(y') \propto \exp(-y'/\lambda - y'/\lambda')$. Based on this, the random variables $\theta_1, \theta_2, \theta_3$ that follow the probability density function f can be calculated and their formula is as follows:

$$\begin{aligned}\theta_1 =&\int_{-\infty}^{\xi} f(y')dy'\\ =&\frac{\lambda\cdot(\cosh(\frac{1}{\lambda'})-\cosh(\frac{1}{\lambda})\cdot\exp(\frac{1}{\lambda'}+\frac{1}{\lambda})\cdot(\xi-\mu))}{2(\lambda'+\lambda)\cdot(\cosh(\frac{1}{\lambda'})-1)}\end{aligned} \tag{11}$$

$$\begin{aligned}\theta_2 =&\int_{\xi}^{y-1} f(y')dy'\\ =&\frac{\lambda\cdot(\cosh(\frac{1}{\lambda'}))\cdot(e^{\frac{1}{\lambda'}}-e^{\frac{1}{\lambda}})\cdot(1-e^{-\frac{1}{\lambda'}-\frac{1}{\lambda}})}{4(\lambda-\lambda')\cdot(\cosh(\frac{1}{\lambda'})-1)}\cdot(1-\exp((\frac{1}{\lambda'}-\frac{1}{\lambda})\cdot(\xi-y+1))\end{aligned} \tag{12}$$

$$\begin{aligned}\theta_3 &= \int_{y+1}^{+\infty} f(y')dy' \\ &= \frac{\lambda \cdot (\cosh(\frac{1}{\lambda'}) - \cosh(\frac{1}{\lambda}) \cdot \exp(\frac{\mu-y-1}{\lambda'} - \frac{\mu-y+1}{\lambda}))}{2(\lambda'+\lambda) \cdot (\cosh(\frac{1}{\lambda'}) - 1)}\end{aligned} \tag{13}$$

For the remaining space $(y-1, y+1)$, we obtain its probability density function through the standard importance sampling function. We will introduce the specific process in the algorithm. The formula of φ in the algorithm 1 is as follows:

$$\varphi = \frac{1}{2\lambda'} \cdot \frac{\cosh(\frac{1}{\lambda'}) - \exp(-\frac{1}{\lambda})}{\cosh(\frac{1}{\lambda'} - 1)} \cdot \exp(\frac{y-\mu}{\lambda} - \frac{\max\{0, y-\mu-1\}}{\lambda'}) \tag{14}$$

The algorithm 1 represents a specific process of NOP, where the input is original calculated count value μ, a noise-added calculated count value y, an original scale parameter λ, and an adjusted scale parameter λ', and the output is an updated y.

3.3. Self-adaptive Grid-partitioning Noise Optimization Algorithm

We use the algorithm 2 to describe the grid-based adaptive noise-added publishing method in detail. The pseudocode of this algorithm is shown in Algorithm 2. The input of the algorithm is data set T, privacy budget ε, initial privacy budget λ_{max}, and privacy budget variance λ_Δ. The output is Adaptive Grid AG. The algorithm first averages the data set to get a uniform grid set UG and an original privacy set Λ. Then the grid to be optimized is selected recursively from the UG. We adjust the scale parameters and complete the noise optimization process with former algorithm. If this process does not satisfy the differential privacy, the changes made to the noise scale parameters are restored. Finally, AG is adaptively partitioned with the updated UG set.

We need to explain the process of selecting the grid to be optimized. Ideally, running a noise optimization algorithm on a selected set of grids may reduce the overall more error and make slightly lower privacy overhead, so the criteria for PickQueries function selection grids is to maximize the ratio between the value of overall error reduction and privacy budget increase value. First, calculate the privacy budget increase value. Each grid has the same scale parameter $\lambda_i (i \in [1, m_1 * m_1])$ at first with the global sensitivity $g_1 = \sum_{i \in [1, m_1 * m_1]} \frac{2}{\lambda_i}$. Then we use the noise optimized algorithm for the selected $j-th$ grid, and the corresponding global sensitivity is changed to $g_2 = \frac{2}{\lambda_j - \lambda_\Delta} + \sum_{i \in [1, m_1 * m_1] \wedge i \neq j} \frac{2}{\lambda_i}$, the privacy budget applied by the noise optimized algorithm is $g_2 - g_1 = \frac{2}{\lambda_j - \lambda_\Delta} - \frac{2}{\lambda_j}$. Second, we calculate the total error reduction value. The relative error of each grid is $\frac{\lambda_i}{max\{y_i, \delta\}} (i \in [1, m_1 * m_1])$. Then, the relative error of grid UG is $\sum_{i \in [1, m_1 * m_1]} \frac{\lambda_i}{max\{y_i, \delta\}}$. After applying the noise optimized algorithm, the total relative error becomes $\frac{\lambda_j - \lambda_\Delta}{max\{y_i, \delta\}} + \sum_{i \in [1, m_1 * m_1] \wedge i \neq j} \frac{\lambda_i}{max\{y_i, \delta\}}$, and the change of total relative error is obtained to $\frac{2\lambda_j - \lambda_\Delta}{max\{y_i, \delta\}}$, so that the average relative error variation is $\frac{1}{m_1 \cdot m_1} \cdot \frac{2\lambda_j - \lambda_\Delta}{max\{y_i, \delta\}}$. Therefore, the ratio between the overall error reduction value and the privacy budget increase value is $\frac{1}{m_1 \cdot m_1} \cdot \frac{2\lambda_j - \lambda_\Delta}{max\{y_i, \delta\}} / (\frac{2}{\lambda_j - \lambda_\Delta} - \frac{2}{\lambda_j})$. So we choose the grid that can maximize this ratio as the set of grids to be optimized.

Algorithm 1: NOP (μ, y, λ, λ')

Input: $\mu, y, \lambda, \lambda'$
Output: y
Initial: $mark = true$
if $\mu > y$ **then**
    μ=-μ,y=-y
    $mark = false$
$\xi = \min\{\mu, y - 1\}$
generate a random variable u uniformly distributed in $[0, 1]$
if $\mu \in [0, \theta_1]$ **then**
    $f(y') \propto \exp(y'/\lambda + y'/\lambda')$ // y' *is a random variable generated from* $(-\infty, \xi]$;
    else
        if $\mu \in [\theta_1, \theta_1 + \theta_2]$ **then**
            $f(y') \propto$
            $\exp(y'/\lambda' - y'/\lambda)$ // y' *is a random variable generated from* $(\xi, y - 1]$;
    else
        if $\mu \in [1 - \theta_3, 1]$ **then**
            $f(y') \propto \exp(-y'/\lambda -$
            $y'/\lambda')$ // y' *is a random variable generated from* $(y + 1, \infty)$;
    else
        while $true$ **do**
            generate a random variable y' uniformly distributed in (y-1,y+1)
            generate a random variable μ' uniformly distributed in [0,1]
            if $\mu' \leq f(y')/\varphi$ **then**
                $break$;
if $mark = true$ **then**
    return y'
    else
        return $-y'$

3.4. Privacy Analysis

We propose self-adaptive grid-partitioning noise optimization algorithm after the first-level partition. The algorithm first adds Laplace noise with uniform scale parameters to each grid. Then continue the process if the grid satisfies the ε-differential privacy, otherwise return empty set. Next, the grid to be optimized is recursively selected in the grid set, and the corresponding noise scale parameters are changed. We judge the condition that whether the ε-differential privacy is satisfied. If it is satisfied, the noise optimized algorithm is continued to be called; Otherwise, the changes made to the noise scale parameters are restored to satisfy the ε-differential privacy requirements. In summary, the proposed algorithm generally satisfies ε-differential privacy.

Algorithm 2: $SGNO\ (T, N, \delta, \varepsilon, \lambda_{max}, \lambda_{\Delta}, \alpha)$

Input: $Dataset\ T$, $the\ sanity\ bound\ \delta$, $privacy\ budget\ \varepsilon$, $constant\ \lambda_{max}, \lambda_{\Delta}$,
Output: $Adaptive\ Grids\ AG$
Initial: $let\ T\ be\ partitioned\ uniformly\ with\ m_1 = \max(10, \frac{1}{4}\lceil\frac{N\varepsilon}{c}\rceil)$
$let\ m = m_1 * m_1\ and\ g_i\ be\ the\ i-th(i \in [1, m,])\ grid\ in\ UG$
$initialize\ \Lambda = [\lambda_1, ..., \lambda_{m_1 \cdot m_1}], such\ that\ \lambda_i = \lambda_{max}$
if $GS(UG, \Lambda) > \varepsilon$ **then**
    $return\ \emptyset$
$Y = LaplaceNoise(UG, \Lambda)$ // *Add Laplace noise to every grid in UG*
$Let\ UG' = UG$
while $UG' \neq \emptyset$ **do**
    $U_{\Delta} = PickQuries(UG', Y, \Lambda, \delta)$
    for $i\ from\ 1\ to\ m$ **do**
        if $g_i \in U_{\Delta}$ **then**
            λ_i=λ_i-λ_{Δ}
    if $GS(UG, \Lambda) \leq \varepsilon$ **then**
        for $i\ from\ 1\ to\ m$ **do**
            if $g_i \in U_{\Delta}$ **then**
                y_i=$NOP(g_i, y_i, \lambda_i + \lambda_{\Delta}, \lambda_i)$
        else
            for $i\ from\ 1\ to\ m$ **do**
                if $g_i \in U_{\Delta}$ **then**
                    λ_i=λ_i+λ_{Δ}
        $UG' = UG \setminus U_{\Delta}$
$Update\ UG\ by\ Y$
$let\ UG\ be\ partitioned\ by\ m_2 = \lceil\frac{Y'(1-\alpha)\varepsilon}{C_2}\rceil\ to\ get\ AG$
Return AG;

4. Evaluation

In this section we compare the effectiveness of the algorithm proposed in section 4 with some previous methods. We introduce the process and results in detail.

4.1. Environment

Experiment Platform Table 2 shows the relevant configuration information of the experiment platform. We have implemented the encoding algorithm proposed in section 4 in the Linux operating system.

Experiment Database We chose three data sets for spatial data and conducted experiments separately because our two data publishing algorithms are applied to different forms of data sets.

Table 2. Configuration of Experiment Platform

Hardware and software	Configuration
Processor	Intel@ Xeon@ CPU E5-2670 2.27 GHz
Internal storage	32GB
Hardware	1TB Mechanical hard disk
Network card	Intel 82551 10M/100M Adaptive network card
OS	Ubuntu Server 16.04.1 LTS

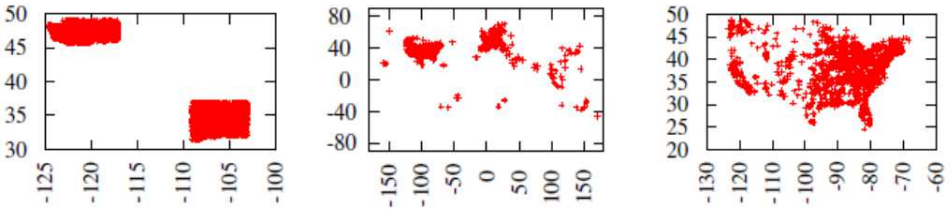


Fig. 6. Illustration of datasets

In order to verify our proposed grid-based adaptive noise-added publishing algorithm, we used three datasets shown in Figure 6.

(1)$Road$: This data set consists of the GPS coordinates of road intersections in Washington State and New Mexico and is derived from the 2006 U.S. census data in the TIGER/Line data set. There are approximately 1.6 million data points in the data set, roughly corresponding to human activities. As shown in the first picture of Figure 6, the distribution of data points is somewhat special. Two data points are dense where is distributed in two states, and almost no data points in there.

(2)$Checkin$: This data set consist of the check-in data of the location-based social network Gowalla which records the time and location of the user's check-in from February 2009 to October 2010. We use the location information for evalution. There are approximately one million data points in the data set. As shown in the second picture of Figure 6, the data distribution is sparse.

(3)$Landmark$: This data set contains information on the location of landmarks such as schools, post offices, shopping malls, construction sites, and train stations in 48 states in the United States. It originated from the 2010 Census TIGER. The data set contains approximately $900,000$ data points. As shown in the third picture of Figure 6, the data distribution is relatively uniform.

Table 3 gives the detailed information of the three data sets, including the number of data points, the size of the domain, and the size of the query area used in the evaluation of the experiment, where q_1 is the smallest query area and q_6 is the largest query area.

Experimental Process For spatial data, we propose a self-adaptive grid-based algorithm for adding noise, which is based on AG and adaptively adds noise according to the number of data points in each grid. In addition to comparing it with AG, we also compare it with

Table 3. Information on Datasets

Dataset	Number of data points	Size of the domain	Size of the query area q_1	Size of the query area q_6
Road	1.6M	25*20	0.5*0.5	16*16
Checkin	1M	360*150	6*3	192*96
Landmark	0.9M	60*40	1.25*0.625	40*20

the mixed-tree partitioning algorithm [22] which belongs to the same general-distributed class publishing algorithm to verify the effectiveness of the grid-partitioning algorithm.

Evaluation Metrics The differential privacy protection mechanism is mainly obtained by adding noise to the original calculated count value. This mechanism has two purposes. On one hand, it should to protect the privacy of each user. On the other hand, it should obtain that the published results will be still usable. Therefore, we quantify these two aspects through various index parameters, hoping to reach a balance.

For grid-based spatial data publishing algorithms, we measure the accuracy of published data by calculating relative errors.

For a query r, we use $A(r)$ to represent the correct answer for r. For the method M and a query r, we use to represent the query r which is answered using an index structure constructed by the method M. The formula for the relative error is:

$$RE_M(r) = \frac{|Q_M(r) - A(r)|}{max\{A(r), \rho\}} \tag{15}$$

where the query r has 6 kinds of sizes, q_1 is the smallest, and the length and width of q_{i+1} are respectively increased by 2 times on the basis of q_i, and q_6 is maximal and covers $1/4$ to $1/2$ of the entire space. The specific information is shown in Table 3. We randomly generate 200 queries for each query size and calculate their relative errors.

ρ is set to $0.001|D|$, where D represents the total number of data points in the data set. The reason why we maximize the denominator is to prevent $A(r) = 0$. When the query r is medium in size, $RE_M(r)$ tends to be the largest. When the query is large, it may be small because $A(r)$ is large.

4.2. Evaluation

We proposed a grid-based spatial partitioning data publishing algorithm for publishing geo-spatial data based on a differential privacy protection model. In the experimental results, K_{hy} represents the kd-hybrid algorithm proposed in [24], UG represents uniform grid partition, AG represents adaptive grid partition, and SGNO represents our proposed grid-based advanced noise-added self-adaptive grid publishing algorithm. When $\varepsilon = 0.1$, the granularity of the Road, Checkin, and Landmark datasets is 126, 100, and 95, respectively calculated by the UG method by the formula. When $\varepsilon = 1$, the granularity of the Road, Checkin, and Landmark data sets is 400, 316, and 300, respectively. When

$\varepsilon = 0.1$, the first level of the AG method divides the granularity of the reference [26] recommended value of the experiment, which means the granularity of the Road, Checkin, and Landmark data sets is 16, 32, and 32 respectively. When $\varepsilon = 0.1$, the granularity of the Road, Checkin, and Landmark data sets is 32, 64, and 64 respectively. UG and AG take the corresponding $c = 10$, $c_2 = 5$, $\beta = 0.5$, in the granularity formula. Figure 7 to Figure 12 give the average curve of the relative error of the random query region for the four algorithms on the three data sets when the value of ε is 0.1 or 1 respectively. In order to make the experimental results more general, we will run the four algorithms 50 times, and finally take the average of them.

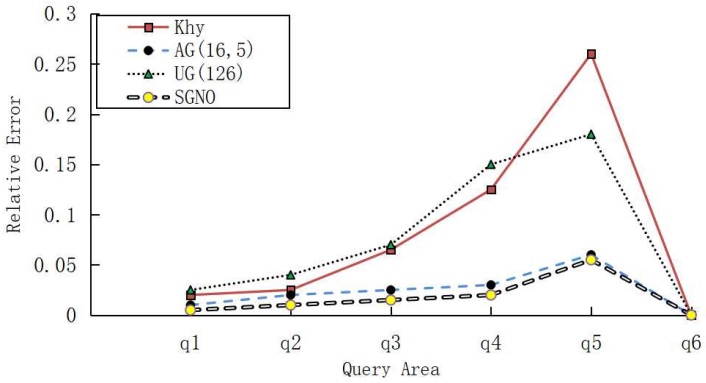


Fig. 7. Experiment Result on Dataset Road with $\varepsilon = 0.1$

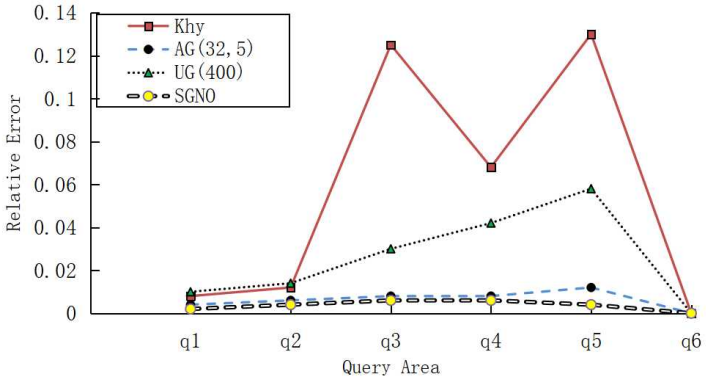


Fig. 8. Experiment Result on Dataset Road with $\varepsilon = 1$

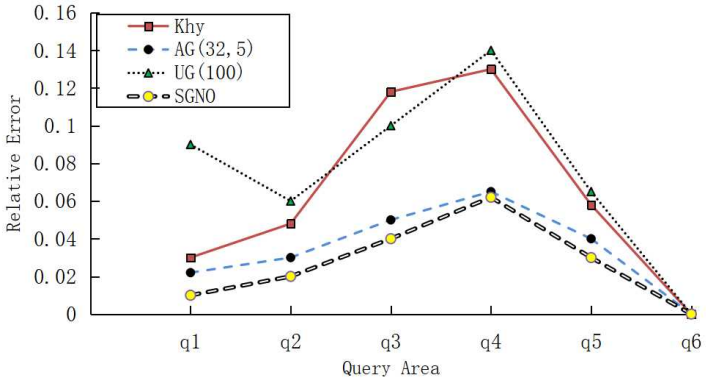


Fig. 9. Experiment Result on Dataset Checkin with $\varepsilon = 0.1$

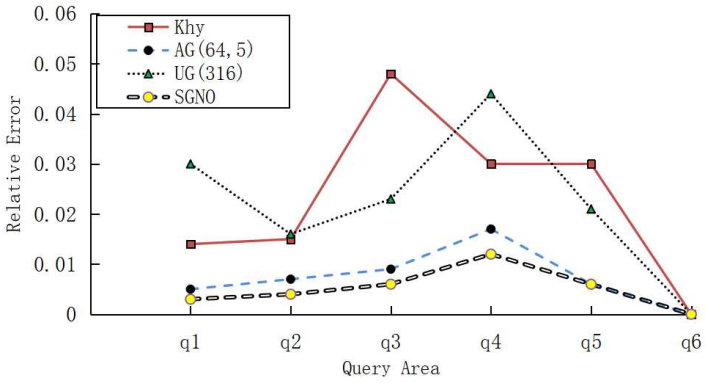


Fig. 10. Experiment Result on Dataset Checkin with $\varepsilon = 1$

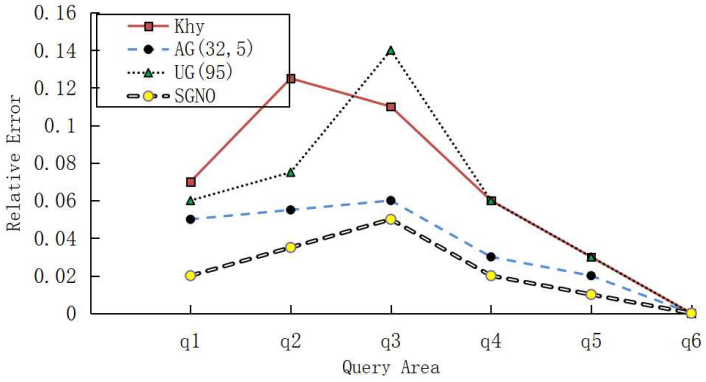


Fig. 11. Experiment Result on Dataset Landmark with $\varepsilon = 0.1$

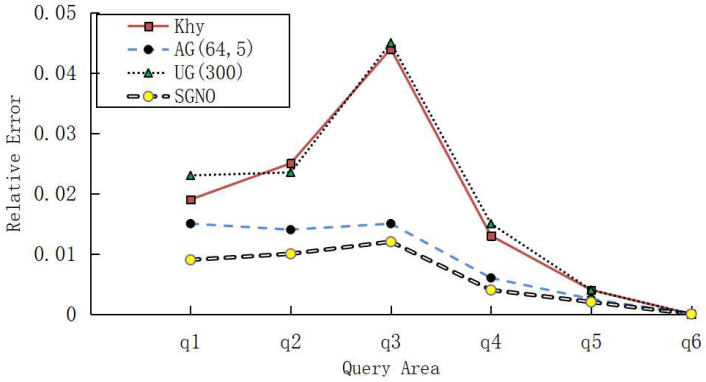


Fig. 12. Experiment Result on Dataset Landmark with $\varepsilon = 1$

Figure 7 and Figure 8 show the experimental results for the data set Road at $\varepsilon = 0.1$ and $\varepsilon = 1$ respectively. It can be seen that the relative error of the four algorithms increases with the increase of the query area, and the relative error at the query q_5 reaches the maximum, at q_6 decreases sharply and is almost close to zero. The reason for this trend is that the query area q_6 occupies between $1/4$ and $1/2$ of the entire query space, making the true answer value large, thus the relative error is small.

When $\varepsilon = 0.1$, the relative errors of the first four queries of kd-hybird and UG are relatively close, but the kd-hybird changes more greatly in the latter two queries. The AG and our method SGNO are superior to kd-hybird and UG on any query. The relative error of the first four queries of SGNO is better than AG, and the query result of q_5 is close to AG. The reason is that when the query range is relatively large, the actual query value is also relatively large, which makes the effect of noise optimization cannot be clearly displayed. When $\varepsilon = 1$, the relative error of the corresponding four algorithms is reduced since the degree of privacy protection is reduced thus the amount of added noise is reduced too.

Figure 9 and Figure 10 show the experimental results of the data set Checkin when $\varepsilon = 0.1$ and $\varepsilon = 1$ respectively. It can be seen that the relative error of the four algorithms increases with the increase of the query area, and the relative error at the query q_4 reaches a maximum when decreasing sharply at q_5 and q_6.

When $\varepsilon = 0.1$, the results of kd-hybrid and UG are mostly similar. The overall trend of UG and SGNO is relatively close, but the relative error of SGNO is generally lower than AG, especially when the query area is relatively small. The reason is that SGNO optimizes the amount of noise added in each grid. In that case, we reduce the relative error of each grid so the entire error is improved. However, when the query area is relatively large, the real calculated count value is relatively large, making the optimization effect difficult to show. When $\varepsilon = 1$, kd-hybrid and UG results change more drastically. Figure 11 and Figure 12 show the results of the Landmark data sets when $\varepsilon = 0.1$ and $\varepsilon = 1$ respectively. With the increase of the query area, the four algorithms increase the relative error, and because the data points of the data set are relatively evenly distributed, the relative error reaches the maximum at the relatively small q_3 in the query area. When

$\varepsilon = 0.1$ and $\varepsilon = 1$, the overall trend of UG and SGNO is relatively close. SGNO is mostly better than the other three algorithms.

From the figure, we can observe our SGNO methods have performed better than other methods. The performance of the UG method is similar to that of the kd-hybrid method. Above all, the results of SGNO in most queries are better than the other three algorithms.

5. Conclusion

This article is based on application scenarios for the partition based data distribution algorithm. For the partition-based data distribution method, we first uniformly partitions the original spatial data, add a Laplace noise with uniform scale parameters, and then select the set of grids to be optimized in a standard way that is based on the maximum ratio between the value of overall error reduction and privacy budget increase value, then operate noise optimized algorithm for the grid. This process is recursive until all grids have been optimized. This grid-based self-adaptive noise-added publishing algorithm solves the problem of the noise scale parameters added uniformly to each grid and the waste of the first-level grid privacy budget.

At the same time, for the above differential privacy data publishing algorithm, problem how to support dynamic data partitioning is the future research direction.

Acknowledgements Supported by the Double-class Construction Innovation Project 014-3190518. Supported by the National Nature Science Foundation of China 61370198, 61370199, 61672379 and 61300187. Supported by the Liaoning Provincial Natural Science Foundation of China NO. 2019-MS-028.

References

1. Xiong, P., Zhu, T. Q., Meng, X. F.: A Survey on Differential Privacy and Applications, Chinese Journal of Computers, 37(1), 101-120.(2014)
2. Gherari, M., Amirat, A., Laouar, R., Oussalah, M.: A smart mobile cloud environment for modelling and simulation of mobile cloud applications, International Journal of Embedded Systems, 9(5), 426-443.(2017)
3. Chen, P., Chen, J., Huang, J.: Multi-user location-dependent skyline query based on dominance graph, International Journal of Computational Science and Engineering, 13(3), 209-218.(2016)
4. Dwork, C.: Differential privacy[J], Lecture Notes in Computer Science, 26(2), 1-12.(2006)
5. Li, N., Li, T., Venkatasubramanian, S.: T-closeness: privacy beyond k-anonymity and l-diversity[C], Proceeding of the IEEE 23rd International Conference on Data Engineering (ICDE), 106-115.(2007)
6. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differential private spatial decompositions, IEEE International Conference on Data Engineering (ICDE), 20-31.(2012)
7. Zhang, X. J., Meng, X. F., Chen, R.: Differentially Private Set-Valued Data Release against Incremental Updates, International Conference on Database Systems for Advanced Applications, 392-406.(2013)
8. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data[C], Proceedings of IEEE 29th International Conference on Data Engineering (ICDE), 757-768.(2012)
9. Ebadi, H., Sands, D., Schneider, G.: Differential privacy: now it's getting personal[C], Proceedings of the 42nd Annual Symposium on Principles of Programming Languages, 69-81.(2015)

10. Gruska, D. P.: Differential privacy and security[J], Fundamenta Infomaticae, 143(1), 73-87.(2016)
11. Dwork, C., McSherry, F., Nissim, K. et al.: Calibrating noise to sensitivity in private data analysis, Theory of Cryptography, 265-284.(2006)
12. Mcsherry, F., Talwar, K.: Mechanism design via differential privacy, IEEE Symposium on Foundations of Computer Science, 94-103.(2007)
13. Geng, Q., Viswanath, P.: The optimal noise-adding mechanism in differential privacy, IEEE Transactions on Information Theory, 62(2), 925-951.(2016)
14. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy, Proceedings of the 32nd International Conference on Machine Learning, 63(6), 4037-4049.(2015)
15. Mcsherry, F. D., Meng, X. F.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis, Communications of the ACM, 53(9), 89-97.(2015)
16. Xiao, XX., Bender, G., Hay, H. et al.: iReduct:differential privacy with reduced relative errors, ACM SIGMOD International Conference on Management of Data, 229-240.(2011)
17. Li, Y. D., Zhang, Z., Winslett, M. et al.: Compressive mechanism:utilizing sparse representation in differential privacy, Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society(WPES), 177-182.(2011)
18. Peng, S., Yang, Y., Zhang, Z. et al.: DP-tree:indexing multi-dimensional data under differential privacy, Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 864-864.(2012)
19. Hardt, M., Talwar, K.: On the geometry of differential Privacy, Proceedings of the 42nd Annual ACM Symposium on Theory of Computing(STOC), 705-714.(2010)
20. Zhang, X. L., Wu, Y. J., Wang, X. D.: Differential Privacy Data Release through Adding Noise on Average ValuE, International Conference on Network and System Security, 37(1), 417-429.(2012)
21. Dwork, C.: Differential privacy: a Survey of results, International Conference on Theory and Applications of MODELS of Computation, 1-19.(2008)
22. Cormode, G., Procopiuc, C., Srivastava, D. et al.: Differentially private spatial decompositions, Proceedings of IEEE 28th International Conference on Data Engineering (ICDE), 41(4), 21-31.(2011)
23. Kamel, I., Faloutsos, C.: Hilbert R-tree: an improved R-tree using fractals, International Conference on Very Large Data Bases, 500-509.(1994)
24. Roy, I., Setty, S. T. V., Kilzer, A. et al.: Airavat: security and privacy for MapReduce, Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation(NSDI), 297-312.(2010)
25. Machanavajjhala, A., Kifer, D., Gehrke, J. et al.: L-diversity: privacy beyond k-anonymity, ACM Transactions on Knowledge Discovery from Data(TKDD), 1(1), 3-14.(2007)
26. Xiao, Y., Xiong, L., Yuan, C.: Differentially private data release through multidimensional partitioning, Proceedings of the 7th VLDB Workshop on Secure Data Management (SDM), 150-168.(2011)

Zhaobin Liu received the Ph.D degree in computer science from Huazhong University of Science and Technology, China, in 2004. He is currently a Professor in the school of information science and technology, Dalian Maritime University, China. He has been a Senior Visiting Scientist at The University of Auckland, New Zealand, in 2017, and a visiting scholar at University of Central Florida, USA, in 2014 and University of Otago, New Zealand in 2008 respectively. His research interests include big data, cloud computing and data privacy.

Haoze Lv is currently in school with the Department of Computer Science in Dalian Maritime University. His research mainly focuses on Big data and privacy protection.

Minghui Li is currently pursuing the master's degree with the Department of Computer Science in Dalian Maritime University. Her research mainly focuses on Big data and privacy protection.

Zhiyang Li (corresponding author) is currently an associate professor at the Information Science and Technology College, Dalian Maritime University, China. He received the Ph.D. degree in computation mathematics from Dalian University of Technology, China in 2011. His research interests include computer vision, cloud computing and data mining.

Zhiyi Huang received the BSc degree in 1986 and the PhD degree in 1992 in computer science from the National University of Defense Technology (NUDT) in China. He is an Associate Professor at the Department of Computer Science, University of Otago. He was a visiting professor at EPFL and Tsinghua University in 2005, a visiting scientist at MIT CSAIL in 2009, and a visiting professor at Shanghai Jiao Tong University in 2013. His research fields include parallel/distributed computing, multicore architectures, operating systems, green computing, cluster/grid/cloud computing, high-performance computing, and computer networks. He has more than 130 publications in peer-reviewed conferences and journals, many of which are top ranked.

Received: September 1, 2018; Accepted: September 12, 2019.

Computer Science and Information Systems 16(3):939–957 https://doi.org/10.2298/CSIS180923034D

Enhanced Artificial Bee Colony with Novel Search Strategy and Dynamic Parameter

Zhenxin Du[1,2] and Keyin Chen[3,4]

[1] School of Computer Information Engineering,Hanshan Normal University
Chaozhou 521041, China
duzhenxinmail@163.com
[2] College of Information Engineering, Shanghai Maritime University
Shanghai 201306, China
1355121995@qq.com
[3] Nanjing Research Institute for Agricultural Mechanization, Ministry of Agriculture and Rural Affairs
Nanjing 210014, China
308829497@qq.com
[4] School of Information and Communication Engineering, HeZhou University
Hezhou 542899, China

Abstract. There is only one guiding solution in the search equation of Gaussian bare-bones artificial bee colony algorithm (ABC-BB), which is easy to result in the problem of premature convergence and trapping into the local minimum. In order to enhance the capability of escaping from local minimum without loss of the exploitation ability of ABC-BB, a new triangle search strategy is proposed. The candidate solution is generated among the triangle area formed by current solution, global best solution and any randomly selected elite solution to avoid the premature convergence problem. Moreover, the probability of crossover is controlled dynamically according to the successful search experience, which can enable ABC-BB to adapt all kinds of optimization problems with different landscapes. The experimental results on a set of 23 benchmark functions and two classic real-world engineering optimization problems show that the proposed algorithm is significantly better than ABC-BB as well as several recently-developed state-of-the-art evolution algorithms.

Keywords: artificial bee colony, triangle search, dynamic parameter, engineering optimization.

1. Introduction

With the continuous development of science and technology, many global optimization problems constantly exist in all most of engineering and science fields, such as queuing system [46]and structural design [4][16]. Unfortunately, many of these problems are often characterized as non-convex, discontinuous or non differentiable, thus it is difficult to deal with them with traditional optimization algorithms. Evolution algorithms (EAs), as a powerful tool, are playing an increasingly important role in solving such kind of problems, such as Particle Swarm Optimization (PSO) [42], Ant Colony Optimization (ACO) [18], Differential Evolution (DE) [43], Artificial Bee Colony (ABC) algorithm [30]. ABC is a

new EA proposed by Karaboga, which is drawn inspiration from the intelligent foraging behavior of honey bees. The comparison results indicate that ABC performs competitively and effectively when compared to the selected state-of-the-art EAs, such as DE and PSO [31]. Owing to its simple structure, easy implementation and outstanding performance, ABC has received increasing interest and has been widely used in many engineering optimization problems since its invention such as network problems [36], image problems [9], engineering problems [29], clustering problems [20][32][34].

However, similar to other EAs, ABC also tends to suffer from the problem of poor convergence. The possible reason is that the solution search equation which is used to generate new candidate solutions, has good exploration capability but poor exploitation capability [21][49], and thereby it causes the problem of slow convergence speed. Therefore, the performance of ABC can be improved by enhancing the exploitation ability and finding better balance between exploration and exploitation. A large number of improved ABC variants have been presented by exploiting the valuable information from the current best solution or other good solutions. Firstly, Zhu and Kwong proposed [49] a global best (gbest)-guided ABC (GABC) algorithm based on the inspiration of PSO. In GABC, the information of the gbest solution is incorporated into the solution search equation of ABC to improve the exploitation. However, as claimed in [25], the search equation of GABC may cause an oscillation phenomenon and thus may also degrade convergence since the guidance of the last two terms may be in opposite directions. Afterwards, Zhou [48] proposed an improved ABC with Gaussian bare-bones search equation (ABC-BB for short) algorithm based on the utilization of the global best solution to make up the defect of GABC. Moreover, they proposed an ensemble algorithm composed of ABC-BB and general opposition learning initialization strategy (GOBL), named GBABC. In ABC-BB, an important feature of the newly proposed search equation is that positions of the new food sources are sampled through a Gaussian distribution with dynamical mean value and variance value. The experimental results demonstrate the effectiveness of ABC-BB and GBABC in solving complex numerical optimization problems when compared with other algorithms.

However, in ABC-BB and GBABC, there is only one guiding individual in the search equation of Gaussian bare-bones artificial bee colony algorithm (ABC-BB), which is easy to result in the problem of premature convergence. In order to overcome this drawback, a new triangle search strategy is proposed in this paper. The candidate solution is generated among the triangle area formed by current solution, global best solution and randomly selected elite solution, which is beneficial to augment the search area and prevent premature convergence. Moreover, the probability of crossover is controlled dynamically according to the successful search experience, which can enable ABC-BB to adapt all kinds of optimization problems with different landscapes. The experimental results show that the proposed algorithm is significantly better than ABC-BB as well as several recently-developed variants of PSO, DE and ABC.

The rest of this paper is organized as follows. In section 2, the related works are briefly reviewed, including the basic ABC and some improved ABCs. In section 3, we present the proposed approach in detail. Section 4 presents and discusses the experimental results. Finally, the conclusion is drawn in section 5.

2. Related Works

2.1. Basic ABC

The ABC algorithm is a swarm-based stochastic optimization method, which is inspired by the waggle dance and intelligent foraging behavior of honey bees. In ABC, the position of a food source denotes a possible solution of the optimization problem, and the nectar amount of each food source denotes the quality (fitness) of the corresponding solution. In order to find the best food source, three different types of bees, i.e., employed bee, onlooker bee and scout bee, are responsible for the different search tasks. Firstly, the first half of the colony consists of employed bees, which are responsible for randomly searching better food source in the neighborhood of the corresponding parent food source and then passing information of the food sources to onlooker bees. Secondly, the second half of the colony is composed of onlooker bees, which further search the better food sources according to the information provided by employed bees. Thirdly, if the quality of a food source is not improved by a preset number of times (*limit*), this food source is abandoned by its employed bee, and then this employed bee becomes a scout bee that begins to seek a new random food source. The original ABC algorithm includes four phases, i.e., initialization phase, employed bee phase, onlooker bee phase and scout bee phase. After the initialization phase, ABC turns into a loop of employed bee phase, onlooker bee phase and scout bee phase until the termination condition is satisfied. The details of each phase are described as follows.

Initialization phase: Similar to other EAs, ABC also starts with an initial population of SN randomly generated food sources. Each food source x_i=($x_{i,1}$,$x_{i,2}$,...,$x_{i,D}$) are generated randomly according to Eq.(1).

$$x_{i,j} = a_j + rand\,(0,1)\,(b_j - a_j) \tag{1}$$

where *SN* is the number of food sources, and *SN* is equal to the number of employed and onlooker bees; *D* is the dimensionality (variables) of the search space; a_j and b_j are the lower and upper bounds of the *j*th variable respectively; $rand(0,1)$ is a random real number in range of [0,1]. Then, the fitness values of the food sources are calculated by Eq.(2).

$$fit_i = \begin{cases} 1/(1+f_i), f_i \geq 0 \\ 1+|f_i|\,, f_i < 0 \end{cases} \tag{2}$$

Employed bee phase :In this phase, each employed bee generates a new food source $v_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,D})$ in the neighborhood of its parent position $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ using Eq.(3).

$$v_{i,j} = x_{i,j} + \phi_{i,j}(x_{i,j} - x_{k,j}) \tag{3}$$

where $k \in \{1,2,...,SN\}$ and $j \in \{1,2,...D\}$,are randomly chosen indexes; k has to be different from i;$\phi_{i,j}$is a random number in the range [-1,1]. If the new food source v_i is better than its parent x_i, then x_i is replaced with v_i.

Onlooker bee phase:After receiving the information from the employed bees, the onlooker bees begin to select food sources for exploitation. The probability of selecting a

food source depends on its nectar amount, which can be calculated as follows.

$$p_i = fit_i / \sum_{i=1}^{SN} fit_i \tag{4}$$

Where p_i is the selection probability and fit_i is the fitness value of the ith food source. Once an onlooker bee completes its selection, it would also produce a modification on its chosen food source using Eq.(3). As in the case of the employed bees, the greedy selection method is employed to retain a better one from the old food source and the modified food source as well.

Scout bee phase:If a position x_i cannot be improved for at least $limit$ times, then that food source is assumed to be abandoned. Then, the scout produces a food source randomly as in Eq.(1) to replace x_i.

2.2. Improved ABCs

The solution search equation plays an important role in ABC. Eq.(3) indicates that the solution search equation of ABC is good at exploration but poor at exploitation. To improve the exploitation ability and utilize the beneficial information of the current best solution, Zhu and Kwong [49] proposed a new search equation (names GABC), shown in Eq.(5).

$$v_{i,j} = x_{i,j} + \phi_{i,j}.(x_{i,j} - x_{k,j}) + \psi_{i,j}.(x_{best,j} - x_{i,j}) \tag{5}$$

Where the third term in the right-hand side of Eq.(5) is a new added term called gbest term, $x_{best,j}$ is the jth element of the global best solution, is a uniform random number within [0,C], and C is a nonnegative constant and is suggested to set to 1.5. Although this new search equation has been shown superior to the original one, its mechanism of utilizing the gbest can still cause inefficiency to the search ability of the algorithm and slow down convergence speed. Because the guidance of the last two terms may be in opposite directions, and this can cause an oscillation phenomenon [25]. Therefore, in order to address these issues in Eq.(5), Zhou et al. [48] designed a Gaussian bare-bones search equation inspired by the concept of BBPSO, shown in Eq.(6).

$$v_{i,j} = \begin{cases} N((x_{i,j} + x_{best,j})/2, |x_{i,j} - x_{best,j}|), if\ rand_j \leq CR \\ x_{i,j}, otherwise \end{cases} \tag{6}$$

Where $N(.)$ represents a Gaussian distribution with mean and variance ; x_{best} is the global best solution of current population, $rand_j$ is a uniformly distributed random number within the range [0,1]; CR is a new introduced parameter which controls how many elements in expectation can be derived from its parent x_i for v_i. Since there is only one dimension of x_i to be updated for v_i in the original search equation, the introduction of CR is helpful to inherit more information from x_{best} to enhance the exploitation. In Eq.(6), new candidate solutions are generated in the search space formed by the current solution x_i and the global best solution x_{best}. Compared with the original search equation Eq.(3), and the Eq.(5), the Gaussian bare-bones search equation Eq.(6) has two advantages. First, Eq.(6) takes advantages of the global best solution to guide the search of new candidate solutions, which is beneficial to enhance the exploitation ability of ABC. Second, the oscillation phenomenon can be avoided because Eq.(6) can be considered as one

single term. To better balance the exploration and exploitation, Eq.(6) is only used in the onlooker bee phase, while the employed bee phase still use the original solution search equation Eq.(3) for generating new candidate solutions. By adopting the new equation Eq.(6) in the onlooker bee phase, the newly proposed algorithm is named ABC-BB. Moreover, Zhou [48] also proposed a general opposition learning strategy (GOBL) in the scout bee phase. By combining the ABC-BB and GOBL, they [48] proposed a new ensemble algorithm (GBABC). Similarly, Cui et al. [15] also proposed a novel search equation to avoid the oscillation phenomenon of GABC, shown in Eq.(7).

$$v_{i,j} = \begin{cases} x_{i,j} + \phi_{i,j}.(x_{i,j} - x_{k,j}), if\ rand < P \\ x_{i,j} + \psi_{i,j}.(x_{best,j} - x_{i,j}), otherwise \end{cases} \tag{7}$$

Where P is a parameter defined by the user. x_{best} is the current best solution of the population, and x_i is the ith food source. x_k is a randomly selected food source from the population, which is different from x_i. is the uniformly distributed random number in the range of [-1, 1] and is a uniform random number in the range of [0, 1.5]. j is randomly selected from $1, 2, \ldots, D$. Obviously, with the guidance from only one term, the novel search strategy can easily avoid the oscillation phenomenon. Moreover, parameter P could be used to control how to appropriately exploit the valuable information of the current best solution. In 2015, Gao et al. [23] also proposed a kind of Gaussian search equation, shown in Eq.(8).

$$v_{i,j} = N((x_{k,j} + x_{best,j})/2, |x_{k,j} - x_{best,j}|) \tag{8}$$

Where k is a randomly chosen index from 1,2,*SN* with the constraint that $k \neq i$, the meaning of the other symbol is the same as Eq.(7). Gao et al. [23] proposed a novel ensemble ABC variants by combining Eq.(8), CABC [25], parameter adaption strategy and fitness-based neighborhood mechanism, named BCABC. The experimental results show that the performance of BCABC is better than some newly proposed state-of-the-art algorithms.

3. The Proposed Algorithm

In this section, we will firstly propose a novel enhanced ABC with triangle search strategy and parameter strategy, then the pseudo code of the novel ABC variants will be given.

3.1. The New Triangle Search Strategy

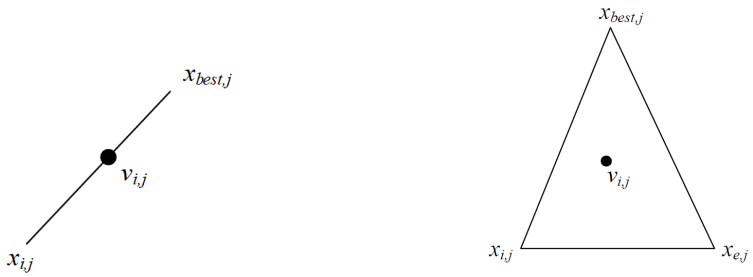
In the search equation Eq.(6) of ABC-BB, the new candidate $v_{i,j}$ is generated in the line determined by the current solution $x_{i,j}$ and the gbest $x_{best,j}$ solution. Fig.1(a) demonstrate the evolution process of Eq.(6). However, we can see from Fig.1(a) that there is only one guiding individual (gbest) in the search equation Eq.(6), which is easy to result in the problem of premature convergence and locally convergence because the population is always guided by the global best solution monotonically. In order to overcome this drawback, a new search equation based on triangle search strategy is proposed in this paper, shown in Eq.(9).

$$v_{i,j} = \begin{cases} N(\frac{x_{i,j}+x_{best,j}+x_{e,j}}{3}, \frac{|x_{i,j}-x_{best,j}|+|x_{best,j}-x_{e,j}|+|x_{e,j}-x_{i,j}|}{3}), & if\ rand_j \leq CR \\ x_{i,j}, otherwise \end{cases} \tag{9}$$

Where CR is the crossover probability, the meaning of $N(.)$ and x_{best} are the same as Eq.(6). x_e is a randomly chosen from the elite solutions (the top $p \cdot SN$solutions in current population, $p \in (0, 1)$. In Eq.(9), the candidate solution $v_{i,j}$ is generated among the triangle area determined by the current solution, the global best solution and a randomly selected elite solution, which is beneficial to augment the search area, increasing information sharing and thereby prevent premature convergence. The search equation of Eq.(9) is given in Fig.1(b).

Li et at.[35] indicated that "the more information is efficiently utilized to guide the flying, the better performance the PSO algorithm will have". Now that Eq.(9) uses more elite solutions to guide the search, we have reason to believe that the novel search equation will decrease the premature problem of ABC-BB and enhance the performance of ABC-BB.

In EABC-BB, the mutation strategy uses more information of elite solutions. Because of its randomness characteristic, it may hold down the premature problem of ABC-BB and enable ABC-BB have more chance to explore more peaks denoted by different elite solutions.



(a) ABC-BB (b) EABC-BB

Fig. 1. Evolution process of a solution according to (a):Eq.(6) and (b):Eq.(9) in 2-D parametric space.

3.2. Parameter Adaption Strategy

Like other DE variants, the parameter CR also greatly affects the performance of ABC-BB [48]. The experimental results of literature [48] show that a higher value of CR is more suitable for solving unimodal problems, while a lower one is better for multimodal problems. Therefore, it is difficult to determine the optimal control parameters because they are problem dependent. In this paper, a simple self-adaptive strategy is proposed to dynamically update CR. In some well-known self-adaptive DE algorithms, such as SaDE [41] and JADE [47], the initial value of CR is independently generated by a normal distribution of mean 0.5 and standard deviation 0.1. After a predefined number of generations, the CR is updated according to the search experiences of successful crossover

probabilities. By following this idea, a new self-adaptive CR strategy is proposed as follows:

$$CR = N(\bar{S}_{CR}, 0.1) \tag{10}$$

Where $N(\bar{S}_{CR}, 0.1)$ represents a normal distribution whose mean and standard deviation are $\bar{S}_{CR}$ and 0.1, respectively. $\bar{S}_{CR}$ is set to be 0.3 considering that it is suitable for CR in literature [48], then the value will be adjusted dynamically at the end of each generation as follows:

$$\bar{S}_{CR} = \text{mean}(S_{CR}) \tag{11}$$

where mean(.) is the general arithmetic average; and S_{CR} is the set of all successful *CR* at generation g.

By combining the novel search equation Eq.(9) and the adaptive parameter adaption strategy, a novel ABC variants, named enhanced ABC-BB (EABC-BB for short), is proposed. The pseudo code of EABC-BB is described in Algorithm 1, where *FE*s is the number of consumed fitness function evaluations, and max_*FE*s, as the stopping criterion, is the maximal number of fitness function evaluations. $trial_i$ records the unchanged times of x_i's fitness value.

Clearly, Compared to ABC-BB, EABC-BB only modifies the search equation and the parameter adaption strategy. Therefore, EABC-BB does not increase the ABC-BB's complexity any more, both EABC-BB and the ABC-BB have the same time complexity O(g_{max}· *SN*· *D*), where g_{max} is the maximum number of generations.

Algorithm 1:Pseudo code of the proposed EABC-BB

Initialization:Generate SN solutions that contain D variables according to Eq.(1), *FEs*=0
while $FEs < max\_FEs$
  for i=1 to SN //employed bee phase
    Generate a new solution v_i using Eq.(3), then calculate $f(v_i)$
    if $f(v_i) < f(x_i)$
      Replace x_i by v_i and $trial_i$=0
    else
      $trial_i$=$trial_i$ +1
    end if
  end for //end employed bee phase
    Select the top $p.SN$ solutions as elite solutions from population
  for i=1 to SN //onlooker bee phase
      Select a solution x_e from elite solutions randomly to search
      Generate a new candidate solution v_e in the neighborhood of x_e using Eq.(9) and calculate $f(v_e)$
    if $f(v_e) < f(x_e)$
      Replace x_e by v_e and $trial_i$=0
    else
      $trial_e$=$trial_e$ +1
    end if
  end for //end onlooker bee phase
    $FEs = FEs + SN \times 2$ Select the solution x_{max} with maximum $trial$ value //Scout bee phase
  if $trial$(max)>$limit$ //only one food source with max $trial$ value can be initialized
      Replace x_{max} by a new solution generated according to Eq.(1), FEs=FEs+1, $trial$ (max)=0;
  end if //end scout bee phase
end while
Output the food source with the smallest objective value

4. Computational Study

In this section, the experimental results will be given. Specifically, in section 4.1, the benchmark functions will be given, then in section 4.2, the new parameter p will be analyzed. In section 4.3, the proposed EABC-BB will be compared with other recently-proposed ABC variants. The section 4.4 give the comparisons between the EABC-BB and some newly proposed state-of-the-art EAs. At last, the new EABC-BB will be applied to solve two well-known engineering optimization problems.

4.1. Benchmark Functions

In this section, 23 well-known benchmark functions [48] with different dimensions (D=30 and D=50, respectively) are employed to validate the performance of EABC-BB. The 13 test functions are scalable problems. Among these problems, F01-F04 are unimodal func-tions, and F05 is the Rosenbrock function which is multimodal when D$>$3. F06 is a step function which has one minimum and is discontinuous, while F07 is a noisy quartic function. F08-F13 are multimodal functions with many local minima. The remaining 10 functions (F14-F23) are shifted and/or rotated types taken from the CEC 2005 competition[44]. The definitions of these functions are given in [48].

4.2. Adjusting the Parameter p

In EABC-BB, only one parameter p is used whose value may affect the performance. Therefore, in the subsection, we investigate different p values to select the best one for maximizing the performance of EABC-BB. The available p values are in the range [0.05,0.2] in steps of 0.05, i.e., there are four choices for p in total. F01-F13 are selected as test functions. All the selected 13 functions are tested at D=30, the maximal number of fitness function evaluations (*max_FEs*) is set to 5,000·D. The number of food sources *SN* and *limit* are set at 30 and 100, respectively. Each function is run 30 times, and the mean error ($f(x)$-$f(x^*)$), x^* is the global optimum) and standard deviation values are recorded. The experimental results are listed in Table 1. In the last line of Table 1, in order to compare the significance between two parameters values, the Wilcoxon singed-rank test [17]is conducted. The Wilcoxon signed-rank test is a non-parametric statistic hypothesis test, which can be used as an alternative to the paired t test when the results cannot be assumed to be normally distributed. "†", "‡" and "≈" indicate the EABC-BB with p=0.1 is better than, worse than, and similar to the EABC-BB with other p values. As seen from Table 1, the EABC-BB at p=0.1 performs best, while a higher or lower p value will weaken the performance of EABC-BB. Therefore, the parameter p is set at 0.1 in the following all experiments.

4.3. Comparison with Other ABCs

This subsection presents a comparative study of EABC-BB with MGABC [15], BC-ABC [23], ABC-BB [48] at both D=30 and 50. These three algorithms are all proposed recently and have relatively good performance according to their reports. Moreover, each of the three algorithms have proposed a kind of improved search equations, respectively.

Table 1. The impact of different p values on the EABC-BB performance for 13 test functions

Func	p=0.05	p=0.15	p=0.2	p=0.1
	Mean error(std dev)	Mean error(std dev)	Mean error(std dev)	Mean error(std dev)
F01	1.45E-34(2.51E-34)†	1.69E-76(2.92E-76)†	2.79E-76(3.78E-76)†	4.66E-81(3.29E-80)
F02	4.86E-11(7.39E-11)†	1.87E-56(3.25E-56)‡	8.02E-48(1.38E-47)‡	1.69E-41(1.08E-40)
F03	2.05E+02(1.64E+02)†	1.88E+03(1.79E+03)†	2.63E+03(6.39E+02)†	1.15E+02(2.16E+02)
F04	2.25E+01(1.33E+02)†	5.36E-03(6.73E-03)‡	3.71E-03(2.37E-03)‡	6.40E-01(9.95E-01)
F05	1.16E+02(7.20E+01)†	1.75E+00(2.36E+00)‡	1.12E+01(4.49E+00)‡	1.52E+01(4.44E+00)
F06	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)
F07	5.13E-03(1.25E-03)†	8.71E-03(3.10E-03)†	8.65E-03(2.42E-03)†	2.74E-03(3.56E-03)
F08	3.94E+01(6.83E+01)†	3.94E+01(6.83E+01)†	1.33E-02(9.96E-09)†	3.82E-04(6.84E-12)
F09	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)
F10	3.04E-14(2.14E-14)†	7.99E-15(0.00E+00)†	7.99E-15(0.00E+00)†	3.39E-15(4.24E-15)
F11	7.37E-03(1.27E-02)†	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)$^{\approx}$	0.00E+00(0.00E+00)
F12	3.32E-26(5.74E-26)†	1.57E-32(4.53E-32)†	1.57E-32(7.31E-32)†	6.28E-33(7.77E-33)
F13	6.97E-32(1.18E-31)†	1.50E-33(5.24E-34)†	1.50E-33(2.38E-33)†	5.99E-34(7.42E-34)
$\dagger\ddagger\approx$	11/0/2	7/3/3	7/3/3	−

Note that the GBABC proposed in literature [48] is added an additional initialization method (General Opposite Based Learning, GOBL) at ABC-BB, this is unfair to the other algorithms because they are all randomly initialized. Therefore, We only use the ABC-BB for the comparison.

For a fair comparison, all the competitive algorithms have the same parameter settings, i.e., SN=30, $limit$=100, Max_FEs=5000·D for F01-F13 and max_FEs=10,000·D for F14-F23.For other specific parameters, C=1.5 and P=0.3 for MGABC, CR=0.3 for ABC-BB, which are the same as in the original literatures. Each algorithm is run 30 times per function, and the mean error and standard deviation values are presented in Tables 2 and 3 for D=30 and 50, respectively. Moreover, to compare the significance between two algorithms, the paired Wilcoxon signed-rank[17] test is used. "†", "‡" and "≈" indicated EABC-BB is better than ,worse than, and similar to its competitor according to the Wilcoxon signed-ranked test at α=0.05.

For D=30, from the results presented in Table 2, it is clear that EABC-BB performs significantly better than the other three competitors on the majority of test functions. To be specific, EABC-BB outperforms MGABC on 17 out of 23 test functions, while MGABC only achieves better result on the F05 (Rosenbrock) and F19 (Shifted Rosenbrock). As F05 and F19 are Rosenbrock and shifed Rosenbrock functions and their global optimum is inside a long, narrow, parabolic shaped flat valley, the variables are strongly dependent, and the gradients do not generally point towards the optimum. If the population is guided by the global best solution or some other good solutions, the search will fall into some unpromising areas[14]. Therefore, EABC-BB is beaten by MGABC at these two functions. As a multi strategies ensemble algorithm, BCABC also performs better than EABC-BB on functions F01, F05, F16 and F19, while EABC-BB beats it on the other 16 functions. For ABC-BB, EABC-BB wins on16 functions and only loses on function F04 and F19.

Table 2. The results achieved by MGABC, BCABC, ABC-BB and EABC-BB at D=30 (Mean error±std dev)

Func.	MGABC	BCABC	ABC-BB	EABC-BB
F01	1.66E-62±4.05E-62†	3.46E-85±7.21E-84‡	4.89E-48±2.28E-48†	4.66E-81±3.29E-80
F02	2.43E-32±1.48E-32†	3.92E-36±5.81E-37†	2.36E-29±6.17E-29†	1.69E-41±1.08E-40
F03	5.36E+03±1.20E+03†	2.17E+03±1.29E+03†	3.51E+03±2.19E+02†	1.15E+02±2.16E+02
F04	1.06E+00±2.37E-01†	1.55E+00±3.62E+00†	2.32E-02±5.17E-03‡	6.40E-01±9.95E-01
F05	5.95E-01±1.02E+00‡	5.30E+00±3.17E+00‡	2.15E+01±3.66E+00†	1.52E+01±4.44E+00
F06	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00
F07	2.37E-02±5.24E-03†	1.01E-02±8.83E-02†	1.84E-02±8.21E-03†	2.74E-03±3.56E-03
F08	3.82E-04±2.64E+01$^{\approx}$	3.82E-04±4.83E-12$^{\approx}$	3.82E-04±5.75E-12$^{\approx}$	3.82E-04±6.84E-12
F09	0.00E+00±0.00E+00$^{\approx}$	2.43E-12±6.17E-13†	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00
F10	3.67E-14±3.43E-15†	7.72E-14±5.85E-14†	1.46E-14±2.59E-15†	3.39E-15±4.24E-15
F11	0.00E+00±0.00E+00†	6.79E-12±3.27E-12†	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00
F12	1.57E-32±2.80E-48†	1.57E-32±8.38E-38†	1.57E-32±7.09E-46†	6.28E-33±7.77E-33
F13	1.49E-33±2.87E-33†	1.35E-32±8.13E-32†	1.35E-32±7.96E-48†	5.99E-34±7.42E-34
F14	5.96E-14±1.27E-14†	5.68E-14±2.73E-14†	6.82E-14±2.77E-14†	4.11E-14±1.72E-14
F15	6.47E+03±1.39E+03†	6.17E+03±3.66E+03†	3.67E+02±2.26E+02†	9.14E-02±2.96E-01
F16	8.88E+06±2.09E+06†	1.54E+06±5.23E+06‡	6.11E+06±1.33E+06†	4.83E+06±4.89E+06
F17	3.29E+04±4.94E+03†	7.52E+03±4.81E+03†	4.92E+03±2.92E+03†	3.14E+03±1.14E+03
F18	8.38E+03±1.73E+03†	2.73E+03±4.63E+02†	2.04E+03±6.73E+02†	1.86E+03±8.12E+02
F19	1.36E+01±2.14E+01‡	7.42E+01±3.99E+01‡	6.42E+01±8.17E+01‡	8.19E+01±8.35E+01
F20	4.91E-01±2.83E-01†	2.20E-02±3.89E-02†	2.32E-02±3.57E-02†	2.11E-02±5.64E-02
F21	2.09E+01±4.80E-02$^{\approx}$	2.09E+01±3.37E-02$^{\approx}$	2.09E+01±6.86E-02$^{\approx}$	2.09E+01±5.59E-02
F22	5.40E-13±1.27E-14†	6.82E-13±4.16E-13†	7.57E-14±3.62E-14†	5.22E-14±1.55E-14
F23	2.14E+02±2.30E+01†	1.12E+02±4.82E+01†	1.22E+02±3.46E+01†	1.07E+02±2.13E+01
$\dagger/\ddagger/\approx$	17/2/4	16/4/3	16/2/5	-

In order to investigate the scalability of EABC-BB, we also compare EABC-BB with all the competitors on the 23 test functions with 50D. The experimental results are given in Table 3. As seen from Table 3, EABC-BB consistently gets significantly better results than its competitors. Overall, EABC-BB outperforms MGABC, BCABC and ABC-BB on 15, 18 and 15 out of 23 functions. As can be seen from Tables 2 and 3, it is clear that EABC-BB is the best algorithm among four algorithms.

4.4. Comparison with Some Newly Proposed EAs

To further investigate the performance of EABC-BB, we compare EABC-BB with four newly proposed EAs, including DE and PSO variants and an ensemble algorithm GBABC:

- sinusoidal differential evolution algorithm (sinDE) (Draa et al. 2015) [19]
- Gaussian bare-bones artificial bee colony (GBABC) (Zhou et al. 2016), note that GBABC=ABC-BB+GOBL [48].
- Self regulating particle swarm optimization algorithm (SRPSO) (Tanweer et al. 2015) [45]
- Social Learning Particle Swarm Optimization (SLPSO) (Cheng et al. 2015) [10]

For a fair comparison, the control parameters of four competitive EAs are set to the suggested values offered by their corresponding literatures.

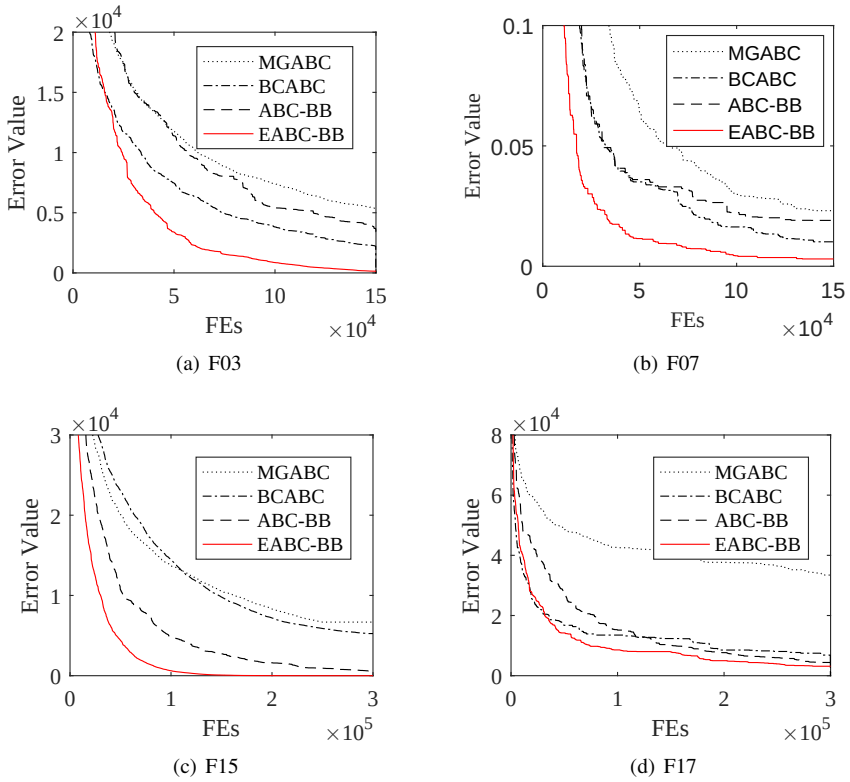


Fig. 2. Convergence performance of different ABCs on 4 functions

The stopping criterion is the same as previous subsections. Each algorithm is run 30 times per function, and the mean error and standard deviation values are given in Table 5. From Table 5, we can see that EABC-BB obtains significantly better results on the majority of test functions compared with other competitors. To be specific, sinDE wins on 5 test functions compared with EABC-BB, but on the other 17 test functions EABC-BB performs better.

SRPSO adopt self regulation strategy to control the premature convergence problem, which is similar to EABC-BB. Therefore, SRPSO has excellent performance. But in contrast to EABC-BB, SRPSO outperforms EABC-BB on only 6 test functions, but loses on 15 functions. SLPSO performs better than EABC-BB on only 2 functions, while EABC-BB outperforms it on other 21 test functions. GBABC is composed of ABC-BB and GOBL strategy, it outperforms EABC-BB on 5 functions and loses on 14 test functions compared with EABC-BB.

Overall, it is clear that EABC-BB is the best algorithm among 5 newly improved algorithms.

The convergence curves of MGABC, BCABC, ABC-BB and EABC-BB on four representative functions are shown in Fig.2. As seen from Fig.2, EABC-BB has faster convergence ability than other algorithms. The reason why EABC-BB have an advantage over

Table 3. The results achieved by MGABC,BCABC,ABC-BB and EABC-BB at *D*=50 (Mean error±std dev)

Func.	MGABC	BCABC	ABC-BB	EABC-BB
F01	3.22E-61± 2.25E-61†	5.85E-09±1.17E-08†	3.86E-41±4.51E-41†	6.84E-63± 1.53E-62
F02	2.84E-31± 1.63E-31‡	1.70E-09±3.41E-09†	2.87E-26±1.01E-26†	1.20E-29± 1.64E-29
F03	2.37E+04± 7.16E+02†	1.70E-09±3.58E+03†	3.53E+04±3.11E+03†	2.14E+04± 9.16E+03
F04	2.28E+01± 4.08E+00†	1.09E+01±6.97E-01†	1.25E+00±6.47E-01†	1.22E+00± 5.98E-01
F05	1.25E+00±1.92E+00‡	1.90E+01±3.42E+01‡	9.60E+01±4.94E+01†	5.32E+01± 2.78E+01
F06	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00
F07	0.26E+00± 4.34E-02†	4.37E-02±2.35E-02†	3.82E-02±6.45E-03†	1.43E-02± 1.68E-03
F08	2.27E-11± 3.48E-12†	2.09E+04±1.83E-10†	2.10E+04±6.83E+01†	0.00E+00± 0.00E+00
F09	0.00E+00±0.00E+00$^{\approx}$	6.13E-04±1.10E-03†	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±0.00E+00
F10	7.37E-14± 4.58E-15†	5.02E-06±1.00E-05†	3.01E-14±1.77E-15†	1.86E-14± 3.55E-15
F11	0.00E+00±0.00E+00$^{\approx}$	6.36E-03±5.70E-03†	0.00E+00±0.00E+00$^{\approx}$	0.00E+00±6.29E-03
F12	9.42E-33±0.00E+00†	5.49E-11±1.09E-10†	9.42E-33±0.00E+00†	4.99E-33±1.08E-33
F13	1.50E-33±0.00E+00$^{\approx}$	4.58E-33±5.37E-33†	1.50E-33±0.00E+00$^{\approx}$	1.50E-33±0.00E+00
F14	1.56E-13± 2.84E-14†	5.80E-09±1.16E-08†	1.27E-13±2.84E-14†	5.68E-14±0.00E+00
F15	2.97E+04± 3.60E+02†	2.63E+04±3.55E+03†	1.93E+04±4.60E+03†	6.36E+03±6.08E+03
F16	3.07E+07± 3.83E+06‡	3.19E+07±2.88E+06‡	3.20E+07±7.64E+06‡	4.00E+07±1.02E+07
F17	1.16E+05± 1.03E+04†	1.15E+05±1.41E+04†	4.96E+04±5.55E+03‡	5.61E+04±2.95E+03
F18	2.07E+04± 1.28E+03†	2.02E+04±9.02E+02†	6.88E+03±2.38E+03‡	7.49E+03±4.9E+03
F19	4.67E+00± 6.93E+00‡	4.12E+01±5.38E+01‡	6.85E+01±7.17E+01‡	8.04E+01±8.63E+00
F20	5.52E-01± 2.39E-01†	3.95E-02±4.29E-02†	5.61E-06±7.78E-06†	1.52E-02±1.74E-02
F21	2.11E+01± 2.39E-02†	2.11E+01±2.33E-02†	2.11E+01±2.89E-02†	2.11E+01±2.35E-02
F22	1.13E-13±0.00E+00†	4.79E-09±7.37E-09†	1.27E-13±2.84E-14†	7.95E-14±3.11E-14
F23	6.72E+02± 8.41E+01†	2.73E+02±3.91E+01‡	3.08E+02±2.20E+01†	2.75E+02±2.51E+01
$^{\dagger}/^{\ddagger}/^{\approx}$	15/4/4	18/4/1	15/4/4	–

Table 4. Parameter settings for sinDE, SRPSO, SLPSO, GBABC and EABC-BB

Algorithms	Parameter settings
sinDE	$NP = 30, freq = 0.25$
SRPSO	$N = 30, w_{initial} = 1.05, w_{final} = 0.5, c_1 = c_2 = 1.49445, V_{max} = 0.06078 * Range$
SLPSO	$M = 30$
GBABC	$SN = 30, limit = 100, CR = 0.3$
EABC-BB	$SN = 30, limit = 100, p = 0.1$

other algorithms can be found from the novel search equation Eq.(9) and the parameter adaption strategy (see Eq.(10)). By contrast the mean value of the normal distribution of Eq.(9) with that of Eq.(6), we can found that the ratio of the current solution $x_{i,j}$ is 1/2 in Eq.(6) because there are only two solutions ($x_i$and the global best solution x_{best}), but the ratio of $x_{i,j}$ in Eq.(9) is 1/3 because there are three solutions (x_i, the elite solution x_e and the global best solution x_{best}). That is to say, the EABC-BB is guide by more elite solutions and has stronger exploitation ability. Meanwhile, because different elite solutions have equal chance to be a leader, thus the premature convergence phenomenon in ABC-BB can be effectively avoided and the novel search equation can enable EABC-BB explore different function peaks denoted by different elite solutions. Moreover, the parameter *CR* adaption strategy can also enhance the performance of ABC-BB.

Table 5. The comparison results between EABC-BB and other EAs at D=30

Func.	index	sinDE	SRPSO	SLPSO	GBABC	EABC-BB
F01	Mean error	3.07e-53†	3.08e-74†	3.60e-37†	2.63e-49†	4.66E-81
	std dev	8.70E-53	1.30E-75	8.31E-37	5.27e-49	3.29E-80
F02	Mean error	1.72e-38†	8.23e-48‡	1.80e-16†	2.53e-31†	1.69E-41
	std dev	4.00E-38	3.87E-48	3.14E-16	2.69e-31	1.08E-40
F03	Mean error	1.37e+03†	3.59e+00‡	1.38e+03†	4.48e+01‡	1.15E+02
	std dev	4.30E+02	3.44E+00	3.13E+03	7.18E+01	2.16E+02
F04	Mean error	3.50e+00†	4.22e-02‡	1.39e+01†	1.39e-07‡	6.40E-01
	std dev	3.81E+00	3.05E-02	9.32E+01	1.88e-07	9.95E-01
F05	Mean error	5.19e+01†	3.00e+01†	3.50e+01†	3.63e+00†	1.52E+01
	std dev	2.05E+01	3.15E+01	2.60E+01	3.40E+00	4.44E+00
F06	Mean error	0.00e+00≈	0.00e+00≈	2.75e-06†	0.00e+00≈	0.00E+00
	std dev	0.00E+00	0.00E+00	2.25E-06	0.00E+00	0.00E+00
F07	Mean error	3.97e-03†	8.93e-03†	3.34e-02†	9.82e-05†	2.74E-03
	std dev	9.98E-04	2.08E-03	8.66E-03	7.06e-05	3.56E-03
F08	Mean error	4.47e+02†	9.93e+02†	3.95e+02†	3.82e-04†	3.82E-04
	std dev	3.18E+02	2.72E+02	5.57E+02	4.54e-13	6.84E-12
F09	Mean error	1.56e+01†	3.06e+01†	2.02e+01†	0.00e+00≈	0.00E+00
	std dev	3.01E+00	4.52E+00	1.80E+01	0.00E+00	0.00E+00
F10	Mean error	7.99e-15†	3.07e-14†	9.00e-14†	4.44e-16‡	3.39E-15
	std dev	0.00E+00	3.17E-15	8.89E-14	0.00E+00	4.24E-15
F11	Mean error	2.44e-50†	6.89e-03†	3.69e-16†	0.00e+00≈	0.00E+00
	std dev	6.88E-50	1.00E-02	6.96E-14	0.00E+00	0.00E+00
F12	Mean error	1.85e-34‡	2.07e-32†	4.33e-13†	1.57e-32†	6.28E-33
	std dev	5.23E-34	4.63E-02	6.40E-13	5.47e-48	7.77E-33
F13	Meanerror	3.15e+02†	5.49e-32†	2.10e-33†	1.35e-32†	5.99E-34
	std dev	1.26E+02	6.45E-32	3.64E-33	5.47e-48	7.42E-34
F14	Mean error	3.29e+03†	6.13e-13†	1.49e-14†	6.44e-14†	4.11E-14
	std dev	1.88E+03	4.92E-14	3.35E-14	1.93e-14	1.72E-14
F15	Mean error	2.06e+03†	4.42e-04‡	3.31e+03†	2.94e+02†	9.14E-02
	std dev	4.43E+02	5.82E-04	1.85E+03	1.11E+02	2.96E-01
F16	Meanerror	5.18e+01‡	5.37e+06†	1.57e+07†	2.80e+06‡	4.83E+06
	std dev	2.28E+01	4.18E+06	1.14E+07	1.02E+06	4.89E+06
F17	Mean error	3.70e+03†	5.39e+01‡	6.52e+03†	7.38e+03†	3.14E+03
	std dev	4.86E-13	4.22E+01	2.73E+03	1.46E+03	1.14E+03
F18	Mean error	2.11e+01‡	3.39e+03†	7.26e+03†	2.43e+03†	1.86E+03
	std dev	3.10E-02	7.87E+02	1.89E+03	4.72E+02	8.12E+02
F19	Mean error	7.85e+00‡	2.72e+01‡	4.62e+02†	3.64e+01‡	8.19E+01
	std dev	2.50E+00	2.80E+01	3.31E+02	4.94E+01	8.35E+01
F20	Mean error	5.56e+01†	1.87e-02†	2.41e+03†	1.86e-02†	1.61E-02
	std dev	1.22E+01	1.14E-02	3.21E+03	1.63e-02	5.64E-02
F21	Mean error	6.60e+01†	2.09e+01≈	2.05e+01‡	2.09e+01≈	2.09E+01
	std dev	3.32E+01	3.34E-02	1.18E+01	8.62e-02	5.59E-02
F22	Mean error	5.19e+04†	1.65e+01†	5.50e+01†	6.63e-14†	5.22E-14
	std dev	3.66E+04	1.93E+00	1.56E+01	2.12e-14	1.55E-14
F23	Mean error	5.63e+00‡	1.51e+02†	6.81e+01‡	1.28e+02†	1.07E+02
	std dev	6.58E-01	3.60E+01	8.08E+00	1.77E+01	2.13E+01
†/‡/≈	–	17/5/1	15/6/2	21/2/0	14/5/4	–

4.5. Application to Two Real Optimization Problems

This section is about the performance evaluation of EABC-BB algorithm on the two well-known non-linear optimization problems. The explanations of the two problems and the obtained results via EABC-BB are given in the following sub-section in details.

Pressure Vessel Design (PVD) Problem This benchmark engineering problem is an engineering optimization problem. In this problem,the total cost, including a combination of welding cost, material and forming cost, is to be minimized. The welded beam is fixed and designed to support a load. The involved variables are the thickness $Ts(x_1)$, thickness of the head $Th(x_2)$, the inner radius $R(x_3)$, and the length of the cylindrical section of the vessel $L(x_4)$.The problem can be expressed as follows:

$$x = (x_1, x_2, x_3, x_4)^T := (T_s, T_h, R, L)^T \tag{12}$$

$$minf(x) = 0.6224x_1x_2x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{13}$$

$$g_1(x) = 0.019x_3 - x_1 \leq 0 \tag{14}$$

$$g_2(x) = 0.00954x_3 - x_2 \leq 0 \tag{15}$$

$$g_3(x) = 1,296,000 - \pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 \leq 0 \tag{16}$$

$$g_4(x) = x_4 - 240 \leq 0 \tag{17}$$

$$x \in R^4 : (0, 0, 10, 10)^T \leq x \leq (99, 99, 200, 200)^T \tag{18}$$

As can be seen, the PVD problem is a constraint problem with four constraints, i.e., g_1, g_2, g_3 and g_4. As the ABC algorithm originally developed for unconstrained continuous global optimization problems, it also requires a constraint handling method while dealing with the constrained global optimization problems. This paper adopts the penalty function method introduced in literature [5] to transfer the constraint problem to a unconstrained problem.

The parameters setting are the same as literature [5], i.e.: the maximum function evaluations (*Max_FEs*) is set at 500,000,the population size (*SN*) is set at 100.Finally, the results were obtained after running the EABC-BB algorithm 20 times for each problem. The experimental results are given in Table 6. From Table 6, it is clear that EABC-BB is the best algorithms to solve the PVD problem, since the EABC-BB algorithm has the ability to produce better results than the previous literature with all of the aforementioned constraint handling methods. It's worthy to note that DGABC is another DE and ABC hybrid algorithm, which is second to EABC-BB in PVD problem. Similar to GBABC, in DGABC a chaotic opposition-based population initialization method is employed, but to make a fair comparison, all algorithms except for GBABC adopt random initialization strategy.

Because GBABC has compared with some state-of-the-art EAs [48], the comparison with GBABC can be looked on as the comparison with these EAs indirectly.

According to Table 6, the best solution obtained by EAB-BB is *x*=[0.77817354 0.3847-4404 40.31987228 199.99647520], its constraints are $g_1(x)$=-4.996000058099526E-09, $g_2(x)$=-9.245844880001464E-05, $g_3(x)$=-0.024784596171230 and $g_4(x)$= -40.0035248-00000008. We can see that the solution *x* satisfy all constraints.

Table 6. Experimental results for the pressure vessel design (PVD) problem. The experimental results of the first nine algorithms are taken from [5]

Literature/ methods	x_1(best)	x_2(best)	x_3(best)	x_4(best)	f(best)	f(worst)	f(average)	std.dev.
Baykasoglu [5]	0.8125	0.4375	42.0975	176.648	6059.83	6823.602	6149.727	210.70
Coello [12]	0.8125	0.4375	40.3239	200.000	6288.74	-	-	-
Coello and Mezura-Montes [13]	0.8125	0.4375	40.0973	176.654	6059.94	-	-	-
Mezura-Montes et al. [39]	0.8125	0.4375	42.0983	176.637	6059.71	6846.628	6355.343	256.00
Mezura and Coello [38]	0.8125	0.4375	42.0984	176.636	6059.71	-	6379.938	210.00
Akhtar et al. [3]	0.8125	0.4375	41.9768	182.284	6171.00	-	-	-
He et al. [28]	0.8125	0.4375	42.0984	176.636	6059.71	-	-	-
Parsopoulos and Vrahatis [40]	-	-	-	-	6154.71	9387.770	8016.370	745.80
Aguirre et al. [1]	0.8125	0.4375	42.0984	176.636	6059.71	-	6071.013	15.10
He and Wang [26]	0.8125	0.4375	42.0912	176.746	6061.07	6363.804	6147.133	86.45
He and Wang [27]	0.8125	0.4375	42.0984	176.636	6059.71	6288.677	6099.932	86.20
Cagnina et al. [8]	0.8125	0.4375	42.0984	176.636	6059.71	-	6092.049	12.17
Maruta et al. [37]	0.8125	0.4375	42.0984	176.636	6059.71	7332.841	6358.156	372.7
Kim et al. [33]	0.8125	0.4375	42.0984	176.636	6059.71	6060.074	6059.727	0.065
Chun et al. [11]	0.8125	0.4375	42.0984	176.636	6059.71	6090.526	6060.330	4.357
Akay and Karaboga [2]	-	-	-	-	6059.71	-	6245.308	205.00
Brajevic and Tuba [7]	0.8125	0.4375	42.0984	176.636	6059.71	-	6192.116	204.00
Gandomi et al. [22]	0.8125	0.4375	42.0984	176.636	6059.71	6318.950	6179.130	137.20
Baykasoglu and Ozsoydan [6]	0.8125	0.4375	42.0984	176.636	6059.71	6090.526	6064.336	11.28
WSA [5]	0.7865	0.3934	40.7526	194.78	5929.62	5984.756	5958.410	15.07
SLPSO [10]	0.7884	0.3897	40.8542	192.688	5903.20	6425.556	6139.762	215.4
SRPSO [45]	0.9638	0.4764	49.9397	98.8235	6284.45	6290.601	6287.521	2.326
sinDE [19]	0.8713	0.4327	44.7686	147.183	6144.05	6309.273	6277.016	68.69
MGABC [15]	0.7788	0.3849	40.3404	199.741	5888.75	5941.013	5893.319	21.25
BCABC [23]	0.7924	0.3917	41.0587	189.959	5910.16	6051.407	5975.132	37.45
ABC-BB [48]	0.7828	0.3881	40.5345	197.198	5903.80	5972.571	5941.527	37.83
GBABC [48]	0.7834	0.3869	40.5319	197.07	5901.55	5975.849	5941.258	38.43
ABC-elite [14]	0.7796	0.3851	40.3643	199.385	5891.19	5933.783	5900.800	16.72
DGABC[24]	0.779	0.3851	40.3613	199.421	5887.12	5939.237	5919.170	24.78
EABC-BB	0.7781	0.3847	40.3198	199.996	**5885.34**	5914.163	**5888.892**	17.34

The Frequency Modulated (FM) Sound Wave Synthesis Problem The FM sound wave synthesis plays an important role in modern music systems. It provides a simple and efficient method creating complex sound timbres. This subsection applies the proposed EABC-BB method to optimize the parameters of an FM synthesizer. The details of the problems are described as follows.

The FM sound synthesis aims to optimize the parameter of an FM synthesizer with a D-dimensional vector $\boldsymbol{x}$. In this paper, we only consider the case of D=6 by the suggestions of the literature[48]. The objective of this problem is to optimize a six-dimensional vector x=$\{a_1,w_1,a_2,w_2,a_3,w_3\}$ of the sound wave given in Eq.(19). The problem is to generate a sound similar to the object sound. The formulas for the estimated sound wave and the target sound are given as follows:

$$\mathrm{y}(t) = a_1 \times \sin(w_1 t \cdot \theta + a_2 \times \sin(w_2 t \cdot \theta + a_3 \times \sin(w_3 t \cdot \theta))) \tag{19}$$

$$\mathrm{y}_0(t) = 1.0 \times \sin(5.0t \cdot \theta - 1.5 \times \sin(4.8t \cdot \theta + 2.0 \times \sin(4.9t \cdot \theta))) \tag{20}$$

where $\theta = 2\pi/100, x_i \in [-6.4, 6.35]$.

The goal of this problem is to minimize the sum of squared errors between the estimated sound and the target sound, as given by Eq.(20).This problem is a highly complex multimodal one having strong epistasis, with minimum value $f(x)$=0.

Table 7. Comparison of EABC-BB with different methods for the FM sound wave synthesis problem

Algorithms	Mean	Std dev	Best	Worst
ABC	6.78	6.52	1.76	11.83
SLPSO	6.91	4.42	4.12	12.78
SRPSO	6.74	5.96	**0**	13.53
sinDE	6.49	**0.58**	5.92	7.89
GABC	4.88	5.23	0.036	11.82
MGABC	5.58	5.92	7.72E-05	14.36
BCABC	4.54	4.49	3.12E-16	12.23
ABC-BB	2.85	2.68	0.27	11.83
GBABC	2.23	3.52	0.22	10.28
ABC-elite	0.77	2.14	1.62E-12	8.94
DGABC	0.26	3.12	**0**	9.72
EABC-BB	**0.072**	1.51	**0**	5.82

$$f = \sum_{t=0}^{100} \left(y(t) - y_0(t)\right)^2 \tag{21}$$

In the experiment, EABC-BB as well as the other 11 EAs are applied to solve this problem. For all algorithms, the parameters are set according to suggestions of the literature [48], i.e., *SN*=30, *max_FEs*=200,000. Each algorithm is run 30 times, and the mean and standard deviation values are reported in Table 7. From the results, it is clear that EABC-BB performs better than other 11 EAs in terms of the quality of the final solutions. Similar to the PVD problem, in order to make a fair comparison, we use DGABC without chaotic opposition strategy. DGABC also shows relatively good performance in FM problem, which shows again that the DE and ABC hybrid algorithm is an effective method to improve the performance of ABC and DE.

5. Conclusions

This paper presents an enhanced ABC-BB (EABCBB for short), which aims to overcome the premature convergence problem of ABC-BB. In EABC-BB, the solution search equation of onlooker bees is replaced with trigonometry based search equation. In this equation, the candidate is generated around the current solution, the global best solution and a randomly selected elite solution. By doing so, more function peaks represented by different elite solutions can be explored and the premature convergence problem of ABC-BB thereby can be effectively decreased. The reason is that the trigonometry method can keep up the population diversity and improve the exploration capability without lowering the exploitation ability of ABC-BB. Meanwhile, in order to overcome the crossover (*CR*) parameter sensitive problem in ABC-BB, a simple self-adaptive strategy is proposed to dynamically update *CR* according to the search experiences of successful *CR* probabilities. A comprehensive set of experiments is conducted on 23 benchmark functions and two well-known real-world engineering optimization problems to verify the performance

of our proposed approach EABC-BB. Some other well-known ABCs and state-of-the-art EAs, such as newly developed PSO variants and DE variants, are used for comparison. The experimental results demonstrate that our approach achieve better performance in term of solution accuracy and convergence speed, thus we may reasonably conclude that the proposed EABC-BB is a new competitive algorithm. It is worthy note that another DE and ABC hybrid algorithm DGABC also show excellent performance in aforementioned two engineering optimization problems, which demonstrates again that the DE and ABC hybrid algorithm is a promising research direction if the algorithm is well designed.

Acknowledgements This research is funded by the Science and Technology Plan Project of Chaozhou under Grant No. 2018GY45, National Natural Science Foundation of China (No. 61672338) and National Key Research and Development Program of China (No. 2016YFD0702100), National Natural Science Area Foundation Project of China (No. 61863011) and Guangxi Natural Science Youth Foundation Project of China (No. 2015-GXNSFBA139264).

References

1. Aguirre, A.H., Zavala, A.M., Diharce, E.V., Rionda, S.B.: COPSO: constrained optimization via PSO algorithm. Center for Research in Mathematics (2007)
2. Akay, B., Karaboga, D.: Artificial bee colony algorithm for large-scale problems and engineering design optimization. Journal of Intelligent Manufacturing 23(4), 1001–1014 (2012)
3. Akhtar, S.and Tai, K., Ray, T.: A socio-behavioural simulation model of engineering design optimization. Engineering Optimization 34(3), 341–354 (2002)
4. Aydogdu, I., Akin, A., Saka, M.P.: Design optimization of real world steel space frames using artificial bee colony algorithm with levy flight distribution. Advances in Engineering Software 92(1), 1–14 (2016)
5. Baykasoğlu, A., Akpinar, S.: Weighted superposition attraction (wsa): a swarm intelligence algorithm for optimization problems–part 2: constrained optimization. Applied Soft Computing 37(4), 396–415 (2015)
6. Baykasoğlu, A., Ozsoydan, F.B.: Adaptive firefly algorithm with chaos for mechanical design optimization problems. Applied soft computing 36(11), 152–164 (2015)
7. Brajevic, I., Tuba, M.: An upgraded artificial bee colony (abc) algorithm for constrained optimization problems. Journal of Intelligent Manufacturing 24(4), 729–740 (2013)
8. Cagnina, L.C., Esquivel, S.C., Coello, C.A.C.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. Informatica 32(3), 319–326 (2008)
9. Chen, J., Yu, W., Tian, J.: Image contrast enhancement using an artificial bee colony algorithm. Swarm and Evolutionary Computation 38(2), 287–294 (2017)
10. Cheng, R., Jin, Y.: A social learning particle swarm optimization algorithm for scalable optimization. Information Sciences 291(3), 43–60 (2015)
11. Chun, S., Kim, Y.T., Kim, T.H.: A diversity-enhanced constrained particle swarm optimizer for mixed integer-discrete-continuous engineering design problems. Advances in Mechanical Engineering 5(1), 1–11 (2015)
12. Coello, C.A.C.: Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry 41(2), 113–127 (2000)
13. Coello, C.A.C., Mezura-Montes, E.: Use of dominance-based tournament selection to handle constraints in genetic algorithms. Intelligent Engineering Systems through Artificial Neural Networks 11(1), 177–182 (2001)

14. Cui, L., Li, G., Lin, Q.: A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation. information sciences. Information Sciences 367(3), 1012–1044 (2016)
15. Cui, L., Zhang, K., Li, G.: Modified gbest-guided artificial bee colony algorithm with new probability mode. Soft Computing 22(1), 1–27 (2017)
16. Deng, Y.H.: What is the future of disk drives, death or rebirth? ACM Computing Surveys 43(3), 1–27 (2011)
17. Derrac, J., S, G., Molina, D.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 11(1), 3–18 (2011)
18. Dorigo, M., Maniezzo, V., Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics 26(1), 29–41 (1996)
19. Draa, A., Bouzoubia, S., Boukhalfa, I.: A sinusoidal differential evolution algorithm for numerical optimisation. Applied Soft Computing 27(2), 99–126 (2015)
20. Du, Z., Han, D., Li, K.C.: Improving the performance of feature selection and data clustering with novel global search and elite-guided artificial bee colony algorithm. The Journal of Supercomputing (2019), https://doi.org/10.1007/s11227-019-02786-w
21. Du, Z., Han, D., Liu, G.: An improved artificial bee colony algorithm with elite-guided search equations. Computer Science and Information Systems 14(3), 751–767 (2017)
22. Gandomi, A.H., Yang, X.S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems,. Engineering with Computers 29(1), 17–35 (2013)
23. Gao, W., Chan, F., L., H.: Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood. Information Sciences 316(3), 180–200 (2015)
24. Gao, W., L., H., J., W.: Enhanced artificial bee colony algorithm through differential evolution. Applied Soft Computing 48(11), 137–150 (2016)
25. Gao, W., Liu, S., Huang, L.: A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. IEEE Transactions on Cybernetics 43(3), 1011–1024 (2013)
26. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence 20(1), 89–99 (2007)
27. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation 186(2), 1407–1422 (2007)
28. He, S., Prempain, E., Wu, Q.H.: An improved particle swarm optimizer for mechanical design optimization problem. Engineering Optimization 36(5), 585–605 (2004)
29. Jafrasteh, B., Fathianpour, N.: A hybrid simultaneous perturbation artificial bee colony and back-propagation algorithm for training a local linear radial basis neural network on ore grade estimation. Neurocomputing 38(2), 287–294 (2017)
30. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. APPLIED MATHEMATICS AND COMPUTATION 214(1), 108–132 (2009)
31. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. Advances in Engineering Software 39(3), 459–471 (2007)
32. Karaboga, D., Ozturk, C.: A novel clustering approach: artificial bee colony (abc) algorithm. Applied Soft Computing 11(2), 652–57 (2011)
33. Kim, T.H., Maruta, I., Sugie, T.: A simple and efficient constrained particle swarmoptimization and its application to engineering design problems. Journal of Mechanical Engineering Science 224(2), 389–400 (2010)
34. Kumar, Y., Sahoo, G.: A two-step artificial bee colony algorithm for clustering. Neural Computing and Applications 28(3), 537–551 (2017)

35. Li, Y., Z., Z., Lin, S.: Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems. Information Sciences 293(3), 370–382 (2015)
36. Lozano, M., García-Martínez, C.: Optimizing network attacks by artificial bee colony. Information Sciences 377(1), 30–50 (2017)
37. Maruta, I., Kim, T.H., Sugie, T.: Fixed-structure h∞ controller synthesis: a metaheuristic approach using simple constrained particle swarm optimization. Automatica 45(3), 553–559 (2009)
38. Mezura-Montes, E., Coello, C.A.C.: Useful infeasible solutions in engineering optimization with evolutionary algorithms. In: Mexican International Conference on Artificial Intelligence. pp. 652–662. Springer, Verlag Berlin Heidelberg (2005)
39. Mezura-Montes, E., Coello, C.C., Landa-Becerra, R.: Engineering optimization using a simple evolutionary algorithm. In: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence. pp. 149–156. IEEE, Sacramento, CA, USA (2003)
40. Parsopoulos, K.E., Vrahatis, M.N.: Unified particle swarm optimization for solving constrained engineering optimization problems. In: International Conference on Natural Computation. pp. 582–591. Springer, Berlin, Heidelberg (2005)
41. Qin, A.K., Huang, V.L.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation 13(2), 398–417 (2009)
42. R., E., J., K.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science. pp. 39–43. Ieee, Nagoya, Japan (1995)
43. Storm, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
44. Suganthan, P.N., Hansen, N., J., L.J.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, Singapore. Nanyang Technological University, Nanyang Technological University (2005)
45. Tanweer, M.R., Suresh, S., Sundararajan, N.: Self regulating particle swarm optimization algorithm. Information Sciences 294(3), 182–202 (2015)
46. Wei, Y.H., Xu, C., Hu, Q.Y.: Transformation of optimization problems in revenue management, queuing system, and supply chain management. International Journal of Production Economics 146(2), 588–597 (2013)
47. Zhang, J., Sanderson, A.C.: Jade: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation 13(5), 945–958 (2009)
48. Zhou, X., Wu, Z., Wang, H.: Gaussian bare-bones artificial bee colony algorithm. Soft Computing 20(3), 907–924 (2016)
49. Zhu, G., S., K.: Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and Computation 217(7), 3166–3173 (2010)

Zhenxin Du received the B.S. degree in computer science from Zhejiang Sci-tech University in 2011. He is is a lecturer of computer science and engineering at Hanshan Normal University. His main research interests include computation intelligence, data mining and cloud computing.

Keyin Chen (corresponding author) received the Ph.D degree from South China Agricultural University. He is currently a post doctor of agricultural electrification and automation engineering at Nanjing Institute of Agricultural Mechanization and also at the same time is lecturer of electrical engineering engineering at Hezhou University. His research interests include agricultural robot, machine vision and bionic intelligence technology.

Received: September 23, 2018; Accepted: August 27, 2019.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information Systems : the International journal /
Editor-in-Chief Mirjana Ivanović. – Vol. 16, No 3 (2019) - . – Novi Sad (Trg D. Obradovića 3): ComSIS Consortium, 2019 - (Belgrade : Sigra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 (Print) 2406-1018 (Online) = Computer Science and Information Systems
COBISS.SR-ID 112261644

Cover design: V. Štavljanin
Printed by: Sigra star, Belgrade