J Inf Process Syst, Vol.20, No.3, pp.375~390, June 2024
https://doi.org/10.3745/JIPS.01.0105

ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Network Anomaly Traffic Detection Using WGAN-CNN-BiLSTM in Big Data Cloud–Edge Collaborative Computing Environment

Yue Wang*

Abstract

Edge computing architecture has effectively alleviated the computing pressure on cloud platforms, reduced network bandwidth consumption, and improved the quality of service for user experience; however, it has also introduced new security issues. Existing anomaly detection methods in big data scenarios with cloud–edge computing collaboration face several challenges, such as sample imbalance, difficulty in dealing with complex network traffic attacks, and difficulty in effectively training large-scale data or overly complex deep-learning network models. A lightweight deep-learning model was proposed to address these challenges. First, normalization on the user side was used to preprocess the traffic data. On the edge side, a trained Wasserstein generative adversarial network (WGAN) was used to supplement the data samples, which effectively alleviates the imbalance issue of a few types of samples while occupying a small amount of edge-computing resources. Finally, a trained lightweight deep learning network model is deployed on the edge side, and the preprocessed and expanded local data are used to fine-tune the trained model. This ensures that the data of each edge node are more consistent with the local characteristics, effectively improving the system's detection ability. In the designed lightweight deep learning network model, two sets of convolutional pooling layers of convolutional neural networks (CNN) were used to extract spatial features. The bidirectional long short-term memory network (BiLSTM) was used to collect time sequence features, and the weight of traffic features was adjusted through the attention mechanism, improving the model's ability to identify abnormal traffic features. The proposed model was experimentally demonstrated using the NSL-KDD, UNSW-NB15, and CIC-ISD2018 datasets. The accuracies of the proposed model on the three datasets were as high as 0.974, 0.925, and 0.953, respectively, showing superior accuracy to other comparative models. The proposed lightweight deep learning network model has good application prospects for anomaly traffic detection in cloud–edge collaborative computing architectures.

Keywords

Abnormal Traffic Mining, Big Data, BiLSTM, Cloud–Edge Collaborative Computing, CNN, Wasserstein Generative Adversarial Networks

1. Introduction

The popularization of the Internet of Vehicles (IoV), mobile edge computing (MEC), industrial Internet, augment/virtual reality, and other technologies has led to an increase in intelligent terminal data. Centralized cloud computing cannot meet the requirements of the exponential growth in the num-

Manuscript received March 13, 2023; first revision May 30, 2023; accepted June 25, 2023.

* **Corresponding Author:** Yue Wang (wangyue@nyist.edu.cn)
School of computer and software, Nanyang Institute of Technology, Nanyang, Henan, China (wangyue@nyist.edu.cn)

ber of devices and data on the ultimate user experience. MEC can effectively compensate for the shortcomings of cloud computing by sinking the cloud-computing capability and IT service environment to edge nodes and providing storage or computing services to users nearby [1].

MEC has rich application scenarios, such as private network applications in enterprise parks and IoV, which present convenience and security challenges [2]. This is a threat to users. The MEC nodes on the edge gather sensitive information from surrounding users, and malicious users can use the edge nodes for horizontal or vertical attacks. Malicious users can be divided into two categories: attackers with illegal identities and those with legitimate identities (the most destructive attackers). Attackers with legitimate identities typically have internal permissions to understand the location of sensitive data. Attackers impersonating the identities of legitimate users and the intentional or unintentional malicious behaviors of legitimate users pose significant challenges to MEC security protection. However, new challenges have arisen owing to resource limitations and other characteristics of MECs [3].

Uploading behavioral data from many terminal equipment to cloud computing centers for pre-processing and model training increases the training burden on the cloud, and mass data also incur additional time consumption during the upload process. Simultaneously, this may also lead to the leakage of user privacy. However, MEC nodes exhibit resource-constrained and distributed characteristics, which make it difficult for them to carry protection configurations in cloud computing [4]. Therefore, to enhance the quality of service (QoS) for user experience while considering the limited resources of MEC scenarios, lightweight monitoring of abnormal traffic within MEC scenarios is crucial for protecting MEC security.

Existing anomaly detection methods in MEC scenarios experience hurdles, such as severe sample imbalance, difficulty in dealing with complex network traffic attacks, and difficulty in effectively training large-scale data or overly complex deep-learning network models.

A lightweight deep learning network model, WGAN-CNN-BiLSTM, which uses Wasserstein generative adversarial networks (WGAN) [5], convolutional neural networks (CNN) [6], and bidirectional long short-term memory networks (BiLSTMs) [7] was proposed, implementing illegal intrusion traffic detection through collaborative computing. The training samples with voluminous data are stored on a cloud server with strong computing and storage capacity, and tasks with a small amount of calculation are unloaded to the users' ends and edge ends. This reduced the transmission delay of the calculation results and improved the QoS for the user. Among these, data preprocessing operations were performed at the users' end. At the edge ends, local data are expanded, and each edge node fine-tunes the already-trained lightweight model based on local data to ensure that it is more in line with the local characteristics, thereby effectively improving the detection ability of each edge node. Simultaneously, it avoids using abnormal traffic datasets in a single scenario and instead chooses traffic datasets from multiple scenarios for training to cope with complex and diverse attack types of traffic, thereby ensuring the feasibility of the designed solution.

Section 2 introduces traditional abnormal traffic detection models based on cloud computing and abnormal traffic detection models based on edge or fog computing. Section 3 introduces cloud–edge architectures and the designed lightweight deep learning network model, WGAN-CNN-BiLSTM. Section 4 verifies the reliability of the proposed model through experiments on three datasets. Section 5 discusses the experimental results, summarizes the limitations of the designed model, and presents prospects for the next steps of the study.

2. Related Works

Along with the popularization of artificial intelligence and 5G communication, deep neural network (DNN) technology also shows certain advantages in big data processing and network intrusion detection and has been widely used [8-10]. For example, Kumar et al. [11] proposed an anomaly network detection method based on an improved particle swarm optimization algorithm that could effectively improve the recognition ability of data mining. Jiang et al. [12] proposed an improved support vector machine (SVM) model for an abnormal network traffic identification strategy and framework that can effectively identify abnormal network traffic and intrusion detection by coordinating packets with similar traffic. However, this model is relatively simple, and it is difficult to deal with complex networks.

The authors of [13] proposed a hierarchical deep learning scheme (STL-HDL) based on big data to detect abnormal network data traffic. It uses behavior and content characteristics to represent network traffic characteristics. Combined with the distribution characteristics of the data in the cluster, the reliability of intrusion detection was improved.

According to the data characteristics of the power Internet of Things (IoT), Tang et al. [14] effectively realized the detection of abnormal traffic in the power IoT by extracting network traffic characteristics, using machine learning to analyze and identify abnormal traffic, establishing attack models, and comparing them with equipment images. Among them, the method is traditional, and its processing performance is difficult to match with current Internet developments.

Fei et al. [15] proposed an attack detection system (CNN-LSTM). Combining a CNN and an LSTM can ensure detection accuracy and effectively mitigate the impact of data imbalance. However, the impact of data redundancy and other factors was not considered, and the detection accuracy needs to be improved.

Lee et al. [16] provided an anomaly-detection scheme (SVM-DAE) utilizing a SVM and a deep autoencoder (DAE) that can be directly deployed on resource-constrained devices. This scheme uses publicly available traffic data (including labeled malicious traffic) as the dataset for anomaly detection and binary classification, which solves the problem of feature redundancy to a certain extent; however, it requires samples to contain labels and have balanced characteristics.

The scheme designed in [17] can be directly hosted and executed on an edge device. This scheme uses an isolation forest (iForest) to detect normal and abnormal traffic. The training dataset comprises traffic data, and the required training samples do not need to be labeled; therefore, it can deal with unknown threats.

The scheme proposed in [18] is managed by a cloud server for storing data, trained by fog nodes, and MEC servers for model training and anomaly detection using the sample-selected extreme learning machine (SS-ELM) algorithm. This reduces the computational burden on the cloud datacenter, reduces training time, and improves training accuracy. However, the KDD Cup'99 dataset used in the scheme cannot be used for evaluating network intrusion detection systems.

Maimo et al. [19] designed a scalable anomaly-detection framework (DNN-LSTM) for 5G network user traffic. This framework uses deep learning technology to extract 144 traffic features from network traffic and detect user traffic in two stages. The first stage uses a DNN for anomaly detection, and the second stage uses LSTM for anomaly detection and realizes the resource consumption optimization of the detection system. However, the QoS for user experience has not been considered.

To effectively improve the QoS of the user experience, [20] offloaded the computing tasks of the core network to the MEC server on the edge side based on the MEC method. Driven by real user call records, the feedforward DNN algorithm was used for anomaly detection, overcoming the limitations of high false positives and low accuracy and improving the quality of user service experience. Based on this, Hussain et al. [21] achieved the same goal by combining cell region division with a CNN (CRD-DNN). However, these methods have difficulty addressing the issue of abnormal sample feature positions.

To address the issue of the abnormal location of sample features, De Souza et al. [22] proposed a combined framework based on random forest and deep neural network (RF-DNN) for fog computing environments, and the authors of [23] proposed a combined framework integrating deep forest and biological heuristic algorithm (DR-BIA), which effectively improved the robustness of intrusion detection. However, these methods do not consider the issue of an imbalance between positive and negative samples.

In summary, existing network intrusion detection methods typically have the following issues: (1) many methods complete all training tasks on cloud computing platforms, greatly increasing the latency during data transmission, resulting in a lower quality of user service experience; (2) many methods based on edge computing fail to consider the imbalance of a small number of samples, which reduces the model's training effect; and (3) in the face of complex network traffic attacks, many methods based on edge computing are difficult to deal with effectively.

To solve these three problems, different coping strategies have been adopted, and a lightweight deep learning combination architecture integrating CNN-BiLSTM and an attention mechanism has been proposed. The three main points of the innovation are as follows.

(1) Compared to traditional cloud-computing models, adopting a cloud–edge collaborative computing architecture can effectively reduce the computational pressure on cloud-computing centers. Large computing and training tasks that do not require real-time responses are completed in the cloud computing center, and preprocessing operations are completed on the user side without computing resources on the edge and cloud sides. Simultaneously, lightweight network models that have already been trained in the cloud are deployed on the edge side, and only local data are needed to fine-tune the model, which can adaptively improve the local detection performance of each edge node and reduce the transmission delay of the calculation results, thus greatly improving the quality of user experience.

(2) On the edge side, the trained WGAN model is used to effectively expand the data samples, which effectively solves the problem of a few types of imbalances while occupying a small amount of edge-computing resources.

(3) For complex network traffic attack data, the fusion of CNN-BiLSTM and the attention mechanism can simultaneously consider spatial, temporal, and important features, thereby effectively improving the overall accuracy of abnormal traffic detection.

3. Network Anomaly Traffic Model based on Deep Learning in Cloud–Edge Collaborative Environment

3.1 Cloud–Edge System Architecture Model

Fig. 1 shows the designed cloud–edge architecture. The services for many mobile users in the coverage area are provided by the edge node, which can be connected to the cloud service center. Each edge node can make predictions by computing to proactively cache the content that users may request in advance, reduce the user waiting delay, and thus enhance the quality of experience (QoE) [24]. Users can obtain the demand content locally through an edge node or obtain the requested content from the cloud center.

Normalization on the user side is used to preprocess user behavior data, and each edge node receives requests from users in the service area at a time step through the user interface module and stores the request information in the local database [25]. The edge node sends the received request information to the request processing module. The trained WGAN is used to expand the data of a small number of samples, uses the trained lightweight deep learning model to predict the user's future requests (request prediction module), and cooperates with neighboring edge nodes to make cache decisions (cache decision module) using the prediction information. Finally, the cache management module determines the local active cache content. Because users can constantly enter or leave the service scope of all edge nodes and new content is generated constantly, users' demand for content is constantly changing; therefore, the request prediction model is time-sensitive [26]. Additionally, it is necessary to update the data using new training data. Therefore, the current local node can combine multiple neighboring edge nodes to obtain a new model via deep learning training [27].

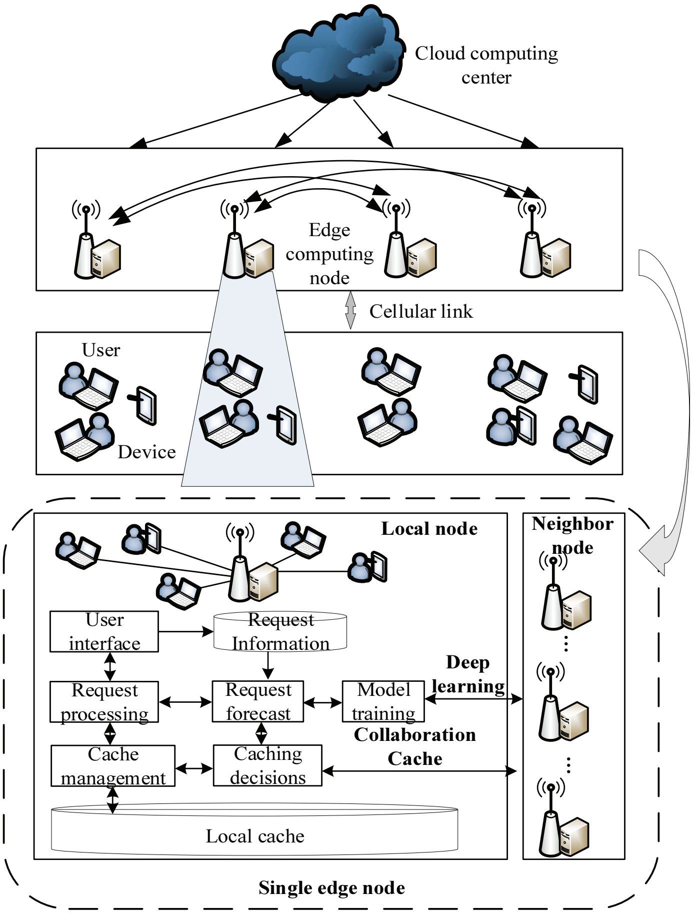


Fig. 1. Cloud–edge collaborative system architecture.

3.2 CNN-BiLSTM Model

A CNN-BiLSTM model was proposed to extract high-dimensional traffic characteristics, and its structure is shown in Fig. 2 [28].

In the CNN, two groups of convolution-pooling layers were used to extract the spatial features. The features were then input into the BiLSTM network to collect the time features. A total of 128 and 64 neurons were set in the two layers of BiLSTM. The final output was the high-dimensional spatiotemporal features extracted by the CNN-BiLSTM neural network and processed by the attention mechanism. The AdamW optimizer was selected in CNN-BiLSTM, selecting the ReLU and sigmoid functions as the activation functions of the CNN and BiLSTM layers, respectively, and the Kaming and Xavier initialization methods were used to set up the initialization parameters of CNN and BiLSTM, respectively.

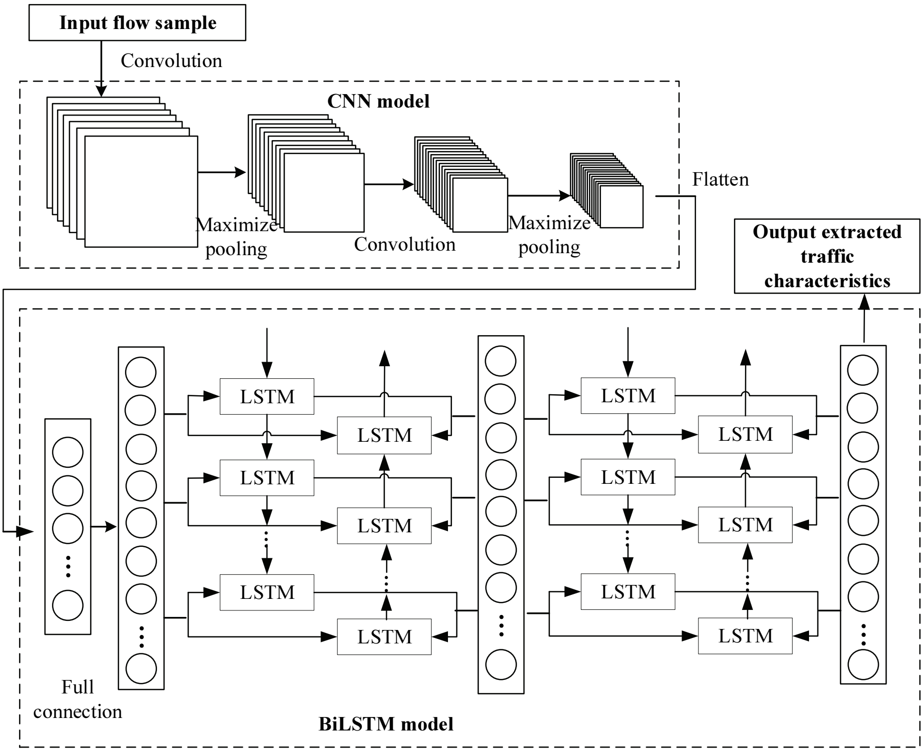


Fig. 2. Structure of CNN-BiLSTM model.

3.3 CNN Model

The 1D, 2D, and 3D neural networks are included in CNN [29], which can be well used for sequence data processing, text classification, and image processing. Among these, sequence data processing is the main application of 1D CNN; 2D CNN is used for text analysis; and 3D CNN is mainly used for processing images and video data. Fig. 3 shows the overall structure of the CNN model.

The input sequence data were classified according to the categories in the CNN, and the sum of the output probabilities was 1. The elements of each convolution kernel in the convolution layer have corresponding weights and deviations. Each neuron in the convolution layer is connected to a neuron in the previous layer [30]. During operation, the input sequence data features are scanned regularly and summed by multiplying the matrix elements in the area where the previous layer is close to each other, and the deviation amount is stacked [31].

The pooling layer in the CNN is used to select the processed features and filter the information, and the preset pooling function is used to replace a single point with the feature statistics of the adjacent area. The fully connected layer is located at the last position in the CNN, and the signal is transmitted only to the other fully connected layers. The role of the fully connected layer is to combine the pooled features nonlinearly and transmit the results to the output layer.

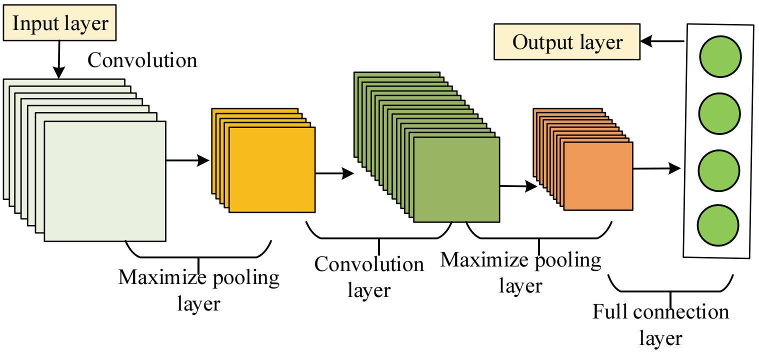


Fig. 3. CNN model.

3.4 BiLSTM Model

The LSTM neural network [32] comprises an input gate (IG), an output gate (OG), and a forgetting gate (FG). The FG is used to determine the amount of information retained in the previous cell state under the current cell state, and IG is used to determine the information to be updated under the current cell state. The OG is used to obtain the output of the current state. A detailed diagram of the network structure is presented in [32]. The LSTM update process is as follows:

$$\begin{aligned}
\hat{c}_t &= \tanh\,(\omega_{xx}x_t + \omega_{hc}h_{t-1} + b_c) \\
i_t &= \delta\big(\omega_{xi}x_t + \omega_{hih}h_{t-1} + \omega_{ct_{t-1}}c_b\big) \\
f_t &= \delta\big(\omega_{xf}x_t + \omega_{hf}h_{t-1} + \omega_{ct}c_{t-1} + b_f\big) \\
o_t &= \delta(\omega_{x0}x_t + \omega_{ho}h_{t-1} + \omega_{cc}c_{t-1} + b_o) \\
h_t &= o_t * \tanh\,(c_t) \\
y_t &= \omega_{yh}h_t + b_y,
\end{aligned} \tag{1}$$

where δ is the sigmoid function; tanh is a function; $\hat{c}_t$ and c_t are temporary and current memory cell values, respectively; i_t and f_t are the IG and FG values, respectively; o_t and h_t are the output results of the current output gate and hidden layer, respectively; y_t is the output result of the network; and ω and b are the weights and offsets, respectively. The LSTM network controls the trend of the historical information using three control gates; thus, it uses long-term historical data.

The BiLSTM network is shown in Fig. 4. The BiLSTM can be obtained by calculating the forward and backward directions of the hidden layer of the LSTM, and more data features can be obtained through this BiLSTM neural network.

The update process for BiLSTM is as follows:

$$\begin{aligned} h^{+} &= \varphi^{+}(h_{t-1}, x_t) \\ h^{-} &= \varphi^{-}(h_{t+1}, x_t) \\ \hat{y}_t &= \omega_{yh} h^{+} + \widehat{\omega}_{yh} h^{-} + b_y, \end{aligned} \tag{2}$$

where φ^{+} and φ^{-} are LSTM neural network operations and ω_{yh} and $\widehat{\omega}_{yh}$ are the forward and backward direction-calculated weights, respectively.

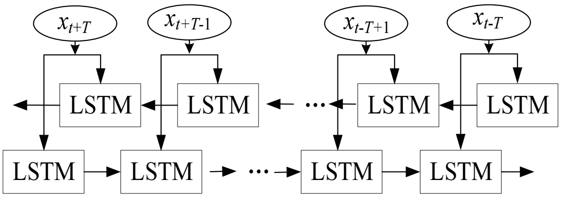


Fig. 4. Structure of BiLSTM model.

3.5 Attention Model

To address the feature redundancy, an attention mechanism was introduced to design a resource allocation mode that allocates limited computing resources to more important tasks to improve resource utilization [33]. Specifically, information is encoded more effectively by assigning personalized weights to features at different times. The information provided by the proposed CNN-BiLSTM model at different times has different effects on the accuracy of the anomaly mining results. However, traditional neural networks cannot identify the importance of the signal value sequences. Therefore, this study improves the CNN-BiLSTM network model and introduces an attention mechanism to automatically identify the importance of historical data at different times. Fig. 5 shows the attention mechanism.

In this model, the output is:

$$H_t^a = \sum_{k=1}^{T+1} \gamma_k h_{t-(T-k)}^s, \tag{3}$$

$$\gamma_k = \frac{\exp\,(s_k)}{\sum_{k=1}^{T+1} \exp\,(s_k)}, \tag{4}$$

$$s_k = \phi_s^T \tanh(\omega_{hs} g_t^s + \omega_{ls} h_t^s), \tag{5}$$

where $T+1$ represents the length of the time series, γ_k is the attention weight, s_k is the importance of each part of the signal sequence, g_t^s and h_t^s represents the output of the CNN and BiLSTM hidden layers.

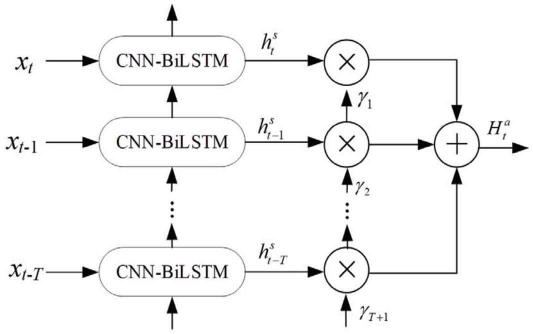


Fig. 5. Structure of attention mechanism model.

4. Experiment and Analysis

Table 1 lists the configuration of the experimental environment. The deep learning frameworks TensorFlow and Keras are adopted, and GPU is used for acceleration. Pandas is mainly used for data reading and preprocessing, and sk-learning is used to calculate evaluation indicators and perform machine learning comparison experiments.

Table 1. Experimental environment

Software and hardware	Type
CPU	Intel Core i7-7700HQ
Memory	32 GB
GPU	NVIDIA Tesla T4 16 GB
Running system	Windows 10
Programming language	Python 3.8
Data processing	Pandas 1.1.2
Machine learning	Sklearn 0.23.2

4.1 Datasets

The NSL-KDD [34], UNSW-NB15 [34], and CIC-IDS2018 datasets [35] were used to demonstrate the proposed model.

(1) NSL-KDD dataset: the experimental network traffic dataset used the KDDTrain+ and KDDTest+ files. The NSL-KDD dataset does not contain redundant or duplicate records; therefore, it is widely used as a benchmark dataset in many intrusion-detection systems. The training set, KDDTrain+, included tag samples of 22 attack types, and the test set, KDDTest+, included tag samples of 39 attack types. Therefore, the NSL-KDD dataset can be used to estimate the generalization ability, making the detection ability of the model more accurate. Each traffic sample in the dataset contains 41 characteristics, including 38 numerical values (such as "int 64" or "float 64") and 3 symbolic values (such as "object"). Additionally, KDDTrain+ and KDDTest+ contain multiple-class tags. However, this model was used to detect exceptions and perform only binary classification. Therefore, this study replaced the dataset labels. The normal traffic label is zero, and the abnormal traffic label is 1.

(2) UNSW-NB15 dataset: an open dataset for network intrusion detection. UNSW-NB15 includes nine attack types and one normal attack type. It comprises three nominal characteristics, two binary characteristics, and 37 numerical characteristics. It is recorded in chronological order and fully represents the temporal correlation between the data. In the experiment, 70% of the dataset was used for training and 30% for testing.

(3) CIC-IDS2018 dataset: many tools are available to convert raw PCAP packets into network stream data. One of the tools that can be used to convert PCAP data into network traffic data is CICFlowMeter, created by the New Brunswick University. CICFlowMeter is a network traffic generator written in Java. It accepts the original PCAP as the input and outputs the two-way network traffic. Traffic direction can be determined from the first data packet. In the field of intrusion detection, because many datasets are internal (for example, the government or financial institutions cannot disclose their data for privacy protection), many datasets cannot reflect the current network situation or lack statistical characteristics. Therefore, obtaining a dataset with full attack coverage, high data richness, real and reliable data, and timeliness is difficult. Simultaneously, the quality of the dataset affected the effectiveness of the intrusion detection model. After comprehensively considering the factors of data size, type, timeliness, and authenticity, CIC-IDS2018 was selected as the evaluation dataset. The captured raw PCAP data were used to generate 80 stream characteristics using the CICFlowMeter.

4.2 Evaluating Indicator

Accuracy, precision, recall, and F1 values were used to evaluate the performance of the mining model. The calculations are as follows:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ Precision &= \frac{TP}{TP+FP} \\ F1 &= 2\times\frac{Precision \times Recall}{Precision + Recall}, \end{aligned} \tag{6}$$

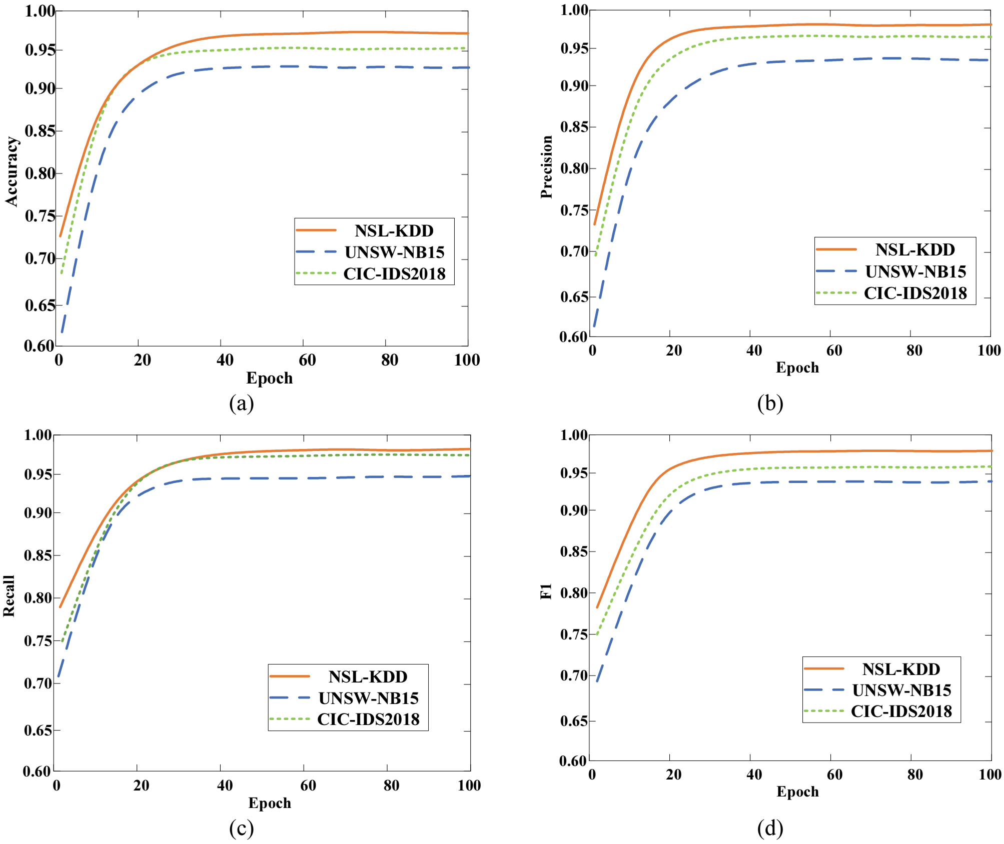
where TP is the number of positive samples classified as positive, TN is the number of negative samples classified as negative, FP is the number of positive samples classified as negative, and FN is the number of negative samples classified as positive.

4.3 Model Training

4.3.1 Result analysis of the evaluating indicator

Fig. 6 shows the changes in the evaluation indicator values of the proposed model with epochs for the three datasets.

As shown in Fig. 6, the proposed WGAN-CNN-BiLSTM model achieved a relatively ideal abnormal traffic mining effect, and each evaluation indicator performed well. The proposed model combines the CNN-BiLSTM network and attention mechanism to better focus on the characteristics of abnormal traffic, especially for datasets with rich data characteristics. Its prediction accuracy is higher, and the precision of the NSL-KDD dataset exceeds 0.974.



(a)

(b)

(c)

(d)

Fig. 6. Changes in model evaluation metrics for different datasets: (a) accuracy, (b) precision, (c) recall, and (d) F1 value.

4.3.2 Model training time

The training times of the proposed WGAN-CNN-BiLSTM for the three experimental datasets are shown in Fig. 7.

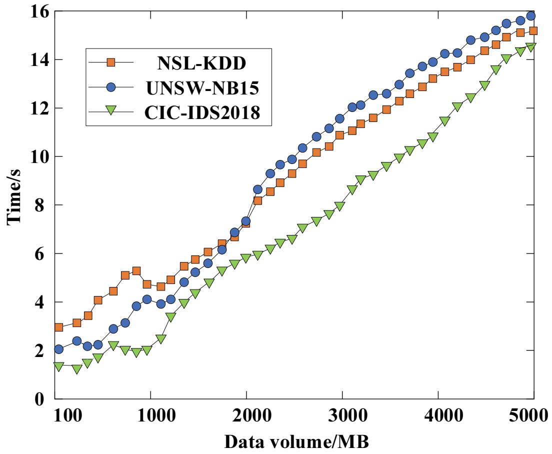


Fig. 7. Model training time for different datasets.

From Fig. 7, the increase in data volume leads to continuous growth in the training time of WGAN-CNN-BiLSTM, but the growth rate is obvious in the later stage. This is mainly because the amount of data is too large and the edge nodes are limited, resulting in network congestion. Therefore, when the amount of data was 5,000 MB, the time was close to 16 seconds.

4.4 Model Performance Verification

The reliability of the WGAN-CNN-BiLSTM CNN–BiLSTM was analyzed by comparing it with several other deep-learning network models, including SVM-DAE [16], DNN-LSTM [19], CRD-DNN [21], RF-DNN [22], and DR-BIA [23]. The test results of the four models on the NSL-KDD, UNSW-NB15, and CIC-ISD2018 datasets are presented in Tables 2–4.

Table 2. Evaluating indicator of different models on NSL-KDD dataset

	SVM-DAE	DNN-LSTM	CRD-DNN	RF-DNN	DR-BIA	WGAN-CNN-BiLSTM
Accuracy	0.943	0.957	0.953	0.961	0.965	0.974
Precision	0.938	0.954	0.949	0.958	0.964	0.979
Recall	0.946	0.956	0.946	0.972	0.980	0.975
F1	0.949	0.960	0.950	0.963	0.962	0.972

Table 3. Evaluating indicator of different models on UNSW-NB15 dataset

	SVM-DAE	DNN-LSTM	CRD-DNN	RF-DNN	DR-BIA	WGAN-CNN-BiLSTM
Accuracy	0.854	0.903	0.898	0.919	0.914	0.925
Precision	0.861	0.916	0.907	0.918	0.920	0.930
Recall	0.878	0.922	0.915	0.906	0.919	0.949
F1	0.870	0.919	0.911	0.915	0.921	0.939

Table 4. Evaluating indicator of different models on CIC-IDS2018 dataset

	SVM-DAE	DNN-LSTM	CRD-DNN	RF-DNN	DR-BIA	WGAN-CNN-BiLSTM
Accuracy	0.887	0.929	0.921	0.948	0.941	0.953
Precision	0.893	0.935	0.928	0.960	0.948	0.958
Recall	0.905	0.941	0.936	0.963	0.963	0.970
F1	0.899	0.938	0.932	0.959	0.955	0.956

Tables 2–4 show that, owing to the comprehensive network features, reliable data of the NSL-KDD dataset, and all comparison models being deep learning network models, all models can achieve high detection results. However, the UNSW-NB15 dataset contains fewer data features and more types of attacks, making mining more difficult. Most models achieved lower results than for the other two datasets. Compared with the SVM-DAE and CRD-DNN models, DNN-LSTM considers both spatial and temporal features in traffic data, thus achieving better results. However, LSTM can only capture unidirectional relationships between temporal features, whereas BiLSTM in the proposed WGAN-CNN-BiLSTM model can capture bidirectional relationships between temporal features, which is more advantageous than LSTM. The RF-DNN and DR-BIA models can effectively solve the problem of

abnormal sample feature positions, thus achieving higher performance than the SVM-DAE, CRD-DNN, and DNN-LSTM models. However, class imbalance issues were not considered in the RF-DNN and DR-BIA models. The proposed WGAN-CNN-BiLSTM model, by introducing WGAN for data expansion, can effectively alleviate class imbalance issues and balance the advantages of spatial and temporal features in SVM-DAE, CRD-DNN, and DNN-LSTM models. In addition, by introducing attention mechanisms to strengthen important features and filter out unimportant features, the effectiveness of the WGAN-CNN-BiLSTM can be further optimized. Therefore, the proposed model achieves the best results in most cases.

5. Conclusion

Existing anomaly detection methods in cloud–edge collaborative computing scenarios face many challenges, such as severely imbalanced samples, difficulty in dealing with complex network traffic attacks, and difficulty in effectively training large-scale data or overly complex deep-learning network models. A lightweight deep learning model was proposed to address these challenges. The experimental results obtained from the NSL-KDD, UNSW-NB15, and CIC-ISD2018 datasets are as follows:

(1) By combining cloud computing with edge computing, large-scale computing tasks are completed in the cloud, small- and medium-sized computing tasks are completed on the edge, and very small computing tasks are completed on the user side, effectively reducing the user waiting delay. Once the data volume reached 5000MB, the training time of the proposed WGAN-CNN-BiLSTM model was close to 16 s, improving the quality of the user experience.

(2) The WGAN is used for data augmentation, and the lightweight network model designed by integrating the CNN, BiLSTM, and Attention mechanisms effectively improves the detection accuracy. For example, the accuracy of the NSL-KDD dataset exceeded 0.974.

(3) Compared with other advanced models, it was found that, temporal characteristics and data class imbalance as well as the spatial characteristics of traffic have a significant impact on performance, both of which are highly worthy of attention.

Although the proposed WGAN-CNN-BiLSTM model achieved good performance indicators, it had certain limitations. For example,

(1) A lightweight deep learning network model was designed. However, to better meet the real-time response needs of users, the model must be further optimized to reduce its training time on the edge side, thereby improving the QoS of the user experience.

(2) In the cloud–edge collaborative computing architecture studied in this paper, all edge nodes are independently and locally trained, which prevents timely interaction and parameter sharing, resulting in an improved overall performance of the WGAN-CNN-BiLSTM model to a certain extent.

To address the limitations of this study, the WGAN-CNN-BiLSTM model will be further optimized to reduce the number of trainable parameters, thereby making it lightweight. Additionally, federated learning was introduced to improve the cloud–edge collaborative computing architecture, achieving model parameter sharing among all edge nodes while protecting user data privacy. Thus, the overall performance of the lightweight deep learning network model can be improved through large-scale joint training.

Conflict of Interest

The author declare that they have no competing interests.

Funding

None.

References

[1] C. Wang, X. Yu, L. Xu, Z. Wang, and W. Wang, "Multimodal semantic communication accelerated bidirectional caching for 6G MEC," *Future Generation Computer Systems*, vol. 140, pp. 225-237, 2023. https://doi.org/10.1016/j.future.2022.10.036

[2] L. Li, Q. Cheng, X. Tang, T. Bai, W. Chen, Z. Ding, and Z. Han, "Resource allocation for NOMA-MEC systems in ultra-dense networks: a learning aided mean-field game approach," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1487-1500, 2021. https://doi.org/10.1109/TWC.2020.3033843

[3] A. S. Almogren, "Intrusion detection in Edge-of-Things computing," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 259-265, 2020. https://doi.org/10.1016/j.jpdc.2019.12.008

[4] X. Zhao, G. Huang, J. Jiang, L. Gao, and M. Li, "Research on lightweight anomaly detection of multimedia traffic in edge computing," *Computers & Security*, vol. 111, article no. 102463, 2021. https://doi.org/10.1016/j.cose.2021.102463

[5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017 [Online]. Available: https://arxiv.org/abs/1701.07875.

[6] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Computing*, vol. 24, pp. 17265-17278, 2020. https://doi.org/10.1007/s00500-020-05017-0

[7] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access*, vol. 8, pp. 32464-32476, 2020. https://doi.org/10.1109/ACCESS.2020.2973730

[8] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili, "Toward a deep learning-based intrusion detection system for IoT against botnet attacks," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 110-120, 2021. https://doi.org/10.11591/ijai.v10.i1.pp110-120

[9] A. D. Smith, "Event detection in educational records: an application of big data approaches," *International Journal of Business and Systems Research*, vol. 15, no. 3, pp. 271-291, 2021. https://doi.org/10.1504/IJBSR.2021.114936

[10] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A machine learning based intrusion detection system for mobile Internet of Things," *Sensors*, vol. 20, no. 2, article no. 461, 2020. https://doi.org/10.3390/s20020461

[11] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 55-68, 2022. https://doi.org/10.1016/j.jpdc.2022.01.030

[12] X. Jiang, H. Zhang, J. Xu, W. Wu, and X. Xie, "Abnormal network data mining model based on deep training learning," *International Journal of Internet Protocol Technology*, vol. 13, no. 4, pp. 228-236, 2020. https://doi.org/10.1504/IJIPT.2020.110314

[13] A. Al and M. Dener, "STL-HDL: a new hybrid network intrusion detection system for imbalanced dataset on big data environment," *Computers & Security*, vol. 110, article no. 102435, 2021. https://doi.org/10.1016/j.cose.2021.102435

[14] D. Tang, J. Chen, X. Wang, S. Zhang, and Y. Yan, “A new detection method for LDoS attacks based on data mining,” *Future Generation Computer Systems*, vol. 128, pp. 73-87, 2022. https://doi.org/10.1016/j.future.2021.09.039

[15] J. Fei, Q. Yao, M. Chen, X. Wang, and J. Fan, “The abnormal detection for network traffic of power IoT based on device portrait,” *Scientific Programming*, vol. 2020, article no. 8872482, 2020. https://doi.org/10.1155/2020/8872482

[16] S. J. Lee, P. D. Yoo, A. T. Asyhari, Y. Jhi, L. Chermak, C. Y. Yeun, and K. Taha, “IMPACT: impersonation attack detection via edge computing using deep autoencoder and feature abstraction,” *IEEE Access*, vol. 8, pp. 65520-65529, 2020.

[17] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, “Passban IDS: an intelligent anomaly-based intrusion detection system for IoT edge devices,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882-6897, 2020. https://doi.org/10.1109/JIOT.2020.2970501

[18] X. An, X. Zhou, X. Lu, F. Lin, and L. Yang, “Sample selected extreme learning machine based intrusion detection in fog computing and MEC,” *Wireless Communications and Mobile Computing*, vol. 2018, article no. 7472095, 2018. https://doi.org/10.1155/2018/7472095

[19] L. F. Maimo, A. L. P. Gomez, F. J. G. Clemente, M. G. Perez, and G. M. Perez, “A self-adaptive deep learning-based system for anomaly detection in 5G networks,” *IEEE Access*, vol. 6, pp. 7700-7712, 2018. https://doi.org/10.1109/ACCESS.2018.2803446

[20] B. Hussain, Q. Du, S. Zhang, A. Imran, and M. A. Imran, “Mobile edge computing-based data-driven deep learning framework for anomaly detection,” *IEEE Access*, vol. 7, pp. 137656-137667, 2019. https://doi.org/10.1109/ACCESS.2019.2942485

[21] B. Hussain, Q. Du, A. Imran, and M. A. Imran, “Artificial intelligence-powered mobile edge computing-based anomaly detection in cellular networks,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 4986-4996, 2020. https://doi.org/10.1109/TII.2019.2953201

[22] C. A. De Souza, C. B. Westphall, and R. B. Machado, “Two-step ensemble approach for intrusion detection and identification in IoT and fog computing environments,” *Computers & Electrical Engineering*, vol. 98, article no. 107694, 2022. https://doi.org/10.1016/j.compeleceng.2022.107694

[23] H. Bangui and B. Buhnova, “Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms,” *Computers and Electrical Engineering*, vol. 100, article no. 107901, 2022. https://doi.org/10.1016/j.compeleceng.2022.107901

[24] W. Zhong, N. Yu, and C. Ai, “Applying big data based deep learning system to intrusion detection,” *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181-195, 2020. https://doi.org/10.26599/BDMA.2020.9020003

[25] S. Garg, R. Singh, M. S. Obaidat, V. K. Bhalla, and B. Sharma, “Statistical vertical reduction-based data abridging technique for big network traffic dataset,” *International Journal of Communication Systems*, vol. 33, no. 4, article no. e4249, 2020. https://doi.org/10.1002/dac.4249

[26] A. Abid and F. Jemili, “Intrusion detection based on graph oriented big data analytics,” *Procedia Computer Science*, vol. 176, pp. 572-581, 2020. https://doi.org/10.1016/j.procs.2020.08.059

[27] F. Jin, M. Chen, W. Zhang, Y. Yuan, and S. Wang, “Intrusion detection on internet of vehicles via combining log-ratio oversampling, outlier detection and metric learning,” *Information Sciences*, vol. 579, pp. 814-831, 2021. https://doi.org/10.1016/j.ins.2021.08.010

[28] L. Gong, X. Zhang, T. Chen, and L. Zhang, “Recognition of disease genetic information from unstructured text data based on bilstm-crf for molecular mechanisms,” *Security and Communication Networks*, vol. 2021, article no. 6635027, 2021. https://doi.org/10.1155/2021/6635027

[29] M. A. Bou-Rabee, M. Y. Naz, I. E. Albalaa, and S. A. Sulaiman, “BiLSTM network-based approach for solar irradiance forecasting in continental climate zones,” *Energies*, vol. 15, no. 6, article no. 2226, 2022. https://doi.org/10.3390/en15062226

[30] Y. Zhan, S. Sun, X. Li, and F. Wang, “Combined remaining life prediction of multiple bearings based on EEMD-BILSTM,” *Symmetry*, vol. 14, no. 2, article no. 251, 2022. https://doi.org/10.3390/sym14020251

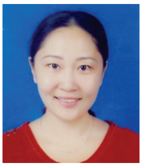
[31] Y. L. Miao, W. F. Cheng, Y. C. Ji, S. Zhang, and Y. L. Kong, “Aspect-based sentiment analysis in Chinese based on mobile reviews for BiLSTM-CRF,” *Journal of Intelligent & Fuzzy Systems*, vol. 40, no. 5, pp. 8697-8707, 2021. https://doi.org/10.3233/JIFS-192078

[32] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a search space odyssey,” 2017 [Online]. Available: https://arxiv.org/abs/1503.04069.

[33] D. Ma, Y. Guo, and S. Ma, “Short-term subway passenger flow prediction based on GCN-BiLSTM,” *IOP Conference Series: Earth and Environmental Science*, vol. 693, no. 1, article no. 012005, 2021. https://doi.org/10.1088/1755-1315/693/1/012005

[34] S. Choudhary and N. Kesswani, “Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT,” *Procedia Computer Science*, vol. 167, pp. 1561-1573, 2020. https://doi.org/10.1016/j.procs.2020.03.367

[35] V. Kanimozhi and T. Prem Jacob, “Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing,” *ICT Express*, vol. 5, no. 3, pp. 211-214, 2019. https://doi.org/10.1016/j.icte.2019.03.003



Yue Wang **https://orcid.org/0009-0008-2052-3470**

She graduated from Huazhong University of Science and Technology with a master's degree in engineering in 2001 and worked at Nanyang Institute of Technology, whose research interests include big data, cloud computing, computer software and theory.