Computer Science and Information Systems 18(2):597–618 https://doi.org/10.2298/CSIS200330012S

Machine Learning Based Distributed Big Data Analysis Framework for Next Generation Web in IoT

Sushil Kumar Singh[1], Jeonghun Cha[1], Tae Woo Kim[1], and Jong Hyuk Park[1, *]

[1]Department of Computer Science and Engineering,
Seoul National University of Science and Technology,
(SeoulTech) Seoul 01811, Korea
{sushil.sngh001007, ckwjdgns, tang_kim, jhpark1}@seoultech.ac.kr

Abstract. For the advancement of the Internet of Things (IoT) and Next Generation Web, various applications have emerged to process structured or unstructured data. Latency, accuracy, load balancing, centralization, and others are issues on the cloud layer of transferring the IoT data. Machine learning is an emerging technology for big data analytics in IoT applications. Traditional data analyzing and processing techniques have several limitations, such as centralization and load managing in a massive amount of data. This paper introduces a Machine Learning Based Distributed Big Data Analysis Framework for Next Generation Web in IoT. We are utilizing feature extraction and data scaling at the edge layer paradigm for processing the data. Extreme Learning Machine (ELM) is adopting in the cloud layer for classification and big data analysis in IoT. The experimental evaluation demonstrates that the proposed distributed framework has a more reliable performance than the traditional framework.

Keywords: machine learning, big data analysis, extreme learning machine, IoT, security, and privacy.

1. Introduction

With the fast-growing development of the digital world as next-generation web, IoT (Internet of Things) is adopted in several applications such as smart services, smart communication, smart community, public safety, and many more. Every aspect of our lives combines many things and next-generation web in IoT communication mediums, such as sensor devices, Bluetooth, Wi-Fi, GPRS, etc. [1]. Internet is a fascinating medium for communication, and it is offered in IoT to connect all objects with their automatic features. IoT has the most significant role with advancement applications in future revolutions with next-generation web, and utilization is continuously increased over the coming times. Next-generation web in IoT means to provide distinct requirements, such as smart devices, accuracy, efficient analysis, low energy, and others. According to Keenan et al.'s report [2], the global IoT data management market value at around $27.13 billion in 2017. It would reach approximately $94.47 billion in 2024 and above 19.51 percent between 2018 to 2024 at a CAGR. Nowadays, both Big

* Corresponding author

data and IoT technology are growing for the data science field with great attention. Big data is collecting the enormous amount of structured, unstructured, or semi-structured raw data that is more complex to managing and analyzing with many traditional tools. As per the study of Verma et al. [3], around 4.4 trillion GB data will be produced by 2020 using IoT devices in smart applications. IoT is serving as the primary role when any enterprises have a vast number of data for analysis purpose in reinforcements. Big data analytics is a rapidly advancing field for managing and analyzing IoT data. [4, 5]. It is connected to smart devices, which helps to take the initiative to improve decision making. Due to the popularity of online media, including WhatsApp, Instagram, Snapchat, Linkedin, and an expanding number of IoT devices, significant data analysis issues in IoT have been raised in smart technological fields. Big data analytics's main task is to extract useful patterns from the massive amount of IoT data that can be used in decision and prediction making responsibilities. However, many researchers are used various technologies such as edge computing, predictive analytics, Apache Spark, Apache Flink, and so on, which have some challenges according to the next generation web in IoT [6]. The machine learning approach analyzes and extracts accurate data from raw structured, unstructured, and semi-structured data to mitigate these challenges by IoT devices [7].

The existing mechanism to big data analysis in IoT applications for next-generation web on a centralized cloud is not adequately satisfied for specific requirements such as resource management, latency, scalability, accuracy, communication bandwidth. However, with the consecutive growth in data-driven applications in IoT and generated a massive information. In recent years, various machine learning paradigms have been discussed to promote valuable data interpretation for IoT applications [8, 9]. Many operations adapt to data control among various communication devices, and it provides intelligence processing, analysis in IoT. Cloud computing is employed for delivering high performance to the IoT server [10]. Traditional machine learning also has significant issues such as low precision, low rate, low latency, and less computational in the cloud layer. An extreme learning machine (ELM) is utilized, which provides excellent performance to address these issues. It increased the learning speed of feedforward neural networks with a hidden layer of transferring data in the cloud network layer to analyze the IoT data.

Feature extraction and scaling are utilizing for processing the IoT data at the edge layer with address the load balancing, data computation issues on the cloud layer. However, the parts of the device layers with data are moving at the cloud, which reduces the intermediate data computation and processing at the edge layer. [11]. On the different side, edge computing serves as a backbone in the IoT and provides computation power and desired latency to smart applications' IoT devices. To mitigate standard limitations in IoT applications such as high computation, load balancing, network traffic and storage by feature extraction, and scaling at the edge layer to data processing. The device layer delivered massive IoT data, and it is collected from various complex and noisy environments [12]. The edge layer is investigating as a type of feature extraction and scaling-based intelligent computing that could overcome the cloud layer's limitation. Due to data transfer with low network enforcement, the centralized cloud computing layer is becoming inefficient for analyzing and processing a massive amount of data collected from IoT devices. Edge nodes provide efficient storage, computing, and networking services with essential data in IoT applications at the edge layer [13]. The cloud layer has a centralized database with an advanced analysis of IoT data using extreme learning techniques and transferring these data to other IoT devices. It provides the leading the creation of current data related smart applications. For identifying the object in the collected video data, AlexNet (Convolutional Neural Network) tools are deployed, where we train the machine intelligence network with the help of the Kaggle open dataset. Then we detect the correct image [14].

The existing research studies to big data analysis in IoT applications on a centralized cloud is not adequately satisfied and have some challenges or requirements such as resource management, computational cost, scalability, accuracy, and latency. This paper addresses and discusses the challenges of accuracy, privacy, load balancing, resource limitation, and centralization using the proposed machine learning-based distributed big data analysis framework for the next-generation web in IoT. In this framework, data storing is utilized in the device layer, the edge layer has a data processing part, and big data analysis is completed in the cloud layer. The primary goal of our study is to provide decentralized and secure big data analysis for the next-generation web in IoT by the proposed framework.

To summarize, the main contributions of this paper are as follows:

- Propose Machine Learning-Based Distributed Big Data Analysis Framework for Next Generation Web in IoT, which provides the precise necessities of advanced applications, including accuracy, performance analysis, load balancing, resource limitation, energy consumption, and schedule.
- We deploy the PCA algorithm for feature extraction, K-means algorithm for scaling for processing data in the edge layer, and ELM algorithm for classification and analysis of big data at the cloud layer.
- Evaluate the proposed framework's performance by comparing it with different traditional machine learning classifiers with the NSL-KDD dataset and evaluating performance for various IoT applications such as attack detection and object detection.
- Finally, we graphically analyze the big data with accuracy, testing time, training time, and precision using the ELM algorithm for IoT applications and compare our proposed work with the existing research.

The remains of the paper, we present several existing methods or techniques to big data analysis for the next-generation web in Section 2; Section 3 proposes a Machine learning-based distributed big data analysis framework that introduces an edge computing paradigm for processing the data and ELM based big data analysis in cloud layer for analysis and classification of IoT data and provides principal component analysis algorithm for feature extraction, K-means algorithm for scaling on edge layer and ELM algorithm for classification and analysis of data in cloud layer; in Section 4, we graphical analyze of the proposed framework on the KDDTest+, KDDTest-21 dataset. Finally, we conclude in Section 5.

2. Seminal Contribution

This section shows the existing research for next-generation IoT with addressing issues such as accuracy, security, latency, energy consumption, centralization, and uses of specific own proposed framework. Li et al. [14] discussed a novel moving strategy to optimize IoT applications' production with deep learning. It tested the fulfillment of executing various deep learning tasks in the edge computing paradigm for IoT. For identifying objects in the collected data, the AlexNet convolutional neural network is used with six layers. The convolution layer is the first five layers, and the other three are a fully connected layer and completed the feature extraction task. However, it is used only for video data and did not provide a real-world edge computing paradigm. Peng Li et al. [15] proposed a deep convolution computational model (DCCM) and learned hierarchical features of big data in IoT. This model is utilized to extend CNN and improve efficiency by using vector space to tensor space. Chhowa et al. [16] provided a smart proposal for health monitoring in IoT to big data analysis and gave accurate healthcare data on IoT-based system. They focused on a deep learning based IoT system for health monitoring devices and provided efficient results to the various doctors in the IoT environment. It ensures the proper knowledge about the critical patient. However, the number of mammogram devices increases, then the delay is also raised to diagnose disease. Mishra et al. [17] provided a framework related to big data analysis in IoT applications with cognitively oriented infrastructure. It provided implementation architecture for adequate data supervision and information search in manufacturing IoT applications. Zhang et al. [18] described a double projection model with deep computation (DPDCM) to feature big data learning in IoT. Raw input data is separated into two subspaces in the hidden layers to understand the interacted big data feature in IoT applications.

Hosseini et al. [19] utilized a dimensionality reduction technique to improved classification accuracy, reduced communication bandwidth, and computation time of data. Also, they proposed a cloud computing solution for interpretation of big EEG data. However, a large number of training samples problem and heterogeneous data of multidomain propagation are not entirely resolved. Vinay et al. [20] discussed the novel FR approach-based framework on ELM to perform appearance tagging for friendly networks operation on extensive data using machine learning. It is only used for face recognition and has more centralized data, communication bandwidth on cloud problems.Ying et al. [21] proposed an integrated framework to enhance smart city applications' performance by enabling effective orchestration of networking and computing supports. Liu et al. [22] developed a deep learning-based visual food recognition algorithm to achieve the best accuracy of massive data analysis in IoT. It designed edge computing-based paradigms to overcome traditional mobile cloud computing's inherent problems in the food recognition system for IoT. Liangzhi Li et al. [11] adopted state-of-the-art edge computing arrangement to address the crowdsensing problem. It provided distribute deep learning principles to extract characteristics from taken IoT data. It reduced communication costs and increased safety protection of data in IoT. However, it has a compatibility problem for all cloud environments. Jeong et al. [23] present a paradigm to address intrusion detection for various research areas such as image segmentation, security distribution networks, fingerprint matching, human tracking, image watermarking, and big data analysis in IoT.

We are categorized related work in some subsections such as security architecture, technological aspects, methodology, dataset labeling requirement, and conventional machine learning classifier. We are providing Table 6 in Section 4 and compare the proposed framework with existing research studies. Some research used centralized security architecture and smart city technological aspects, but we use distributed security architecture and next-generation web in IoT technical aspects. In summary, existing studies have used cloud and edge layer frameworks to big data interpretation in advanced applications. However, such a framework and architecture have some limitations on the centralized cloud, such as a massive amount of data, resource utilization, low accuracy and latency, security, and privacy, etc. Moreover, existing strategies use conventional methods and architecture, which requires more computational power and efficiency. It is essential to compose and develop a new framework for big data interpretation in IoT that considers all instant and future difficulties. Thus, we provide a machine learning-based edge computing framework where feature extraction and scaling concepts are used to process IoT data on the edge layer. Also, we are giving ELM based cloud structure for analyzing structured and unstructured big data in IoT efficiently and quickly on the cloud layer.

3. Proposed Framework for Big Data Analysis in IoT

Based on the limitations such as low efficiency, low latency, centralization, computational cost, resource management of existing studies, we propose a machine learning-based distributed big data analysis framework for the next-generation web in IoT. We focused on processing and analyzing a lot of data on the cloud to develop our framework with ELM based big data analysis in the cloud layer for IoT. We discussed feature extraction and scaling in the edge layer to find specific data with the clustering concept on training data and testing data. We introduce the PCA algorithm for feature extraction, K-means algorithm for scaling, and Naive Bayes algorithm for classification in the edge layer. The proposed framework has various advantages for the next-generation web in IoT, which are the following:

- Provides the distinct requirements of IoT applications such as accurate data, better performance analysis, load balancing, maximum resources.
- With the next-generation web, improves the transparency and connectivity in IoT applications.
- Provide comfortable environments and appropriate high compatible or accurate big data analysis for IoT infrastructure.
- With the proposed framework, improve reliability and efficient operations in IoT applications.

3.1. Proposed Framework Design

IoT structure is a core technology for connecting smart devices and humans to the internet with a wired or wireless medium. It is known as the Internet of Everything (IoE). It deals with a massive amount of IoT data at the device layer. It produced big data from sensing devices and various IoT applications such as smart services, smart communication, smart community, and many more. The device layer is mainly used for data acquisition and recorded from sensors and IoT applications. Due to increasingly continuous various data sources, accuracy, latency, and trust become a challenge in big data analytics. In this situation, big data analysis in IoT is a very critical issue in the cloud layer. Therefore, this data representation generated different types of big data challenges to extract useful data from unstructured and semi-structured data on the device layer. To mitigate these problems, we utilize state-of-the-art machine intelligence-based edge computing layer. High connectivity, processing, scaling, and feature extraction capabilities for edge device nodes at the edge layer with machine-intelligence are the primary role of next-generation web in IoT applications and provide efficient operations with accurate big data analysis. Next-generation web connectivity is provided by advanced technologies between the device layer and edge layer.

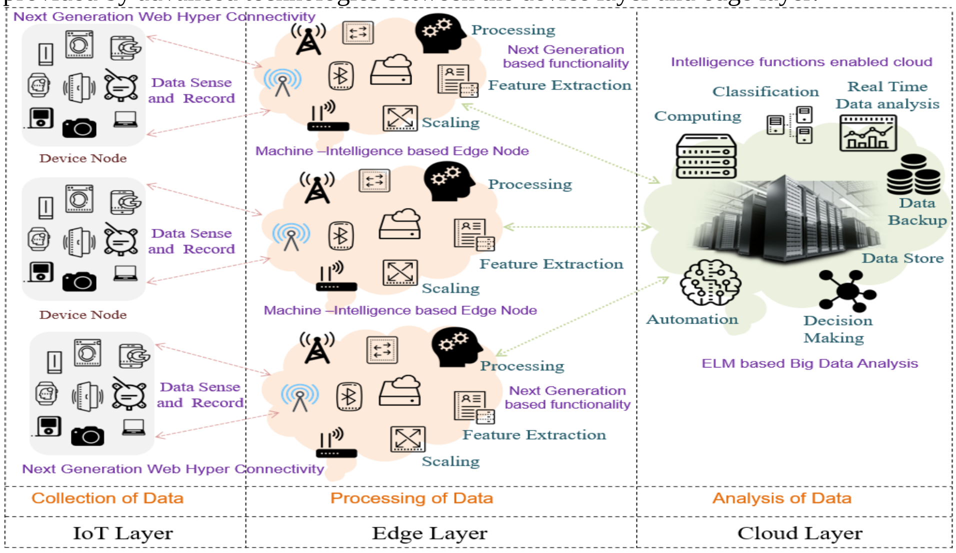

Fig. 1. Architectural overview of the proposed framework.

The edge layer has two next generation-based functionalities, including Feature extraction and scaling for processing of IoT structured and unstructured data. Used data and unused data are categorized to provide data. Feature subset and hidden layer applied to it and found training, testing data set. It used the PCA algorithm for feature extraction, the K-means algorithm for scaling. Every machine intelligence-based edge node has base stations, networking devices, and machine learning, which provide computation power to the physical layer's IoT devices. All edge nodes transfer the extracted data to the cloud layer with a base station and networking device such as a router. Thus, load balancing and energy efficiency issues resolve in the edge and cloud layer using machine-intelligence-based feature extraction and scaling. However, the cloud layer has one data center, so accuracy, speed, computational storage is very low. Therefore, the ELM algorithm is used to classify and analyze data in the cloud layer's proposed framework. It examines the data and improved performance, accuracy, latency, and efficiency of IoT data. The overview architecture for the proposed framework is as shown in Fig. 1.

Table 1. Abbreviation table

Abbreviation	Description	Abbreviation	Description
Z	Dimensional linear Subspace	$\{S_1, S_2, \dots . S_k\}$	Cluster data center sets.
$\{a_1, a_2 \dots, a_M\}$	Labeled or Unlabelled data sets	K	Number of clusters or groups
$\{b_1, b_2 \dots, b_M\}$	Projected data sets	α_i	Output weight of a hidden node
w_i	Weight vector or eigenvector value	H(p)	ELM hidden layer output
X	Input vector Matrix	M	Training sample
Y	Output vector Matrix	H	Matrix hidden layer output
β_{mk}	Binary indicator variable sets.	T	Training data set a target matrix

3.2. Functional Components of the Proposed Framework

This subsection presents the main functional component of the proposed framework. It is divided into three parts, including data acquisition or collection, data processing, and data analysis. Data acquisition is used in the IoT layer, data processing is completed at the edge layer, and data analysis is utilized in the cloud layer. PCA algorithm used for feature extraction, K-means algorithm utilized for scaling data processing at the edge layer, and the cloud layer have an ELM algorithm for analyzing massive data in IoT applications. Machine-Intelligence based distributed big data analysis flow is shown in Fig. 2. (a). The principal component analysis (PCA) algorithm is utilized for feature extraction, the K-means algorithm is used for scaling, and the ELM algorithm for classification and analysis of data. ELM-based classification and data analysis on the cloud layer are shown in Fig. 2. (b). The abbreviation table is shown in Table 1.

Data Collection*:* Data collection is an essential function for the proposed framework, and it is used in the device layer. Various IoT applications such as smart services, smart communication, smart community, and others generated a massive amount of IoT devices. IoT devices are connected to various sensors and detect the data in several forms, including video, audio, multimedia, etc. It can measure and monitor the data in real-time. Various data types are stored in a device layer such as automation, location, streaming, status data, and others. Internal sensors collect IoT data from consumer devices such as smart appliances, smart televisions, wearable health meters, and commercial devices such as traffic monitoring systems, weather forecasting, and commercial security system.

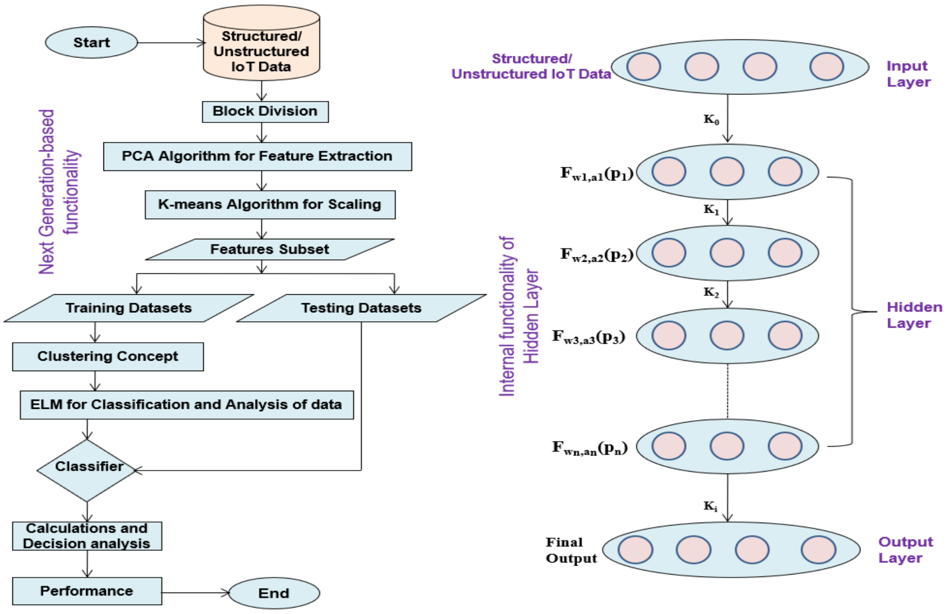

Fig. 2. (a) Flow of Machine Learning-based Distributed Big Data Analysis Framework **(b)** ELM-based Classification and Analysis of Data on Cloud Layer

Data Processing: Data processing is the second functional component of the proposed architecture used in the edge layer. The IoT sensor devices must generate an extensive quantity of data that must be processed before the IoT information can be utilized. However, these data come from IoT devices in various types of formats. To mitigate this problem, feature extraction and scaling concepts are used to process IoT data at the edge layer. For data processing, PCA and K-means algorithms are utilized. a' is average value.

Algorithm 1. Principal Component Analysis

Input Dimensional linear subspace is Z and the input vector of a labeled or unlabelled data set is $\{a_1, a_2, \ldots\ldots a_M\}$.

Output: Projected data set $\{b_1, b_2, \ldots\ldots b_M\}$ and weight vectors $\{w_j\}$ which have essential subspace form.

Process:

$a' = \frac{1}{M}\Sigma_m a_m$ */* M training samples and a_m input vector*

$S = \frac{1}{M}\Sigma_m (a_m - a'_m)(a_m - a'_m)^T$ /*$a'_m = \Sigma_{j=1 \text{ to } R} b_{mj} w_j + a'$

Eigenvector value is $\{w_j\}$ of S.

for all m = 1 to M
 do
  for all j= 1 to Z
   do
    $b_{mj} = (a_m - a'_m)^T w_j$
   end
 end

Feature Extraction

It is one of the primary next-generation web function for machine learning to identify strong and weak relevant features and labels for IoT applications. It is used mainly for load balancing, and improving the communication bandwidth, time delay in IoT applications. Due to the continuously increasing massive amount of data, it is characterized into three groups: big data, data usage, data quality. Big data is also characterized as volume, variety, velocity, and veracity of IoT data. Completeness, noiseless, semantics are the part of data usage, and accuracy, redundancy, efficiency, loading balancing are the part of data quality. Therefore, feature extraction methods are using on the edge layer. IoT devices have many types of sensors, analyzed, monitoring data. Processing system components and communication protocols in IoT is divided into three parts: device to device, the device to the server, and server to server. While many feature extraction algorithms are available, but we are using the PCA algorithm. This algorithm's main objective is to extract the relevant features from various IoT structured and unstructured data sets, which is stored in the device layer of the proposed model. Feature extraction function discriminates the essential and useful features by eliminating redundant features and noise from IoT applications and provide the best-predicted output features of IoT data. The extracted relevant and non-relevant features can help us to identify new useful information that is used in machine learning.

From a machine learning perspective to IoT data, Feature extraction is an essential concept for processing the data on the edge layer. Firstly, IoT data is divided into variable or fixed blocks, then feature extraction and scaling concepts are used. Training sets have various inputs as samples, which are used by machine learning. Supervised learning utilized input vector and related output vector (labels); these all are samples. Vectors (labels) are not required for unsupervised learning. Reinforcement learning is used to understand to expropriate steps to be taken for a specific condition. Determine the main groups (cluster) between comparable sample clusters that comprehend as clustering. Original input samples transferred into a new variable sample or space are called feature extraction and improves the result.

PCA is the most straightforward algorithm for feature extraction of data set in IoT applications, and it is based on actual eigenvalues of IoT data. The PCA algorithm's main objective is to overcome the overfitting problem and decrease the dimensionality of IoT data sets. It may be large or less, while retaining the variation present in the dataset, up to the maximum extent [24]. According to principal component analysis (PCA), the principal subspace has the orthogonal data points. It is the property of PCA and has the maximum projected variance of data. The main task defined as the finding the complete Z orthogonal data set, it is based on M-dimensional vectors $\{w_j\}$. Parallelly find the corresponding linear projections data points $\{Y_{n_j}\}$and minimized the reconstruction error. Average of all data points is a'.

Assume that the given training set have M training samples dented as (a_i, b_i) where i = 1,2,3……..M,$a_i$is training N-dimensional input vector, b_i is corresponding desired P-dimensional output vector. $X = (a_1, a_2, \dots \dots a_M)^T$is input vector matrix and $Y = (b_1, b_2, \dots \dots b_M)^T$is corresponding output vector matrix.

$$B = \frac{1}{\mathrm{M}} \Sigma_{\mathrm{m}} (a_m' - a_m)^2 \quad \textbf{(1)}$$

$$a_m' = \Sigma_{\mathrm{j=1\ to\ R}} b_{mj}\, w_j + a' \quad \textbf{(2)}$$

According to algorithm 1, provide the concept of principal component analysis, and overcome the overfitting problem, extract the features of IoT data in processing. PCA has various versions; it is based on data size. It may be a structured dataset or unstructured dataset. Due to calculation of $\{w_1, w_2, \ldots \ldots w_z\}$, algorithm have different run times values.

For preprocessing, the principal component analysis algorithm is an essential method in machine learning. In preprocessing for PCA, find the features and labels in the processing of IoT data. Various practical applications are involved for PCA such as data compression, data visualization, face recognition, image rendering, and so on.

Data Scaling

Data Scaling is another next-generation web task for processing the IoT data at the edge layer. Scaling is the task of dividing the data set value into several specific groups. The data sets a value in the same groups that are more similar to other data sets value in the same group than those in other groups. The main objective of scaling is to segregate data set groups with the same properties. K-means algorithm is used for scaling the IoT data where identifies a similar type of data set values in a group [25]. It gives the groups (K) which is related to each other.

For machine learning, the K-means algorithm is one of the simplest and popular for clustering. It does not have labels or results in data processing, so it is called unsupervised learning [10, 26]. The K-means algorithm's primary idea is to group (cluster) related data sets values and recognize underlying designs. Based on algorithm, find a set of K group clusters $\{S_1, S_2, \ldots . S_k\}$. It reduces the distance between data features values and the most adjacent data hub. To denote data points value to the group hubs, then apply binary pointer variables βmk ∈ {No, Yes} or {0, 1}. We formulate dilemma as regards with the equation:

$$Minimize \Sigma_{\mathrm{m=1\ to\ M}} \Sigma_{\mathrm{k=1\ to\ K}}\, \beta \mathrm{mk}\, (\mathrm{a_m} - \mathrm{s_k})^2 \qquad \textbf{(3)}$$

Where,

$$\beta \mathrm{mk}\, (\mathrm{a_m} - \mathrm{s_k})^2 = 1 \text{ for data point } a_m \text{ belong to cluster, otherwise } \beta\mathrm{mk} = 0 \text{ and } \mathrm{m} = 1, 2, 3, \ldots \ldots \ldots \ldots \ldots . \mathrm{M}$$

Minimization has two sections: 1) Distance minimize concerning β_{mk} and s_k stable; 2) Distance minimize concerning s_k and β_{mk} is stable [10].

Genetic clustering methods can predict the movement of points is known as k-means. It is used for various IoT applications, but this method has some considerations, such as low efficiency and communication bandwidth compared to Euclidian distance. For classifying the IoT data, it is a successful machine learning approach. It may lead to unsuitable clusters in some cases. With the help of a 1-nearest neighbor classifier, find new data value in the existing clusters. Algorithm 2 describes how to get the optimal group datacenters S_k and the authorization of the data points β_{mk} [10].

Algorithm 2. K-Means [10]

Input: The number of same data points groups is K and unlabelled data sets are $\{a_1, a_2, \ldots a_M\}$.
Output: $\{s_k\}$ is cluster data center and {βmk} is assign data points randomly initiate with $\{s_k\}$
Process:

Repeat
  for $\forall(\mathrm{m} = 1 \text{ to } \mathrm{M})$
    do /* *m=1,2,3…………………M*
    for $\forall$ ($\mathrm{k} = 1$ to K)
      do
      if $k = \arg\min_i (s_i - x_i)^2$
        then $\beta\mathrm{mk} = 1$ /* *βmk is set of binary assigned data points*
      else $\beta\mathrm{mk} = 0$
    end
  end
  for $\forall$ ($\mathrm{k} = 1$ to K)
    do
    $\mathrm{s_k} = \Sigma_{\mathrm{m=1\ to\ M}}\, \mathrm{a_m}\ \beta\mathrm{mk}/\Sigma_{\mathrm{m=1\ to\ M}}\, \beta\mathrm{mk}$ /* *s_k is set of K cluster center*
    end
      while confluence adjust {βmk} or $\{\mathrm{s_k}\}$
end procedure
Return $\{a_1, a_2, \ldots a_M\}$

Data Classification and Analysis

Data classification and analysis are the final next-generation web function of the proposed framework and used with the ELM algorithm's help in the cloud layer. The ELM algorithm is used to analyze and classify the IoT data to address centralization and data handling issues with feedforward neural networks [27]. The relationships between the input and the hidden layer are randomly allocated and continue uninterrupted during the ELM algorithm's training method [28]. It includes two steps in the learning phase: 1) creating the hidden layer output model and 2) find the output combinations. Then, the output combinations are tuned by reducing the cost function using a linear system. By this system, the computational weight of the ELM is continuously decreased in IoT applications. The low computational weight or complexity are used for evaluation result in machine learning and utilized in high dimensional and large-data applications. To mitigate energy consumption and the massive amount of data in IoT, ELM Algorithm is utilized in the proposed framework. ELM has resolved various classification problems because it gives more excellent efficiency for handling the massive amount of data. ELM classifier adopts the hidden connections for classification, and the hidden connection output used as a sigmoid activation formula $q(y) = 1/(1 + e^{-y})$ to evaluate the output value [29].

Suppose that the output formula of the i^{th} hidden node is $h_i(p) = J(a_i, b_i, p)$, where (a_i, b_i) is the hidden connection parameter in the given single hidden layer of ELM. It is the basic method of the ELM algorithm for hidden layer feedforward neural interfaces. The ELM is an algorithm, and it is used for addressed single hidden layer neural interface problems and provide several hidden layers. With the use of the ELM algorithm, we can easily analysis of big data in various IoT applications.

The output function of ELM with Z hidden nodes

$f_Z(p) = \Sigma_{\mathrm{i=1\ to\ Z}}\ \alpha_\mathrm{i} \mathrm{hi}(p)$, Where α_i is output weight of $\mathrm{i^{th}}$ hidden node

$h(p) = [J(h_i(p), \ldots\ldots\ldots.. h_Z(p))]$ is the ELM hidden layer output.

If M given the training samples, then the ELM hidden layer output formula is given as:

$$H = \begin{bmatrix} h(p_1) \\ h(p_2) \\ \vdots \\ \vdots \\ h(p_M) \end{bmatrix} = \begin{pmatrix} J(a_1,\ b_1, p_1) & \cdots & J(a_Z,\ b_Z, p_1) \\ \vdots & \ddots & \vdots \\ J(a_1,\ b_1, p_M) & \cdots & J(a_Z,\ b_Z, p_M) \end{pmatrix} \quad \textbf{(4)}$$

Training data set targetmatrix

$$T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ \vdots \\ t_M \end{bmatrix} \quad \textbf{(5)}$$

ELM is a regulation neural network. However, hidden layer mapping formed by both random hide nodes and its objective function is as follows:

$$Minimize: (\alpha)_{\mathrm{r}}^{\tau 1} \ + \ \mathrm{C}(\mathrm{H}\alpha - \mathrm{T})_{\mathrm{g}}^{\tau 2} \quad \textbf{(6)}$$

Where $\tau_1 > 0, \tau_2 > 0\ and\ r, g = 0, \frac{1}{2}, 1,2, \dots \dots \dots \dots \dots . \infty$

We can use a different combination of $\tau_1, \tau_2, r, g,$ and find different results in various learning algorithms for regression, classification, compression, clustering, and, others. With ELM, computes the hidden layer output formula using training and classification.

Input: Training set F= {a_1, a_2……a_M} with class variable Q = {Q_1, Q_2,…………Q_M}, representation of hidden connections Z and anonymous testing examples Fu = {d_1, d_2……………d_i}

Training:

- Assign the input weight {w_1, w_2,…………….w_M} and G= {b_1, b_2………b_M }^{T}
- Compute the hidden layer output formula H= f(w. F + G)
- Compute the output weight H^+. Q.

Classification:

- Compute hidden layer output formula of new instances Hu = f (w. Fu + G).
- Find the class label of new examples of Fu: Qu = Hu. h.

We solve the single hidden layer neural network and classify IoT data problems using the ELM algorithm at the cloud layer of the proposed framework. It provides efficient performance for handling the massive amount of data and easily analyzing the big data in IoT applications.

4. Experimental Evaluation

This section discusses the experimental evaluation part to evaluate the proposed framework's adequate performance with high accuracy and low latency. We employed the NSL-KDD dataset consisting of sample events associated with five big data analysis classes, as presented in Table 2. We used KDDTest+, KDDTest-21, and KDDTrain+ dataset and disposed of it in a real-time infrastructure. The details and results of our algorithm are described in the subsection, and significant data analysis methods with machine learning and IoT Applications shown in Table 3 [30].

Table 2. Big Data Analysis Categories of IoT Applications.

Real_Time	Offline	Database Level	Manufacturing Level	Large Level
GreenPlum	Skribe	MongoDB	Data Analysis Plan	MapReduce
HANA	Kafka	TB-Level Data	Distributed File	Scala
Parallel Processing	TumeTunnel		TB-Level Data	
Memory Based	Chukwa			

Evaluation Methodology

Weka [31] tool is used to estimate the experimental analysis of our proposed framework for big data analysis. This tool is mainly used for data tunneling, evaluating the analysis model using various machine learning models. Every data occurrence in the dataset has 41 input qualities, and these are categorized into four types: essential attributes, content attributes, traffic attributes, and host-based attributes. Traffic attributes are time-based, which are extracted from traffic by utilizing different types of windows. 10-fold cross-validation technique, we can apply [31,32] across the assembled NSL-KDD dataset and use two classes normal and anomaly as shown in the confusion matrix Table 4. 10 equal size subsets data are offered with the help of a fold cross-validation approach. In this division, training use nine subset data, and testing use one data set.This process is returned but has one situation that 10 datasets hold as the testing set exactly one time. For ELM classifier, the sigmoidal formula is offered as a hidden outcome operation. We used 50 simulations for the ELM algorithm with some parameter numbers on training data. A big data analysis report is evaluated with a confusion matrix. We practiced a productive workstation with Processor E5-1620 v3 (30 MB, 3.70Ghz processor rate), and the bandwidth specifications have been promoted from 1GE-100 GE.

Table 3. Big Data Analysis Methods and IoT Applications

Methods Applications	**Classification**	**Clustering**	**Association Rule**	**Prediction**	**Time Series**	**Proposed ELM Method**
Social Networking	-	√	-	√	√	√
Bioinformatics	-	√	√	-	-	√
Healthcare	-	√	√	-	-	√
Transportation	-	√	√	-	-	√
Market Analysis	-	√	√	√	-	√
Disaster Management	-	-	-	√	√	√
Speech Recognition	√	-	-	-	√	√
e-governance	√	√	√	-	√	√
Industry Management	√	√	√	-	-	√
Human Genetic	-	√	-	-	-	√
Medical Imagine	√	√	-	-	√	√

Table 4. Confusion Matrix [33]

Actual	Classified	
	Normal	**Anomaly**
Normal	T_p	F_n
Anomaly	F_p	T_n

Where, T_p is quantity of the normal profiles correctly classified as normal profiles as true positive,

F_p is number of anomaly profiles incorrectly classified as normally once as false-positive,

F_n is indicates the quantity of normally profiles incorrectly classified as anomaly once as false-negative,

T_n is the number of anomaly profiles correctly classified as anomaly once as true negative.

Confusion matrix, which is used for evaluating the performance of the proposed framework for big data analysis and used the accuracy, false positive rate, precision, recall, F-measure, MCC, and AUC formulas or equation from 7 to 12 [33].

a) Accuracy (True Positive Rate_ACC):

$$Accuracy = \frac{Tp+Tn}{Tp+Fn\ +Tn+Fp} \quad \textbf{(7)}$$

b) False Positive Rate (FPR):

$$FPR = \frac{Fn+Fp}{Tp+Fn\ +Tn+Fp} \quad \textbf{(8)}$$

c) Precision:

$$Precision = \frac{Tp}{Tp+\ Fp} \quad \textbf{(9)}$$

d) Recall (Detection_Rate):

$$Detection_{Rate}\ = \frac{Tp}{Tp+\ Fn} \quad \textbf{(10)}$$

e) F-Measure:

$$F-Measure = \frac{2*Precision\ *Detection\ \ Rate}{Precision\ +Detection\ \ Rate} \quad \textbf{(11)}$$

f) MCC (Mathew Correction Coefficient):

$$MCC = \frac{Tp*Tn-Fp*Fn}{\sqrt{(Tp+Fn)(Tp+Fp)(Tn+Fp)(Tn+Fn)}} \quad \textbf{(12)}$$

g) The area under the receiver operation curve (AUC) shows the true and false positive sample rate.

4.1. Proposed Framework Evaluation

This subsection shows of our framework's evaluation. We applied KDDTest+ and KDDTest-21 datasets and following different machine learning classifiers: Naive Bayes

(NB) [34, 35], Logistic Regression (LR) [36, 37], Jrip (JR) [38], J48 Decision Tree (J48) [39], LMT Decision Tree (LMT), Random Forest (RF), Support Vector Machine (SMO) [40-42], K-Nearest Neighbors (IBK) [43, 44]. All classifier machine learning methods are notified in Table 5. We used the classification method with the ELM algorithm for classification, K-means algorithm for scaling, and PCA algorithm for feature extraction (proposed in subsection 3.2). The 10-fold cross-validation methods displayed accuracy, FPR, precision, detection rate, F-measure, and others over the NSL-KDD dataset With the NSL-KDD data set, we trained 8 different machine learning classifiers in section 4, then utilized 10-fold cross-validation methods to estimate the results. Fig. 3 and Fig. 5 summarize all classifiers' performance results regarding standard evaluation standards as beforehand reported.

According to Fig. 3 and Fig. 4 for the KDDTest+, KDDTest-21 dataset, all machine learning classifiers have a perfect classification capability to execute. According to the proposed framework's performance on the KDDTest+ dataset, it observed accuracy, FPR, precision value, F-measure is the highest of NB classifier compared to others.NB algorithm has the accuracy 98.8% of the proposed framework on the KDDTest+ dataset and 97.8% on the KDDTest-21 dataset. The latency time is 0.01 sec of the K-Nearest Neighbors algorithm of the KDDTest+ dataset's proposed framework. It is smaller than other; 0.08 sec is the NB algorithm's latency time of the proposed framework on the KDDTest-21 dataset.

There is a slightly different of around 0.1 in detection_rate for the classifier NB, J48, and LMT. Furthermore, the RF and NB classifier obtained a similar value of MCC and AUC. Time taken time for making the model is less for IBK classifier compares to others. According to the proposed framework's performance on the KDDTest-21 dataset, the NB classifier's accuracy is similar to RF classifier and FPR, precision, detection_rate, F-measure, and MCC value is highest of NB classifier compares than others. Time taken time for making the model is less for the NB classifier compares to others. FPR, Precision, detection_rate, F-Measure and MCC value of NB algorithm for proposed framework on KDDTest+ dataset is 15.8%, 98.6%, 98.5%, 98.7%, and 97.4%, it is also greater than another algorithm. Similarly, FPR, Precision, detection_rate, F-Measure, and MCC value of the NB algorithm for the proposed framework on KDDTest+ dataset are 31.5%, 95.6%, 94.0%, 93.9%, and 92.7%; it is also greater than another algorithm.

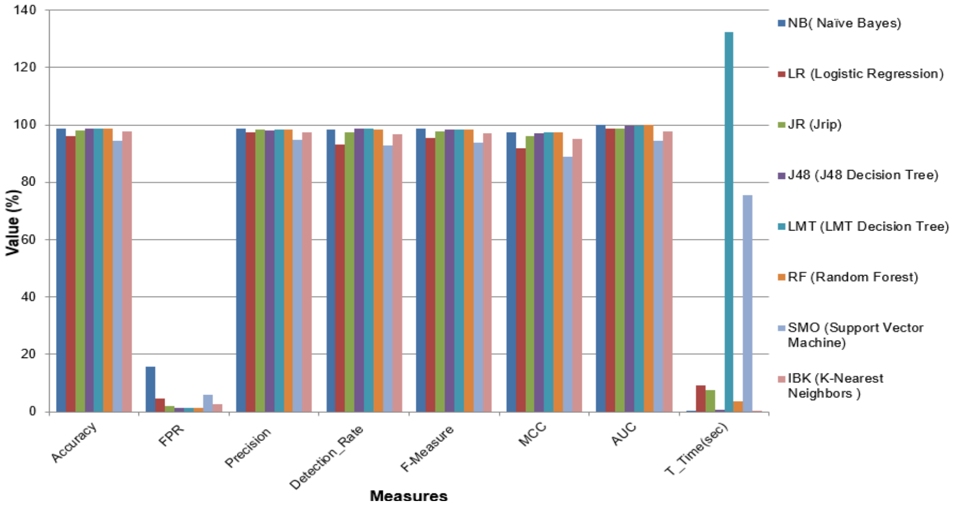

Fig. 3. Proposed framework's evaluation performance on KDDTest+.

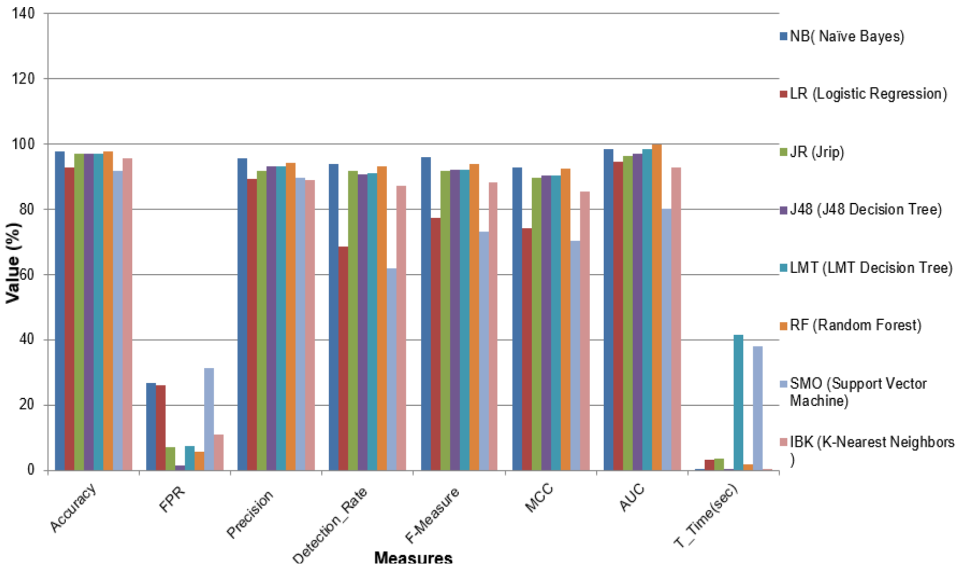

Fig. 4. Proposed framework's evaluation performance on KDDTest-21.

Table 5. Proposed framework's execution for Several IoT applications

Application	Evaluation measure		
	Dataset	**Accuracy (%)**	**Latency(sec)**
Attack Detection	KDDTest+	86.53	0.011
	KDDTest-21	75.77	0.013
Object Detection	MNIST	84.2	0.057
	MS-COCO	78.32	0.048

The performance of the proposed framework of IoT applications, such as attack detection and object detection, is shown in Table 5. Attack detection evaluated on KDDTest+ and KDDTest-21 data set. Accuracy is 86.53% on KDDTest+ and 75.77% on KDDTest-21 data set. Latency time is 0.011sec on KDDTest+ dataset and 0.013sec on KDDTest-21 data set. Object detection was evaluated on MNIST and MS-COCO datasets. Accuracy is 84.2% on MNIST and 78.32% on the MS-COCO data set. Latency time is 0.057sec on the MNIST dataset and 0.048sec on the MS-COCO data set. Comparison with existing research is shown in Table 6 with methodology, security architecture, technology aspects, dataset labeling requirements, and conventional machine learning classifiers.

Table 6. Comparison with existing research studies

Research Work	Year	Security Architecture	Technology Aspects	Methodology	Requirement of dataset labeling	Conventional Machine Learning Classifier
Li et al. [11]	2018	Centralized	IoT	The offloading approach with edge computing is used to optimize the performance of IoT applications	High	No
Peng Li et al. [12]	2018	Cluster-based architecture	IoT	A deep convolutional computational model for big data features learning using tensor representation model	High	Yes
Chhowa et al. [13]	2019	Centralized	IoT	Deep Machine learning for big data analysis	Low	No
Mishra et al. [14]	2015	Centralized	IoT	Cognitive Oriented IoT big data framework	Low	Yes
Zhang et al. [15]	2018	Centralized with BGV encryption	IoT	Double projection deep computational model for bigdata feature learning	High	Yes
Hosseini et al. [16]	2016	Centralized	Epileptic Seizure Prediction	Deep learning with cloud for Epileptic Seizure Prediction	Low	No
Vinay et al. [17]	2015	Centralized	Social Network	For face identification, a cloud-based big data analytics framework is used	Low	Yes
Ying et al. [18]	2017	Centralized	Smart City	Deep reinforcement learning approach with SDN and mobile edge computing	High	No
Liangzhi Li et al. [8]	2018	Centralized	IoT	Distributed deep model for mobile crowdsensing	High	Yes
Singh et al. [39]	2020	Distributed	Smart City	Provide IoT oriented infrastructure for the smart city based on deep learning and blockchain	Medium	No
Proposed work	2020	Distributed	Next Generation Web in IoT	Machine learning-based distributed big data analysis framework for Next Generation Web in IoT	Less	No

However, the proposed work encourages a distributed framework for big data analysis. It is suitable for advanced applications. It needs less labeled data to promote big data interpretation with precision and time detection in advanced applications. It provides higher execution over another significant data analysis time detection, and accuracy. Table 3 shows big data analysis methods and IoT applications via machine learning methods such as classification, clustering, association rule, prediction, and time series [45-47]. Proposed ELM methods are used in maximum IoT applications such as social networking, bioinformatics, smart energy, smart home, e-government, and others compare to other machine learning methods [48-52]. ELM based framework on the cloud layer provides excellent performance at a high data rate.

5. Conclusion

This paper proposed a machine learning-based distributed big data analysis framework for Next Generation Web in IoT. It relies on the edge layer with feature extraction and scaling and cloud layer with the ELM algorithm to facilitate real-time big data analysis and classification in IoT. The PCA algorithm used for feature extraction, the K-means algorithm for scaling, and the ELM algorithm for classification. Analysis of big IoT data and provides a distributive capability wherein extract, scale, and classified IoT data features at the edge layer and improved data fulfillment. Furthermore, the recommended approaches solve big data analysis issues by combining the feature extraction and scaling algorithms. The experimental evaluation on the KDDTest+ and KDDTest-21 dataset determines that the framework realized excellent performance 0.06sec time taken for making the model, 98.8% accuracy rate for KDDTest+ dataset, and 0.08sec time taken for making the model, 97.8% accuracy rate for the KDDTest-21 dataset. Comparing the proposed machine learning-based big data analysis framework approach with the conventional machine learning classifier shows that it can manage label data issues and attain better appearance to the other classifier. The ELM-based method was used in the cloud layer and improved the proposed framework's performance and big data analysis issues easily using a distributed cloud. Finally, we compare the proposed framework with existing research according to methodology, security architecture, technology aspects, dataset labeling requirements, and conventional machine learning classifiers.

We have two ways according to our proposed framework for the future. First, we expect to explore other improvements in our frameworks' advanced applications, such as using machine learning models to eradicate manual feature extraction and scaling for big data analysis. Second, we intend to construct our NSL-KDD dataset to developing a common framework across big data analysis.

Acknowledgment. This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology).

References

1. Zantalis, F., Koulouras, G., Karabetsos, S., Kandris, D.: A Review of Machine Learning and IoT in Smart Transportation. Future Internet, Vol. 11, No. 4, 94. (2019)
2. Keenan, M: The Future of Data with the Rise of the IoT. [Online]. Available:https://www.rfidjournal.com/articles/view?17954(Accessed in October 2018)
3. Verma, A.: Internet of Things and Big Data- Better Together. [Online]. Available: https://www.whizlabs.com/blog/iot-and-big-data/ (Accessed on August 2018)
4. Li, Y., Gao, H., Xu, Y.: Special Section on Big Data and Service Computing. Intelligent Automation and Soft Computing, Vol. 25, No. 3, 511-512. (2019)

5. Zhang, Y., Wang, Y. G., Bai, Y. P., Li, Y. Z., Lv, Z. Y., Ding, H. W.: A New Rockburst Experiment Data Compression Storage Algorithm Based on Big Data Technology. Intelligent Automation and Soft Computing, Vol. 25, No. 3, 561-572.(2019)
6. Liu, M., Cheng, L., Qian, K., Wang, J., Liu, Y.: Indoor acoustic localization: a survey. Human-centric Computing and Information Sciences, Vol. 10, No. 1, 2. (2020)
7. Singh, S.K., Rathore, S., Park, J. H.: BlockIoTIntelligence: A Blockchain-enabled Intelligent IoT Architecture with Artificial Intelligence, Future Generation Computer Systems, Vol. 110, 721-743. (2019)
8. Abeshu, A., &Chilamkurti, N.: Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Communications Magazine, Vol. 56, No. 2, 169-175. (2018)
9. Park, J. H., Salim, M. M., Jo, J. H., Sicato, J. C. S., Rathore, S., & Park, J. H.: CIoT-Net: a scalable cognitive IoT based smart city network architecture. Human-centric Computing and Information Sciences, Vol. 9, No.1, 29. (2019)
10. Singh, S. K., Salim, M.M., Cha, J., Pan, Y., Park, J. H.: Machine Learning-Based Network Sub-Slicing Framework in a Sustainable 5G Environment. Sustainability, Vol. 12, 6250. (2020)
11. Li, L., Ota, K., Dong, M.: Human in the Loop: Distributed Deep Model for Mobile Crowdsensing. IEEE Internet of Things Journal, Vol. 5, No. 6, 4957-4964. (2018)
12. Mohammadi, M., Al-Fuqaha, A., Sorour, S., Guizani, M.: Deep learning for IoT big data and streaming analytics: A survey. IEEE Communications Surveys & Tutorials, Vol. 20, No.4, 2923-2960. (2018)
13. Hu, L., Ni, Q.: IoT-driven automated object detection algorithm for urban surveillance systems in smart cities. IEEE Internet of Things Journal, Vol. 5, No. 2, 747-754. (2017)
14. Li, H., Ota, K., & Dong, M.: Learning IoT in edge: Deep learning for the Internet of Things with edge computing. IEEE Network, Vol. 32, No. 1, 96-101. (2018)
15. Li, P., Chen, Z., Yang, L. T., Zhang, Q., Deen, M. J.: Deep convolutional computation model for feature learning on big data in Internet of Things. IEEE Transactions on Industrial Informatics, Vol. 14, No. 2, 790-798. (2017)
16. Chhowa, T. T., Rahman, M. A., Paul, A. K., Ahmmed, R.: A Narrative Analysis on Deep Learning in IoT based Medical Big Data Analysis with Future Perspectives. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) 1-6. (2019)
17. Mishra, N., Lin, C. C., Chang, H. T.: A cognitive adopted framework for IoT big-data management and knowledge discovery perspective. International Journal of Distributed Sensor Networks, Vol. 11, No. 10, 718390. (2015).
18. Zhang, Q., Yang, L. T., Chen, Z., Li, P., Deen, M. J.: Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning. IEEE Internet of Things Journal, Vol. 5, No. 4, 2896-2903. (2017)
19. Hossain, B., Morooka, T., Okuno, M., Nii, M., Yoshiya, S. and Kobashi, S.: Surgical Outcome Prediction in Total Knee Arthroplasty using Machine Learning. Intelligent Automation and Soft Computing, Vol. 25, No. 1, 105-115. (2019)
20. Vinay, A., Shekhar, V. S., Rituparna, J., Aggrawal, T., Murthy, K. B., & Natarajan, S.: Cloud-based big data analytics framework for face recognition in social networks using machine learning. Procedia Computer Science, Vol. 50, 623-630. (2015)
21. He, Y., Yu, F. R., Zhao, N., Leung, V. C., Yin, H.: Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach. IEEE Communications Magazine, Vol. 55, No. 12, 31-37. (2017)
22. Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Yunsheng, M., Hou, P.: A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. IEEE Transactions on Services Computing, Vol. 11, No. 2, 249-261. (2017)
23. Jeong, Y. S., Park, J. H.: Advanced Big Data Analysis, Artificial Intelligence & Communication Systems. Journal of Information Processing Systems, Vol. 15, No. 1. (2019)

24. Mahdavinejad, M. S., Rezvan, M., Barekatain, M., Adibi, P., Barnaghi, P., Sheth, A. P.: Machine learning for Internet of Things data analysis: A survey. Digital Communications and Networks, Vol. 4, No. 3, 161-175. (2018)
25. Tang, Z., Liu, K., Xiao, J., Yang, L., Xiao, Z.: A parallel k-means clustering algorithm based on redundance elimination and extreme points optimization employing MapReduce. Concurrency and Computation: Practice and Experience, 29(20), e4109. (2017)
26. Sun, W., Du, H., Nie, S., He, X.: Traffic Sign Recognition Method Integrating Multi-Layer Features and Kernel Extreme Learning Machine Classifier, Computers, Materials & Continua, Vol. 60, No. 1, 47-161. (2019)
27. Cao, J., & Lin, Z.: Extreme learning machines on high dimensional and large data applications: a survey. Mathematical Problems in Engineering, (2015). https://doi.org/10.1155/2015/103796
28. Duan, M., Li, K., Liao, X., Li, K.: A parallel multiclassification algorithm for big data using an extreme learning machine. IEEE transactions on neural networks and learning systems, Vol. 29, No. 6, 2337-2351. (2017)
29. Akusok, A., Björk, K. M., Miche, Y., Lendasse, A.: High-performance extreme learning machines: a complete toolbox for big data applications. IEEE Access, Vol 3, 1011-1025. (2015)
30. Marjani, M., Nasaruddin, F., Gani, A., Karim, A., Hashem, I. A. T., Siddiqa, A., Yaqoob, I.: Big IoT data analytics: architecture, opportunities, and open research challenges. IEEE Access, Vol. 5, 5247-5261. (2017)
31. Rathore, S., Loia, V., & Park, J. H.: SpamSpotter: An efficient spammer detection framework based on an intelligent decision support system on Facebook. Applied Soft Computing, Vol. 67, 920-932. (2018)
32. Luo, M., Wang, K., Cai, Z., Liu, A., Li, Y., Cheang, C.F.: Using Imbalanced Triangle Synthetic Data for Machine Learning Anomaly Detection, Computers, Materials & Continua, Vol. 58, No. 1, 15-26, (2019)
33. Rathore, S., Park, J. H.: Semi-supervised learning based distributed attack detection framework for IoT. Applied Soft Computing, Vol. 72, 79-89. (2018)
34. Zhuang, X., Zhou, S.: The Prediction of Self-Healing Capacity of Bacteria-Based Concrete Using Machine Learning Approaches, Computers, Materials & Continua, Vol. 59, No. 1, 57-77. (2019)
35. Ma, T., Pang, S., Zhang, W., Hao, S.: Virtual Machine Based on Genetic Algorithm Used in Time and Power Oriented Cloud Computing Task Scheduling. Intelligent Automation and Soft Computing, Vol. 25, No. 3, 605-613. (2019)
36. Lin, J., Yin, J., Cai, Z., Liu, Q., Li, K., Leung, V.: A secure and practical mechanism of outsourcing extreme learning machine in cloud computing. IEEE Intelligent Systems, Vol. 28, No. 6, 35-38. (2013)
37. Peng, F., Zhou, D. L., Long, M., Sun, X. M.: Discrimination of natural images and computer-generated graphics based on multi-fractal and regression analysis. AEU-International Journal of Electronics and Communications, Vol. 71, 72-81. (2017)
38. Chernick, M. R., Murthy, V. K., &Nealy, C. D.: Application of bootstrap and other resampling techniques: Evaluation of classifier performance. Pattern Recognition Letters, Vol. 3, No. 3, 167-178. (1985)
39. Singh, S. K., Jeong, Y. S., & Park, J. H.: A Deep Learning-based IoT-oriented Infrastructure for Secure Smart City. Sustainable Cities and Society, Vol. 60, 102252. (2020)
40. Kuang, F., Zhang, S., Jin, Z., & Xu, W. A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection. Soft Computing, Vol. 19, No. 5, 1187-1199. (2015)
41. Duan, L., Han, D., & Tian, Q.: Design of intrusion detection system based on improved ABC_elite and BP neural networks. Computer Science and Information Systems, Vol. 16, No. 3, 773-795. (2019)

42. Zhang, L. B., Peng, F., Qin, L., & Long, M.: Face spoofing detection based on color texture Markov feature and support vector machine recursive feature elimination. Journal of Visual Communication and Image Representation, Vol. 51, 56-69. (2018)
43. Tanwar, S., Tyagi, S., Kumar, S.: The role of internet of things and smart grid for the development of a smart city. In Intelligent Communication and Computational Technologies 23-33. (2018)
44. Qi, E., Deng, M.: R&D investment enhance the financial performance of company driven by big data computing and analysis. Computer Systems Science and Engineering, Vol. 34, No. 4, 237-248. (2019)
45. Kumari, A., Tanwar, S., Tyagi, S., Kumar, N., Parizi, R. M., & Choo, K. K. R.: Fog data analytics: A taxonomy and process model. Journal of Network and Computer Applications, Vol. 128, 90-104. (2019).
46. Wang, T., Wen, L., Zhou, Y., & Zhang, J.: A energy balanced routing scheme in wireless sensor networks based on non-uniform layered clustering. International Journal of Sensor Networks, Vol. 27, No. 4, 239-249. (2018)
47. Oh, B.D., Song, H.J., Kim, J.D., Park, C.Y., Kim, Y.S.: Predicting Concentration of PM10 Using Optimal Parameters of Deep Neural Network. Intelligent Automation and Soft Computing, Vol. 25, No. 2, 343-350. (2019)
48. Azzaoui, A. E., Singh, S. K., Pan, Y., & Park, J. H.: Block5GIntell: Blockchain for AI-Enabled 5G Networks. IEEE Access. (2020) https://doi.org/10.1109/ACCESS.2020.3014356
49. Yang, H., Yi, J., Zhao, J., Dong, Z.: Extreme learning machine based genetic algorithm and its application in power system economic dispatch. Neurocomputing, Vol. 102, 154-162. (2013)
50. Cha, J., Singh, S.K., Pan, Y., Park, J. H.: Blockchain-Based Cyber Threat Intelligence System Architecture for Sustainable Computing. Sustainability, Vol. 12, 6401. (2020)
51. Sicato, J. C. S., Singh, S. K., Rathore, S., Park, J. H.: A Comprehensive Analyses of Intrusion Detection System for IoT Environment. Journal of Information Processing Systems, Vol. 16, No. 4, 975-990, 2020. (2020)
52. Zhang, T., Hou, M., Zhou, T., Liu, Z., Cheng, W., & Cheng, Y.: Land-use classification via ensemble dropout information discriminative extreme learning machine based on deep convolution feature. Computer Science and Information Systems, Vol. 1, No. 00, 10-10. (2020)

Sushil Kumar Singh received his M.Tech. Degree in Computer Science and Engineering from Uttarakhand Technical University, Dehradun, India. Currently, he is pursuing his Ph.D. degree under the supervision of Prof. Jong Hyuk Park at the Ubiquitous Computing Security (UCS) Laboratory, Seoul National University of Science and Technology, Seoul, South Korea.His current research interests include Blockchain, Artificial Intelligence, Big Data, and the Internet of Things.

Jeonghun Cha received the B.S. degree in computer science from DongyangMirae University, Seoul, South Korea. He is currently pursuing the master’s degree in computer science and engineering with the Ubiquitous Computing Security (UCS) Laboratory, Seoul National University of Science and Technology, Seoul, South Korea, under the supervision of Prof. Jong Hyuk Park. His current research interests include Information security, Cyber Threat Intelligence, and Internet-of-Things (IoT) security.

Tae Woo Kim received the B.S. degree in computer science from Kumoh National Institute of Technology, Gumi, South Korea. He is currently pursuing the master’s degree in computer science and engineering with the Ubiquitous Computing Security

(UCS) Laboratory, Seoul National University of Science and Technology, Seoul, South Korea, under the supervision of Prof. Jong Hyuk Park. His current research interests include Cloud security, Software Defined Network, and Internet-of-Things (IoT) security.

Dr. Jong Hyuk Park received Ph.D. degrees in Korea University, Korea and Waseda University, Japan. He is now a professor at the Department of Computer Science and Engineering, Seoul National University of Science and Technology, Korea. Dr. Park has published about 300 research papers in international journals and conferences. He is editor-in-chief of Human-centric Computing and Information Sciences (HCIS) by Springer, The Journal of Information Processing Systems (JIPS) by KIPS. His research interests include IoT, Information Security, Smart City, Blockchain, etc. Contact him at jhpark1@seoultech.ac.kr.

Received: March 30, 2020; Accepted: January 19, 2021.