Computer Science and Information Systems 21(4):1389–1410 https://doi.org/10.2298/CSIS240401041S

BLSAE-SNIDS: A Bi-LSTM Sparse Autoencoder Framework for Satellite Network Intrusion Detection

Shi Shuxin[2], Han Bing[1], Wu Zhongdai[1,3], Han Dezhi[2], Wu Huafeng[4,⋆], and Mei Xiaojun[4]

[1] State Key Laboratory of Maritime Technology and Safety
200135 Shanghai, China
han.bing@coscoshipping.com
[2] College of information Engineering, Shanghai Maritime University
201306 Shanghai, China
dezhihan88@sina.com.cn
shishuxin@stu.shmtu.edu.cn
[3] Shanghai Ship and Shipping Research Institute Co.,Ltd.
200135 Shanghai, China
wu.zhongdai@coscoshipping.com
[4] Merchant marine college, Shanghai Maritime University
201306 Shanghai, China
xjmei94@163.com, hfwu@shmtu.edu.cn

Abstract. Due to disparities in tolerance, resource availability, and acquisition of labeled training data between satellite-terrestrial integrated networks (STINs) and terrestrial networks, the application of traditional terrestrial network intrusion detection techniques to satellite networks poses significant challenges. This paper presents a satellite network intrusion detection system named Bi-LSTM sparse self-encoder (BLSAE-SNIDS) to address this issue. Through the development of an innovative unsupervised training Bi-LSTM stacked self-encoder, BLSAE-SNIDS facilitates feature extraction from satellite network traffic, diminishes dimensionality, considerably reduces training and testing durations, and enhances the attack prediction accuracy of the classifier. To assess the efficacy of the proposed model, we conduct comprehensive experiments utilizing STIN and UNSW-NB15 datasets. The results obtained from the STIN dataset demonstrate that BLSAE-SNIDS achieves 99.99% accuracy with reduced computational and transmission overheads alongside enhanced flexibility. Furthermore, results from the UNSW-NB15 dataset exhibit BLSAE-SNIDS' proficiency in detecting various network intrusion attacks efficiently. These findings indicate that BLSAE-SNIDS suits general satellite security networks and offers a novel approach to designing security systems for polar satellite networks, thus exhibiting practical utility.

Keywords: Satellite-terrestrial integrated networks, LSTM, Automatic encoder, Unsupervised learning, Network security, Deep learning.

1. Introduction

As a significant augmentation to terrestrial networks, satellites extend the coverage of such networks, facilitating convenient access for remote users. Consequently, the prevail-

⋆ Corresponding Author

ing concept in sixth-generation mobile communications (6G) is to amalgamate satellite networks with terrestrial networks to construct an STIN system to achieve seamless global coverage. Nevertheless, STIN presents inherent security vulnerabilities, primarily due to satellite node exposure and the utilization of long-distance communication links spanning distances between 150 and 35,800 km [1]. These factors make the system susceptible to cyber-attacks, engendering substantial security risks [2, 3]. As 6G technology continues to advance, the security of STIN has emerged as a paramount concern in recent years.

Due to their extensive coverage, high capacity, and long-distance transmission capabilities, satellite communication networks find widespread applications in navigation, spaceflight, radio and television broadcasting, disaster rescue, and relief. Particularly in remote areas characterized by limited services and dispersed populations, satellite communication emerges as the optimal choice due to its independence from substantial ground infrastructure investments.

The Arctic shipping lanes, pivotal for maritime navigation, pose challenges in relying on ship-to-shore communications for navigation and hazard warnings. These challenges arise from inadequate data on the navigational environment in the Arctic, a lag in communications infrastructure, and a dearth of navigational aids such as beacons, lighthouses, and coastal radio services. Consequently, due to limited experience in Arctic navigation, ships operating in the region are exposed to elevated safety risks. To foster endeavors such as environmental conservation, resource utilization, and commercial transportation in the Arctic, it is imperative to conduct thorough research on communication and navigation support systems tailored for Arctic shipping lanes, aiming to enhance safety during polar navigation. Introducing the message service of a satellite navigation system presents a novel approach to information transmission in the Arctic region.

However, as the network scale expands, the vulnerability of satellite communication transmission links to intrusion grows. Regrettably, a substantial disparity exists between satellite and terrestrial networks concerning computing power, bandwidth, and other resources. Furthermore, upgrading satellite hardware post-launch poses significant challenges. Hence, it is necessary to devise effective Star-Terrestrial Network Intrusion Detection Systems (SAT-IDS) to afford robust protection for STIN.

The primary objective of a Network Intrusion Detection System (NIDS) is the detection of harmful intrusions, typically employed to monitor network traffic, distinguishing between normal and malicious activity to effectively mitigate the latter. Designing NIDSs for satellite networks necessitates consideration of the limited computational power of each satellite node, alongside stringent security and privacy requirements during transmission, thereby emphasizing the importance of resource utilization efficiency [4–6].

Numerous design approaches for NIDSs have been proposed, often leveraging machine learning algorithms to discern traffic patterns [7,8]. Evaluating NIDS efficacy hinges on its ability to accurately detect attacks, demanding comprehensive datasets encompassing both normal and abnormal behaviors. However, applying such datasets from terrestrial networks directly to satellite communications presents challenges owing to the unique characteristics of satellite networks.

1. Satellite and terrestrial networks exhibit differing susceptibilities to various attack types. While terrestrial networks may grapple with challenges such as Backdoor and Botnets, exacerbated by the computational prowess and openness of terrestrial network nodes [9–11] , satellite nodes with constrained resources face concerns over

malicious distributed denial of service (DDoS) attacks, such as Synchronized Sequence Number (Syn) flooding. Accordingly, formulating secure datasets for distinct satellite network domains warrants meticulous consideration.
2. Furthermore, the limited resources and computational capabilities inherent in satellite networks pose additional challenges, as satellite nodes, when targeted, can swiftly succumb to attacks, making recovery arduous.
3. Extracting pertinent feature selections from satellite network traffic is inherently challenging. The intricacies of satellite networks, coupled with the costliness of communication links, render the acquisition of large training datasets prohibitive.

Addressing the challenges inherent in existing satellite network intrusion detection methods, this paper presents a novel approach termed Bi-LSTM sparse autoencoder-based Satellite Network Intrusion Detection System (BLSAE-SNIDS). The system leverages autoencoders for feature learning on unlabeled data, thereby generating new feature representations in an unsupervised manner. These features are inputted into a classifier following a pre-training phase to enhance intrusion detection capability and classification accuracy. The contributions of this study are outlined as follows:

1. Given the intricacies of satellite networks, obtaining ample labeled data for learning purposes proves exceedingly difficult and costly. Consequently, we propose BLSAE-SNIDS, which employs an overarching architecture based on Bi-LSTM sparse autoencoders capable of extracting implicit knowledge from unlabeled data and learning unsupervised representations. This approach enables superior classification outcomes with a limited number of labels.
2. Utilizing the Bi-LSTM sparse autoencoder effectively reduces the dimensionality of input vectors in the classification task. This operation can be performed within the network nodes of the satellite network, substantially curbing the volume of traffic transmitted to the satellite network's control center. Moreover, the computational nodes need not be equipped with high-performance hardware to swiftly detect known and unknown intrusion traffic. This outcome holds significant value, considering the precious resources of satellite network links.
3. The robustness of the Intrusion Detection System (IDS) is evaluated using a real STIN dataset. Extensive experimentation on the STIN dataset demonstrates that the proposed model adeptly detects various satellite network attacks.

2. Related Work

The expansion of satellite networks has brought about considerable security challenges, prompting the development of security technologies, notably Network Intrusion Detection Systems (NIDS). Numerous studies have employed diverse machine learning techniques to construct effective NIDS for detecting attacks in satellite and terrestrial networks. Di et al. [12] proposed a defense architecture against Distributed Denial of Service (DDoS) attacks in satellite networks, combining distributed multipoint detection, proximity source defense, collaborative management, and protection integrity. Additionally, Li et al. [9] devised security datasets for STIN satellite and terrestrial networks, presenting a distributed NIDS employing federated learning (FL) in STIN. This system allocates resources effectively across domains to analyze and block malicious traffic, particularly DDoS attacks.

Ashraf et al. [13] introduced a novel intrusion detection approach, merging Random Forest (RF) and Multi-Layer Perceptron (MLP) using data from satellite and terrestrial networks. This method enhances intrusion detection accuracy compared to other models, validated through experiments on NSL-KDD, KDD-CUP 99, and STIN datasets. For DDoS attacks in Flying Self-Organizing Networks (FANET), Zhang et al. [14] investigated the operational features of satellite nodes, establishing security domains based on login and logout mechanisms. They proposed a hierarchical distributed satellite network intrusion detection model and a collaborative mechanism for intrusion detection agents inside and outside the satellite. Guo et al. [15] introduced a blockchain-based Distributed Collaborative Entrance Defense (DCED) framework to shield the satellite network from DDoS attacks. This system records and aggregates network traffic characteristics at the satellite network's entrance, bolstering its defense capabilities. Azar et al. [16] proposed a hybrid intrusion detection system for Satellite Terrestrial communication systems (STIN) and terrestrial networks (UNSW-NB15). They employed a Random Forest (RF) based Sequential Forward Feature Selection (SFS) method to select critical features from the dataset. Experimental results demonstrate a notable enhancement in detection accuracy and computational efficiency. The SFS-RF model, employing 10 selected features, achieved 90.5% accuracy on the STIN dataset and 78.52% on the UNSW-NB15 dataset. The RF-SFS-GRU model excelled in deep learning, attaining 87% and 79% accuracy on the respective datasets.

The Satellite Intrusion Detection System (SAT-IDS) uses machine learning to differentiate between normal and abnormal network traffic. Various machine learning approaches have been explored in SAT-IDS development, aiming to conduct feature selection tasks to extract pertinent features from satellite network traffic datasets, thereby enhancing classification outcomes. Nonetheless, acquiring extensive labeled training sets poses a significant challenge due to the intricate nature of satellite networks and the costly communication links involved. Consequently, self-supervised learning and unsupervised feature learning have emerged as viable alternatives. In these methodologies, the algorithm initially learns a robust feature representation from vast amounts of unlabeled data. Later, when addressing a specific classification task, this learned feature representation can be applied to labeled data to solve the classification problem through supervised learning. Unlike supervised learning, unsupervised learning does not necessitate assuming that the unlabeled data shares the same distribution as the labeled data, rendering the model more adaptable. Zhu et al. [17] proposes a flexible and innovative framework for satellite network intrusion detection systems, leveraging deep learning techniques. Here, a sparse autoencoder serves as a network feature extractor. However, it is noteworthy that the research is confined to system design and has not been executed practically.

Therefore, this paper proposes a Satellite Network Intrusion Detection System (BLSAE-SNIDS) based on a Bi-LSTM sparse self-encoder. The approach leverages unlabeled data to train the Bi-LSTM sparse self-encoder for feature extraction from satellite network traffic while reducing dimensionality. By reusing the self- encoder's encoder part and integrating it with a classifier for data classification, the model's parameters and training time can be significantly reduced while enhancing its generalization capacity. The method acquires implicit knowledge from unlabeled data through unsupervised learning, facilitating superior classification outcomes even with limited labeled data. This effec-

tively differentiates between normal and abnormal traffic for satellite network intrusion detection.

3. Methods

3.1. Model Overview

This paper introduces a satellite network intrusion detection method based on a Bi-LSTM sparse autoencoder, as depicted in Fig. 1. The proposed method employs unlabeled data to train the Bi-LSTM sparse autoencoder, leveraging the temporal characteristics of satellite network traffic data. Feature extraction and dimensionality reduction are achieved through the encoder section of the Bi-LSTM sparse autoencoder, followed by reconstruction via the decoder section. Subsequently, the output of the trained encoder undergoes classification, mapping to distinct classes through a classifier. This approach effectively learns essential features from input data, reducing model parameters and training time by reusing the autoencoder's encoder segment. Consequently, the method demonstrates proficiency in distinguishing between normal and abnormal traffic for satellite network intrusion detection.

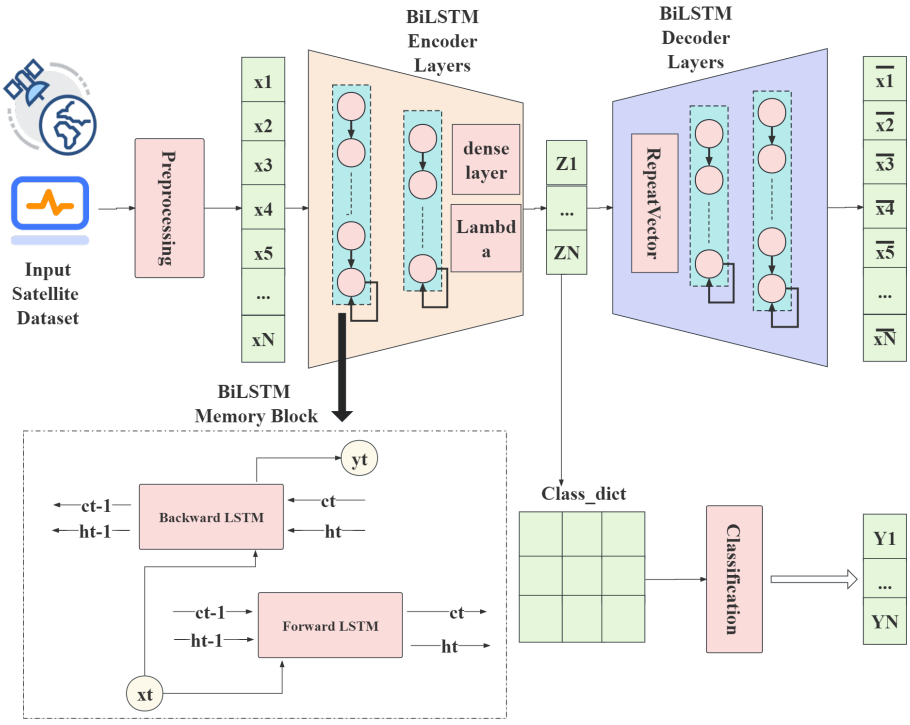

Fig. 1. BLSAE-SNIDS structure diagram

The Bi-LSTM encoder comprises two bidirectional LSTM layers with 64 and 32 hidden units, respectively. It takes a series of high-dimensional input data, efficiently preserves dependencies between multiple data points in the time series, and produces a fixed-size vector. This process reduces the high-dimensional input vector representation to a lower dimension. In our model, the Bi-LSTM encoder consists of two layers: the first with 64 LSTM units and the second with 32 LSTM units. This is succeeded by a Lambda layer with 'n' units and a dense layer with 'n' units, where 'n' represents the number of categories. The Lambda layer maps the encoder output to one of the 'n' categories through a predefined mapping.

The Bi-LSTM decoder generates a fixed-size input sequence using a simplified representation of the input data in the hidden space. During reconstruction, the decoder minimizes the difference, i.e., the reconstruction error, between the original and reconstructed data. The input to the decoder is the dimensionality-reduced output of the encoder. The decoder unfolds the encoding by stacking it in the reverse order of the encoder. It consists of multiple repetition vector layers replicating the encoder output 16 times, followed by two bidirectional LSTM layers with 32 and 64 hidden units, respectively, and a time-distributed dense layer with 1 hidden unit. The final decoder output is a reconstruction of the encoder input.

The class dictionary (Class_dict) matrix records the activation count of each neuron in the encoder for each class label. By selecting the neuron with the highest count for each class in the class dictionary, we determine the relevant class for each encoded sample.

3.2. Bi-LSTM stacked autoencoder

Bi-LSTM LSTM, a temporal recurrent neural network, is adept at processing and predicting significant events with long intervals and time series data delays. Bi-LSTM, linking two hidden layers operating in both input-to-output directions, is proposed for classifying satellite network traffic. Its structure enables efficient utilization of future and past features within specific time ranges during training, thereby enhancing prediction accuracy. The Bi-LSTM layer effectively handles time series problems and improves prediction accuracy. Additionally, the auto-encoder (AE) layer in our proposed structure learns features from data and reduces data dimensionality, complemented by the Bi-LSTM layer to further enhance prediction accuracy [18, 19]. LSTM networks excel in characterizing time series data. Unidirectional LSTM methods may yield errors by relying solely on forward memory processing, potentially failing to capture contextual information. The LSTM layer consists of a series of LSTM units that only process input data in the forward direction. In contrast, Bi-LSTM includes additional LSTM layers processing data in reverse, as illustrated in Fig. 2. Training a Bi-LSTM network is akin to concurrently training two independent unidirectional LSTM networks. One network is trained on the original input sequence, while the other is trained on a reversed copy. This approach furnishes the network with more contextual information, fostering faster and more comprehensive problem learning.

The BiLSTM method extends traditional LSTM by incorporating sequence information in forward (past to future) and backward (future to past) directions, enhancing sequence classification performance. BiLSTM networks have emerged as superior solutions for capturing interactions between contexts. Training BiLSTM networks involves simultaneously training two unidirectional LSTM networks on the input sequence. One LSTM

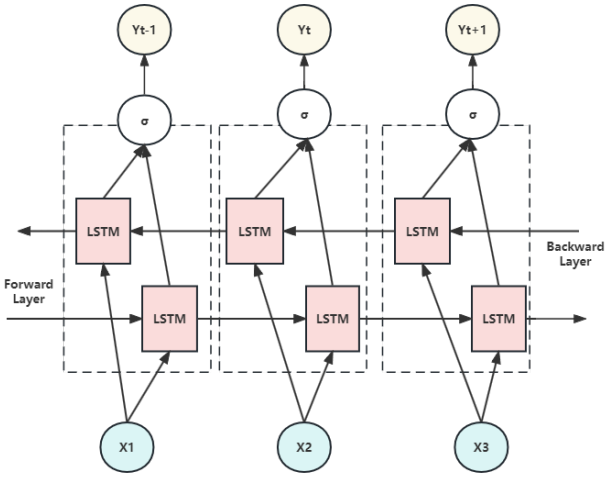

Fig. 2. Expanded view of Bi-LSTM

network processes the input sequence, while the other operates on a reverse copy of the input sequence. This approach provides the network with additional background information, facilitating faster and more comprehensive problem learning.

As depicted in Fig. 2, the LSTM unit comprises forgetting gates, input gates, and output gates to facilitate effective short-term and long-term memorization. Specifically, BiLSTM memory blocks are trained in two directions: one LSTM proceeds from past to future. At the same time, the other progresses from future to past, ensuring that each time step incorporates both past and future data. Each layer computes the respective function for each element in the input sequence. The variables associated with LSTM units are represented accordingly.

$$\begin{aligned} i_t &= \sigma\left(W_{ii}x_t + b_{ii} + W_{hi}h(t \quad 1) + b_{hi}\right) \\ f_t &= \sigma\left(W_{if}x_t + b_{if} + W_{hf}h(t \quad 1) + b_{hf}\right) \\ g_t &= \tanh\left(W_{ig}x_t + b_{ig} + W_{hg}h(t \quad 1) + b_{hg}\right) \\ o_t &= \sigma\left(W_{io}x_t + b_{io}W_{hp}h(t \quad 1) + b_{ho}\right) \\ c_t &= f_t \quad c(t \quad 1) + i_t \quad g_t \\ h_t &= o_t{}^* \tanh\left(c_t\right) \end{aligned} \tag{1}$$

where$(W_{ii}, W_{if}, W_{ig}, W_{io})$ is the input weights, $(W_{hi}, W_{hf}, W_{hg}, W_{ho})$ is the cyclic weights, $(b_{hi}, b_{hf}, b_{hg}, b_{ho})$ is the cyclic bias, h_t denotes the hidden state at moment t, c_t denotes the cell state at moment t, x_t denotes the input at moment t, $h\,(t \quad 1)$ denotes the hidden state of the layer at moment (t-1) or the initial hidden state at moment 0, and i_t , f_t , g_t , o_t denote the input gates, forgetting gates, cell gates and outputs, respectively. gates. is the sigmoid function and ' ' is the Hadamard product.

Automatic encoder The autoencoder is a neural network architecture employed in unsupervised learning, data compression, downscaling, and data generation. It compresses

input data to obtain a concise representation (encoding) and then reconstructs the original data with minimal information loss (decoding). Typically, a standard autoencoder comprises an input layer, an output layer, and multiple hidden layers. Fig. 3 illustrates a basic autoencoder model, which can be elucidated by assessing the reconstruction error (loss) between the encoder, decoder, input, and output data [20–22].

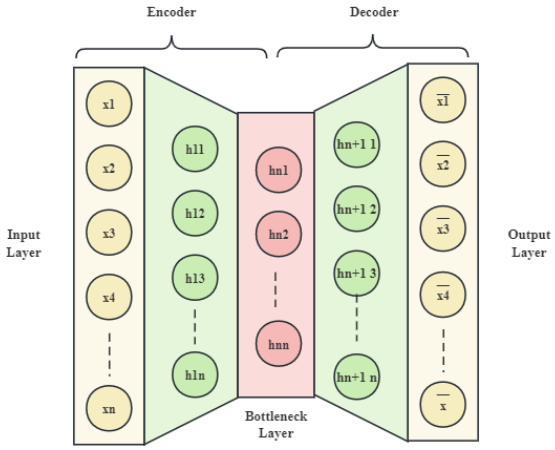

Fig. 3. A simple autoencoder structure

Encoding: The encoding operation maps the input data x, i.e., the high-dimensional vector $x \ 2 \ R^m$, to a low-dimensional bottleneck layer representation (h). This is given by the following equation, where w_i is the weight matrix, b_i is the bias, and f_1 is the activation function.

$$h = f_1 (w_i x + b_i) \tag{2}$$

Decoding: Eq. (3) illustrates how the decoding operation uses the bottleneck layer representation (h) to produce the output of the attempted reconstruction x. Where f is the activation function of the decoder, w is the weight matrix, b denotes the bias, and x denotes the reconstructed input sample.

$$\widehat{x} = f_2 (w_j h + b_j) \tag{3}$$

Reconstruction Loss: as shown in Eq. (4), a reconstruction loss (L) is computed in a typical self-encoder model to minimize the difference between the output and the input. Here, x denotes the input data, $\widehat{x}$ denotes the output/predicted data, and n is the number of samples in the training dataset [22].

$$\min_{w,b} L\,(x \quad \widehat{x})\,; where\ L\,(x \quad \widehat{x}) = \frac{1}{n}\sum_{n=1}^{n} j\widehat{x_t} \quad x_t j \tag{4}$$

Sparse Autoencoder A sparse autoencoder is a variant of the traditional autoencoder that includes a sparsity constraint on the hidden units, ensuring that only a small number of neurons are active at any given time. This constraint is typically enforced by adding a penalty term to the loss function. The sparsity constraint helps the autoencoder to learn more robust and useful features by focusing on the most significant aspects of the data. This characteristic is particularly advantageous for satellite network traffic, where capturing essential features and reducing dimensionality are crucial for efficient and accurate intrusion detection. The sparse autoencoder's structure consists of an encoder, which compresses the input data into a lower-dimensional representation, and a decoder, which reconstructs the original data from this representation. By doing so, the autoencoder can effectively capture the most critical features of the data, improving the performance and generalizability of the intrusion detection system.

Bi-LSTM Stack autoencoder In satellite network intrusion detection, we formulated a model that integrates Bi-LSTM and autoencoder components, as depicted in Fig. 4 The training effect of satellite network traffic using ordinary autoencoders is general. This method treats each traffic instance as an independent input, neglecting the temporal characteristics inherent in satellite network traffic and resulting in elevated training losses.

3.3. Proposed SAT-IDS

Drawing inspiration from machine learning and deep learning methodologies, we introduce two hybrid intrusion detection system (IDS) techniques. The initial approach integrates a Bi-LSTM sparse encoder with a class dictionary termed BLSAE-SNIDS-CD. The second technique combines a Bi-LSTM sparse encoder with a deep neural network (DNN), denoted as BLSAE-SNIDS-DNN.

BLSAE-SNIDS-CD We utilize a trained Bi-LSTM Decoder to encode the network traffic data. Initially, a Class_dict matrix of size $Num\_class \quad Num\_class$, where Num_class represents the number of classes, is initialized. This matrix records the activation frequency of each neuron in the encoder for every class label. Iterating over the training labels, we update the Class_dict matrix based on the following guidelines: for each image, we identify the index of the largest value in its feature mapping (representing the most active neuron in the encoder) and increment the value in the corresponding row and column of the Class_dict matrix by 1. This results in an output of the Class_dict matrix, which illustrates the activation frequency of each encoder neuron in distinguishing between different classes. Ideally, each neuron should activate exclusively for one class and not others.

To initialize a vector named Neuron_class, we map each neuron in the encoder to a class label. We iterate through the rows of the Class_dict matrix and assign each neuron to the class with the highest value in that row. Subsequently, we output the Neuron_class

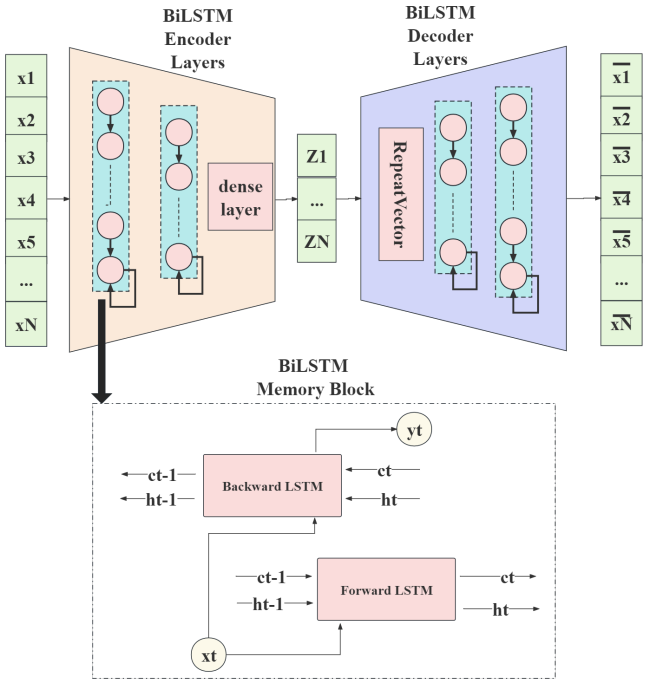

Fig. 4. Structure of Bi-LSTM autoencoder

vector, which indicates the class each neuron represents in the encoder. Then, we iterate over the predicted vector and replace each index with the corresponding class label from the neuron_class vector. Next, we compare the predicted vectors to the true labels of the test images. This comparison yields a Boolean vector indicating the correctness of each prediction.

The model can significantly influence satellite network traffic analysis, particularly in scenarios featuring a substantial volume of unlabeled data and a limited amount of labeled data. Notably, the model does not presuppose that the unlabeled data must align with the distribution of the labeled data.

BLSAE-SNIDS-DNN The structure of BLSAE-SNIDS-DNN, illustrated in Fig. 5, employs the same unlabeled data as BLSAE-SNIDS-CD for training the autoencoder. This enables the model to learn the intrinsic features of the input data and minimize the Mean Square Error (MSE) between the input and output. During training, BLSAE-SNIDS-DNN utilizes the encoder section for feature extraction on labeled data and then constructs a fully connected layer atop it as a classifier. Training the classifier with labeled data facilitates the prediction of the input data class based on the feature vector and minimizes the cross-entropy loss (CE). Subsequently, the encoder portion is employed for feature extraction on test data, and the classifier is utilized to predict its category. Leveraging the encoder's pre-trained weights from unsupervised training significantly enhances the model's accuracy.

3.4. Data Collection and Preprocessing

STIN data set This study employs the dataset provided by [9] to represent the satellite dataset for evaluating our model. The STIN security dataset encompasses various attack types observed in both satellite and terrestrial networks, comprising two satellite attack types and nine terrestrial attack types. The construction of the STIN dataset incorporates flow-based features. The distribution of the STIN satellite dataset is illustrated in Fig. 6.

This section outlines the following data preprocessing steps:

1) Feature selection: Feature selection is imperative to accommodate the limited resources of satellite nodes and minimize training and testing costs, optimizing resource utilization and reducing expenses. Initially, irrelevant features pertaining to the dataset's environmental and temporal aspects, such as quintuple, are discarded. Subsequently, features with missing values or that are entirely composed of zeros are eliminated. Further refinement is achieved through Pearson coefficient analysis and variance selection. Pearson's coefficient assesses the correlation between feature columns, leading to the removal of strongly correlated columns and subsequent reduction of the feature space. Variance selection filters each feature column's variance value, eliminating dimensions with low variance. Table 1 demonstrates the retention of the top 16 flow-level features crucial for intrusion detection, encompassing various benign and malicious traffic types simulated in the prototype.

2) Normalization: Normalization ensures uniformity in the range of values across each feature. Utilizing "MinMax" normalization, the range of feature values is scaled to fall between 0 and 1. The difference between the minimum value of the vector and the scale

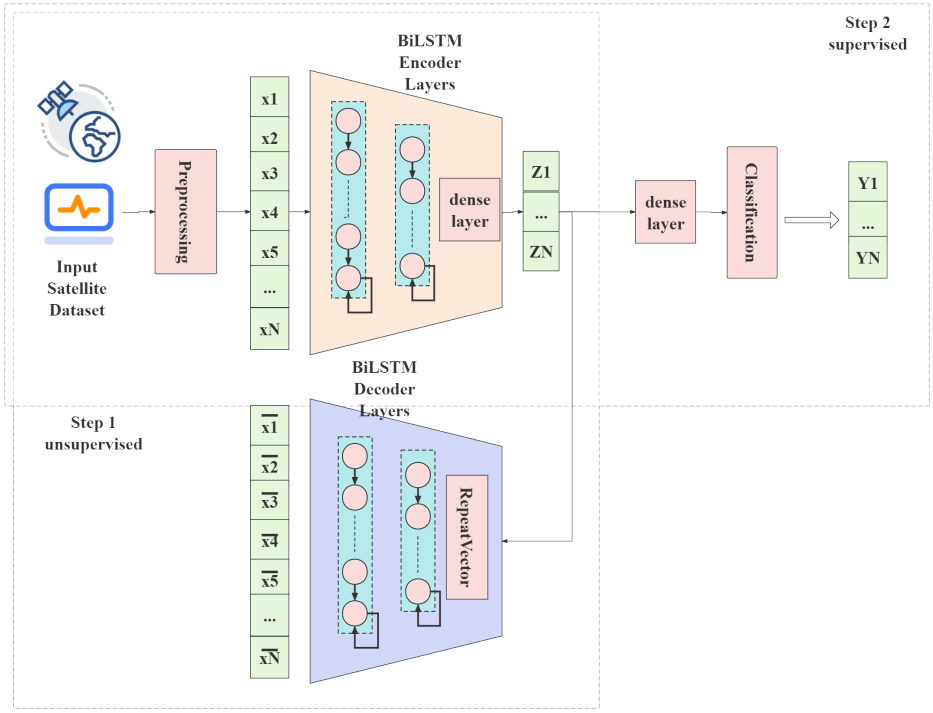

Fig. 5. Structure of supervised polar satellite intrusion detection system

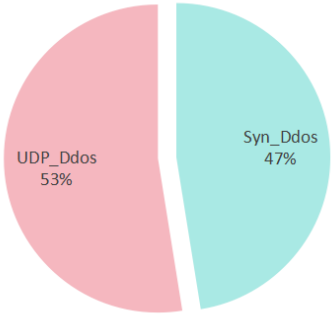

Fig. 6. Distribution of STIN satellite data sets

Table 1. Selected 16 features

Feature	Description
fl_dur	Flow duration
fw_pk	Total packets in the forward direction
l_fw_pkt	Total length of forward packets
l_bw_pkt	Total length of backward packets
pkt_len_min	Minimum length of a fow
pkt_len_max	Maximum length of a flow
pkt_len_std	Standard deviation length of a flow
fl_byt_s	Packet bytes transmitted per second
fl_iat_min	Minimum time between two backward packets,
bw_iat_tot	Total time between of two backward packets
bw_iat_min	Minimum time between of two backward packets
fw_hdr_len	Number of bytes used in forward packet header
bw_pkt_s	Number of backward packets per second
urg_cnt	Number of packets with URG
down_up_ratio	Ratio of forward and backward packet counts
bw_win_byt	Number of backward bytes in the initial window

size determines the new value. The normalization equation is as follows:

$$x_i = \frac{x_i \quad \min(x_i)}{\max(x_i) \quad \min(x_i)} \tag{5}$$

Here, $\overline{x_i}$ represents the normalized value, x_i denotes the i th feature vector, $\min(x_i)$ returns the minimum value of the vector, and $\max(x_i)$ returns the maximum value of the vector.

UNSW-NB15 dataset This section discusses the following data preprocessing steps.

The UNSW-NB15 dataset specializes in terrestrial traffic representation. This dataset is pivotal in safeguarding against unknown attacks, evaluating performance, and ensuring generalizability for machine learning-based intrusion detection systems. However, the NSL-KDD and KDDCup99 datasets have faced criticism due to their outdated attack types, imbalanced training and test set distributions, and lack of support for certain common network protocols, rendering them incompatible with contemporary cybersecurity requirements [23, 24].

To address the limitations of KDDCup/NSL-KDD, Moustaf and Slay [25] developed a more sophisticated intrusion detection dataset called UNSW-NB15, designed to reflect modern attacks and protocols. UNSW-NB15 is an example of an intrusion detection dataset crafted by researchers at the Australian Center for Cybersecurity (ACCS) using the IXIA tool, extracted from 100 GB of normal and modern datasets from attack traffic. The full UNSW-NB15 dataset comprises 2.5 million data records, encompassing one normal category and nine attack categories - Analytsis, Backdoor, DoS, Exploit, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms - with the raw data featuring 49 features.

The dataset creator also provided a partitioned dataset comprising 10% of the records, including a training set (175,341 records) and a test set (82,332 records), as depicted in

Fig. 7. The statistical distributions of the training and test set samples have been validated to be highly correlated, indicating the reliability of the machine learning model partitioning. Additionally, a few categories, such as Analytics, Backdoor, Shellcode, and Worms, comprise less than 2% of the dataset. In this 10% subset, some redundant features were eliminated, reducing the number of features to 42. In our study, we utilized this 10% dataset subset for categorization.

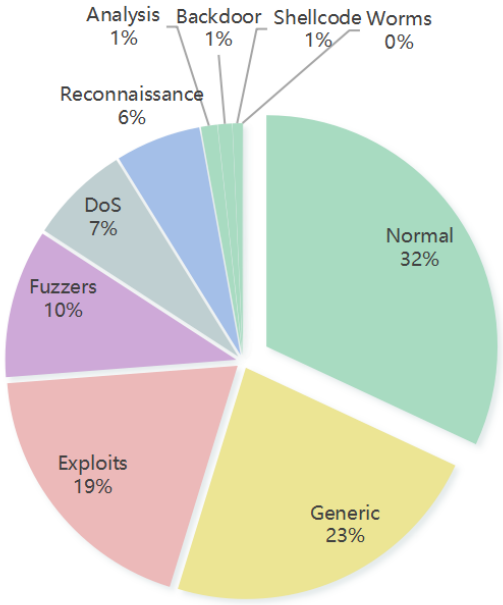

Fig. 7. Distribution of UNSW-NB15 data sets

This section outlines the following preprocessing steps:

1) Data cleaning: The provided UNSW-NB15 training and test sets initially contain 44 features. However, only 42 features are deemed meaningful, while the remaining two features serve as class labels for attacks. ”attack_cat” represents a multi-class label, while ”label” serves as a binary class label. Therefore, irrelevant labels are eliminated based on the model’s task. Additionally, 67,601 rows in the dataset containing null and duplicate values are cleaned.

2) One-hot encoding: The dataset includes three categorical features: ”service,” ”proto,” and ”state.” These features transform using one-hot encoding, a technique that converts each feature value into a binary feature.

3) Normalization: Normalization ensures uniformity in the range of values for each feature. ”MinMax” normalization is employed to scale the feature values between 0 and 1, as demonstrated in equation (5). The difference between the minimum value of the equation and the scale size determines the new value.

4) Feature selection: A comprehensive feature selection approach combining information gain and random forest importance is implemented to identify significant features. Ultimately, 42 optimal flow-level features are retained, which is crucial for intrusion detection and delineating diverse types of benign and malicious traffic simulated in the prototype.

4. Results and Discussion

The proposed BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN are tested on STIN and UNSW-NB15 datasets, with STIN as the satellite dataset and UNSW-NB15 as the ground dataset, to provide high security for the satellite-ground network. The experimental environment is shown in Table 2.

Table 2. Experimental environment

Equipment	
CPU	Intel(R) Xeon(R) Silver 4116 CPU@2.10GHz
GPU	Quadro P4000
RAM	64GB
OS	Ubuntu 16.04
Python	3.8.3
Database	STINUNSW-NB15

4.1. Evaluation Metrics

We utilize standard performance metrics to assess the efficacy of our proposed model on the STIN dataset, including accuracy, precision, recall, and F1 score. They are calculated using the following equations:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Re}call = \frac{TP}{TP + FN} \tag{8}$$

$$F1\ Score = 2 \quad \frac{Precision \quad \text{Re}call}{Precision + \text{Re}call} \tag{9}$$

The performance metrics are defined as follows: True Positive (TP) represents the count of accurately recognized abnormalities. True Negative (TN) represents the count of correctly recognized normal events. False Positive (FP) represents the count of normal events incorrectly diagnosed as abnormal. False Negative (FN) represents the count of abnormal events incorrectly recognized as normal.

4.2. Result Discussion

To illustrate the effectiveness of the proposed SAT-IDS, we conducted tests on two datasets: UNSW-NB15 and STIN. The confusion matrix of BLSAE-SNIDS-CD on the STIN test set is depicted in Fig. 8.

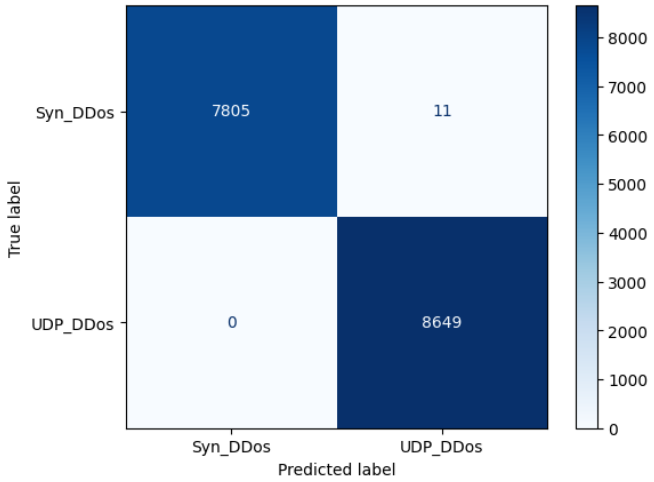

Fig. 8. Confusion matrix of BLSAE-SNIDS-CD on STIN dataset

The confusion matrix of BLSAE-SNIDS-DNN on the test set is displayed in Fig. 9. Notably, the model misclassifies only 1 test case per category, indicating a classification accuracy close to 100%.

The model undergoes training with 50 epochs and a batch size of 256. Subsequently, the learning curve depicting the loss of the training dataset is presented in Fig. 10.

We compare BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN with representative SAT-IDS models. Fig. 11 illustrates the accuracy comparison among all models. It is evident that both BLSAE-SNIDS-CD and BLSAE-SNIDS-DNN exhibit higher accuracy than most of the models, demonstrating strong performance across the board.

The performance comparison of all models using the STIN dataset in terms of accuracy, precision, recall and F1 score is shown in Table 3. The results show that BLSAE-SNIDS-DNN using bi-directional LSTM stacked autoencoder coded features achieved the best performance in all evaluation metrics with 99.99% in each metric.

Table 4 presents the performance of all models utilizing the STIN dataset. Notably, BLSAE-SNIDS-DNN demonstrates superior performance compared to other models, achieving a remarkable accuracy of 99.99% for both attack types. Its performance surpasses that of CNN, SVM, and RF-SFS-GRU by a significant margin.

The STIN security dataset contains different types of attacks from terrestrial and satellite networks. In the literature, researchers have proposed different techniques that are usually applied to terrestrial networks but not applicable to satellite networks due to different reasons such as limited resources, different tolerance to attacks, limited computational

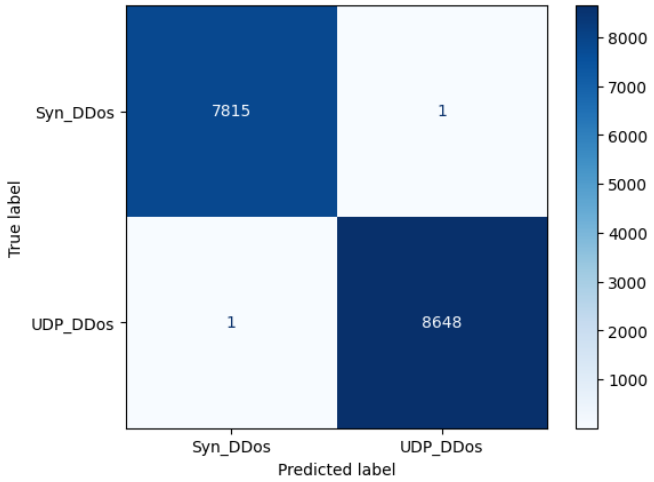

Fig. 9. Confusion matrix of BLSAE-SNIDS-DNN on STIN dataset

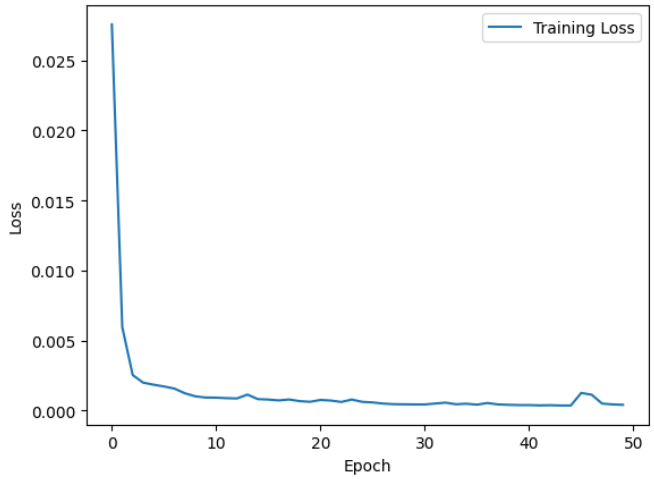

Fig. 10. Learning curves for loss of BLSAE-SNIDS-CD

power, and scarcity of satellite network datasets. The model performance comparison on the UNSW-NB15 dataset is shown in Table 5 and Fig. 12. The results show that the proposed BLSAE-SNIDS-DNN still performs better and outperforms the other models despite the fact that the performance of the other models decreases when used with the STIN dataset. The accuracy score of BLSAE-SNIDS-DNN on the UNSW-NB15 dataset is 90.83%. The model's best performance on these intrusion detection datasets shows the superiority, reliability, and ability to generalize well in the proposed model.

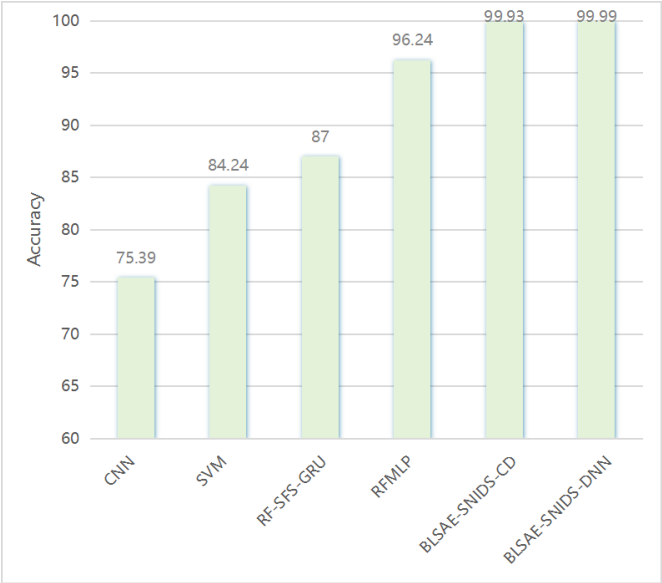

Fig. 11. Accuracy comparison diagram

Table 3. Model performance comparison on STIN satellite dataset

Ref.	Model	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)
	CNN	75.39	82.98	75.4	74.28
	SVM	84.24	83.35	85.62	84.48
[16]	RF-SFS-GRU	87	87	86	86
[13]	RFMLP	96.24	94.28	98.67	96.47
Proposed	BLSAE-SNIDS-CD	99.93	99.99	99.99	99.9
Proposed	BLSAE-SNIDS-DNN	99.99	100	100	100

Table 4. Accuracy of classifiers on STIN satellite dataset

Attack Type	CNN	SVM	RF-SFS-GRU	RFMLP	BLSAE-SNIDS-CD	BLSAE-SNIDS-DNN
Syn_DDos	66.3	86.18	93.36	100	99.85	99.99
UDP_DDos	98.4	83.37	94.49	93.18	100	99.99

Table 5. Model performance comparison on UNSW-NB15 dataset

model	Testing acc. (%)	Precision (%)	Recall (%)	F1-Score (%)
LSTM	63.61	69.16	61.61	65.17
ANN	63.61	69.16	61.61	65.17
RF-SFS-GRU	87	87	86	86
SFS-RF	90.5	91.19	90.41	90
BLSAE-SNIDS-DNN	90.83	91.21	90.85	90.4

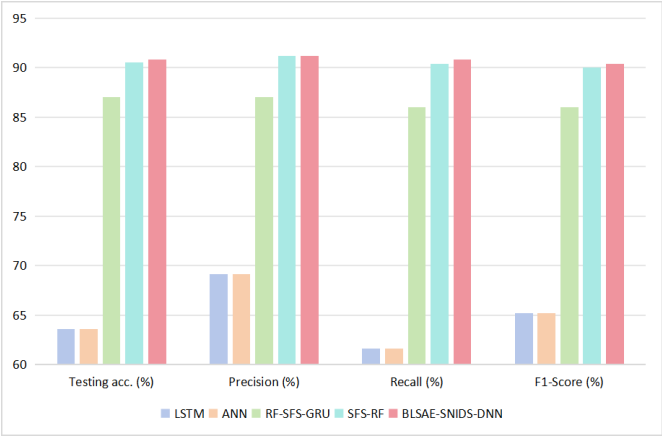

Fig. 12. Performance comparison of all models on UNSW-NB15 dataset

5. Conclusion

This study proposes a satellite network intrusion detection system (BLSAE-SNIDS) based on a Bi-LSTM sparse autoencoder, applied to both the STIN satellite dataset and the modern IDS dataset UNSW-NB15. BLSAE-SNIDS leverages unlabeled dat a to train the Bi-LSTM sparse autoencoder, enabling unsupervised representation learning that reduces dimensionality, parameters, and training time of classifiers. This enhances the model's generalization ability, making it suitable for satellite networks. Comparative analysis with state-of-the-art models demonstrates its superiority. Experimental results indicate that the proposed system achieves higher accuracy than other models on both datasets. However, this study is limited by dataset availability and is confined to the STIN dataset. Future research will aim to construct new datasets encompassing diverse satellite attack types and normal traffic to assess the efficacy of the proposed model.

Acknowledgments. This research is supported by the State Key Laboratory of Maritime Technology and Safety and the Natural Science Foundation of Shanghai under Grant 21ZR1426500.

References

1. Gaofeng Cui, Xiaoyao Li, Lexi Xu, and Weidong Wang. Latency and Energy Optimization for MEC Enhanced SAT-IoT Networks. *IEEE Access*, 8:55915–55926, 2020.
2. Charlotte Van Camp and Walter Peeters. A World without Satellite Data as a Result of a Global Cyber-Attack. *Space Policy*, 59:101458, February 2022.
3. Shangyuan Zhuang, Jiyan Sun, Hangsheng Zhang, Xiaohui Kuang, Ling Pang, Haitao Liu, and Yinlong Liu. Stinattack: A lightweight and effective adversarial attack simulation to ensemble idss for satellite-terrestrial integrated network. In *2022 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–8, 2022.
4. Dezhi Han, Nannan Pan, and Kuan-Ching Li. A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection. *IEEE Transactions on Dependable and Secure Computing*, 19(1):316–327, 2020.

5. Na Gao, Dezhi Han, Tien-Hsiung Weng, Benhui Xia, Dun Li, Arcangelo Castiglione, and Kuan-Ching Li. Modeling and analysis of port supply chain system based on fabric blockchain. *Computers & Industrial Engineering*, 172:108527, 2022.
6. Jiatao Li, Dezhi Han, Zhongdai Wu, Junxiang Wang, Kuan-Ching Li, and Arcangelo Castiglione. A novel system for medical equipment supply chain traceability based on alliance chain and attribute and role access control. *Future Generation Computer Systems*, 142:195–211, 2023.
7. Shaokang Cai, Dezhi Han, and Dun Li. A feedback semi-supervised learning with meta-gradient for intrusion detection. *IEEE Systems Journal*, 17(1):1158–1169, 2022.
8. Yan Wang, Dezhi Han, and Mingming Cui. Intrusion detection model of internet of things based on deep learning. *Computer Science and Information Systems*, (00):58–58, 2023.
9. Kun Li, Huachun Zhou, Zhe Tu, Weilin Wang, and Hongke Zhang. Distributed Network Intrusion Detection System in Satellite-Terrestrial Integrated Networks Using Federated Learning. *IEEE Access*, 8:214852–214865, 2020.
10. Dezhi Han, Yujie Zhu, Dun Li, Wei Liang, Alireza Souri, and Kuan-Ching Li. A blockchain-based auditable access control system for private data in service-centric iot environments. *IEEE Transactions on Industrial Informatics*, 18(5):3530–3540, 2021.
11. Jiatao Li, Dezhi Han, Dun Li, and Hongzhi Li. Blockchain and OR Based Data Sharing Solution for Internet of Things. In Jiachi Chen, Bin Wen, and Ting Chen, editors, *Blockchain and Trustworthy Systems*, pages 116–127, Singapore, 2024. Springer Nature.
12. Ao Di, Shi Ruisheng, Lina Lan, and Lu Yueming. On the Large-Scale Traffic DDoS Threat of Space Backbone Network. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*, pages 192–194, Washington, DC, USA, May 2019. IEEE.
13. Imran Ashraf, Manideep Narra, Muhammad Umer, Rizwan Majeed, Saima Sadiq, Fawad Javaid, and Nouman Rasool. A Deep Learning-Based Smart Framework for Cyber-Physical and Satellite System Security Threats Detection. *Electronics*, 11(4):667, February 2022.
14. Zhang Wen-bo, Sun Peigen, Liu Zhi-guo, and Xu Haifeng. An intrusion detection model for satellite network. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 167–170, Chengdu, China, 2010. IEEE.
15. Wei Guo, Jin Xu, Yukui Pei, Liuguo Yin, Chunxiao Jiang, and Ning Ge. A Distributed Collaborative Entrance Defense Framework Against DDoS Attacks on Satellite Internet. *IEEE Internet of Things Journal*, 9(17):15497–15510, September 2022.
16. Ahmad Taher Azar, Esraa Shehab, Ahmed M. Mattar, Ibrahim A. Hameed, and Shaimaa Ahmed Elsaid. Deep Learning Based Hybrid Intrusion Detection Systems to Protect Satellite Networks. *Journal of Network and Systems Management*, 31(4):82, October 2023.
17. Jianlong Zhu and ChunFeng Wang. Satellite Networking Intrusion Detection System Design Based on Deep Learning Method. In Qilian Liang, Jiasong Mu, Min Jia, Wei Wang, Xuhong Feng, and Baoju Zhang, editors, *Communications, Signal Processing, and Systems*, Lecture Notes in Electrical Engineering, pages 2295–2304, Singapore, 2019. Springer.
18. Bo Zhang, Hanwen Zhang, Gengming Zhao, and Jie Lian. Constructing a PM2.5 concentration prediction model by combining auto-encoder with Bi-LSTM neural networks. *Environmental Modelling & Software*, 124:104600, February 2020.
19. Hongzhi Li, Dezhi Han, and Mingdong Tang. A privacy-preserving storage scheme for logistics data with assistance of blockchain. *IEEE Internet of Things Journal*, 9(6):4704–4720, 2021.
20. Adel Binbusayyis and Thavavel Vaiyapuri. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class svm. *Applied Intelligence*, 51(10):7094–7108, 2021.
21. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Caan: Context-aware attention network for visual question answering. *Pattern Recognition*, 132:108980, 2022.

22. Chongqing Chen, Dezhi Han, and Chin-Chen Chang. Mpcct: Multimodal vision-language learning paradigm with context-based compact transformer. *Pattern Recognition*, 147:110084, 2024.
23. Yuhua Yin, Julian Jang-Jaccard, Wen Xu, Amardeep Singh, Jinting Zhu, Fariza Sabrina, and Jin Kwak. Igrf-rfe: a hybrid feature selection method for mlp-based network intrusion detection on unsw-nb15 dataset. *Journal of Big data*, 10(1):15, 2023.
24. Lijun Xiao, Dezhi Han, Ce Yang, Jiahong Cai, Wei Liang, and Kuan-Ching Li. Ts-dp: An efficient data processing algorithm for distribution digital twin grid for industry 5.0. *IEEE Transactions on Consumer Electronics*, 2023.
25. Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6, 2015.

Shuxin Shi is currently pursuing the PhD degree with Shanghai Maritime University. His research interests include network security and machine learning.

Bing Han received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, Dalian University of Technology, Dalian, China, in 2003 and 2009, respectively. He is currently a Researcher with the State Key Laboratory of Navigation and Safety Technology, Shanghai Ship and Shipping Research Institute, Shanghai, China. He has authored or coauthored over 40 peer-reviewed journal and conference papers. He was also sponsored by the Shanghai Talent Development Fund and Shanghai Rising-Star Program in 2014 and 2015, respectively. His current research interests include intelligent ship and intelligent systems and their applications, the design of deep sea dynamic positioning control systems.

Zhongdai Wu received a bachelor's degree in management information systems from Dalian Maritime University, China, M.B.A. degree from Tongji University, China. and Ph.D. degree in business management from Tongji University. Currently, he is working at COSCO SHIPPING TECHNOLOGY CO.,LTD., as Vice General Manager & CIO and also at Shanghai Institute of Navigation as vice president. His main research specialized in cloud computing, information system security, big data, Marine satellite communications and navigation, machine learning and blockchain.

Dezhi Han received the bachelor's degree from Hefei University of Technology, Hefei, China, in 1990, and the master's and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2001 and 2005, respectively. He is currently a Professor of Computer Science and Engineering with Shanghai Maritime University, Shanghai, China. His research interests include cloud computing, mobile networking, wireless communication, and cloud security.

Huafeng Wu received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2009, the master's degree in traffic information engineering and control from Dalian Maritime University, Dalian, China, in 2004. He was a Postdoctoral Research Fellow with Carnegie Mellon University, Pittsburgh, PA, USA, from 2008 to 2009, and also a Visiting Scholar with Shanghai Jiao Tong University, Shanghai, from 2012 to

2013. He is currently a Professor and a Ph.D. Supervisor with the Merchant Marine College, Shanghai Maritime University, Shanghai. His research interests include the Internet of Shipping, Internet of Things, and wireless sensor networks. Prof. Wu also serves as an Editorial Board Member for the Computer Communications.

Xiaojun Mei received the B.E. degree in navigation technology (excellent engineer education) and the Ph.D. degree in traffic information engineering and control from Shanghai Maritime University, Shanghai, China, in 2016 and 2021, respectively. He was a Postdoctoral Fellow with Shanghai Maritime University from November 2021 to October 2023. From September 2019 to September 2020, he was a visiting Ph.D. student with the Institute for Systems and Robotics, Instituto Superior Te´cnico, University of Lisbon, Lisbon, Portugal. He is currently an Associate Professor with the Merchant Marine College, Shanghai Maritime University and also with the State Key Laboratory of Maritime Technology and Safety, Shanghai Ship and Shipping Research Institute, Shanghai. His current interests include localization in marine/ocean/underwater wireless sensor networks and path planning for AUVs/ASVs.

Received: April 01, 2024; Accepted: July 22, 2024.