J Grid Computing (2023) 21:60
https://doi.org/10.1007/s10723-023-09703-9

RESEARCH

Anti-Attack Intrusion Detection Model Based on MPNN and Traffic Spatiotemporal Characteristics

Jiazhong Lu · Jin Lan · Yuanyuan Huang · Maojia Song · Xiaolei Liu

Received: 6 April 2023 / Accepted: 15 October 2023 / Published online: 28 October 2023

Abstract Considering the robustness and accuracy of conventional intrusion detection models are easily influenced by adversarial attacks, this work proposes an anti-attack intrusion detection model based on a message-passing neural network with traffic spatiotemporal features. Our model can not only effectively distinguish and correlate upstream and downstream traffics, but also clearly embody the relationship between different traffics from the same source node and destination node by the established graph structure. We also improve the model by utilising the characteristics of the existing network attacks, which indicates that the traffic spatiotemporal characteristics could achieve high accuracy on attack traffic detection. Compared to the previous model, extensive experiments show that our proposed model outperforms other similar models in performance. For the multi-classification tasks, our model can effectively detect most of the network attacks and outperform traditional machine learning methods. In terms of model robustness, our model is less affected by adversarial attacks than traditional machine learning models.

Keywords Intrusion detection · Traffic spatiotemporal features · Message passing neural network · Anti-attack · Robustness

J. Lu · J. Lan · Y. Huang
Advanced Cryptography and System Security Key Laboratory of Sichuan Province, School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China
e-mail: ljz@cuit.edu.cn

J. Lan
e-mail: 3200809018@stu.cuit.edu.cn

Y. Huang
e-mail: hy@cuit.edu.cn

M. Song
The University of Leeds, School of Electronic and Electrical Engineering, University of Leeds, Leeds LS2 9JT, UK
e-mail: el19m2s@leeds.ac.uk

X. Liu (✉)
Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan 621900, China
e-mail: luxaole@gmail.com

1 Introduction

Mchine learning has become the mainstream towards attack traffic detection in academics [1–3]. However, ever-growing problems have emerged with these methods. Firstly, a huge amount of high-dimensional data is generated. Traditional machine learning models that focus on the superficial characteristics of traffics are vulnerable to adversarial attacks and cannot be adapted to the high-dimensional data scenarios. Secondly, the data is usually imbalanced. Since the amount of abnormal traffic is much less than normal traffic, the usual algorithms hardly capture its characteristics, which means most of the currently proposed intrusion detection algorithms cannot well

detect small-scale abnormal traffics. Especially in the real-life production environment, the problem of data imbalance becomes more severe. Thirdly, there are quite a few intrusion detection models are trained and tested based on offline data sets, however, real-time detection of attack traffic is required for industrial environments, leaving an important problem faced by current intrusion detection models.

Graph neural networks [4] have achieved well impressive performances in many fields, e.g., the structure prediction for the molecular model of the new coronavirus [5]. Similarly, the graph neural networks can be used to solve intrusion detection challenges by focusing on extracting the characteristics of each traffic and the characteristics between different traffics. The intrusion detection system based on message passing mechanism could obtain the deep characteristics of the traffic without the accuracy reduction in the course of adversarial attack. Moreover, intrusion detection based on the message-passing neural network can make better use of the strong correlation between abnormal traffic in network attacks, such as DOS attacks that continuously send a large number of packets, and botnets with periodic attacks.

Therefore, this paper involves security issues such as the inherent mechanism of network attacks, lateral penetration and directional movement, with a special focus on the nodes, flows and processes of network attacks. By clarifying the mechanism of hidden and detour attacks in the actual network attack, we desgin an intrusion detection model with the combination of MPNN(Message-passing neural networks) and traffic spatiotemporal features, which reveals the relationship between traffics for better robustness in adversarial attacks.

To sum up, our contribution is as follows:

1. Based on MPNN, an intrusion detection model is proposed, which encodes the source address, destination address and other data flow information as nodes on the graph network, and accurately shows the correlation between the flows.
2. We applies the newly proposed graph neural networks model to intrusion detection. The model achieves an accuracy comparable to the latest machine learning models in the case of detect classification. When the model faces adversarial attacks, the accuracy of existing machine learning detection models will drop significantly, but our proposed intrusion detection model will not decrease in accuracy
3. Based on the intrusion detection model proposed by MPNN, the detection efficiency remains stable in the face of attack traffics sent by different tools (Note: the average size of attack traffics sent by different attack tools is different).

Our paper consists of 5 sections, the first section is the introduction, the second section is related work, the third section is our proposed model, the fourth section is the experiment, and the fifth section is the conclusion.

2 Related Work

Network attacks on the Industrial Internet are mainly characterized by slow and multi-stage. Xu [6] et al. proposed a bidirectional LSTM network with multiple feature layers on three Industrial Internet of Things (IIoT) datasets. The proposed method reduced the false positive rate by 46.79% compared to the existing methods. Meanwhile, it also detects attacks with different intervals on the IIoT dataset. Süzen [7] et al. used deep belief networks to design an intrusion detection system, which ensures network security by controlling the traffic in industrial control systems. Experiments showed that they proposed system had better accuracy in intrusion detection and classification. Gao [8] et al cleverly integrated the inter-class self-learning spatial distribution algorithm and cognitive computing into intrusion detection. Experimental results show that their method has high accuracy. Liang [9] et al proposed method makes up for the problem that clustering optimization cannot be performed due to the lack of features. Their result presented a notable improvement in both detection rate and operating time. The anomaly data detection accuracy rate reached 97.80%, and the false positive rate was reduced by 8.80%.

While tag-based intrusion detection systems provide better early warning of known attacks, but not for unknown behaviours, anomaly-based intrusion detection systems can predict unknown attack behaviours, but there is also a potential for false negatives in terms of known attack behaviours. Khraisat [10] et al designed an intrusion detection system using C5 classifier and SVM to detect IoT botnets. Huang [11]

proposed the use of Hidden Markov Models for intrusion detection. Abdel-Basset [12] et al. used forensics to detect cyberattacks, and their method was able to detect anomalies in large amounts of IoT data.

Most of the existing intrusion detection uses offline data, which cannot meet the requirements of real-time attack information collection. Kim [13] et al. Combining features with machine learning can classify and detect data well. Wang [14] et al. using autoencoders for network attack detection. Their method could simultaneously predict and reconstruct the input data, using the rate of change to locate the most suspicious potentially compromised devices. Awotunde [15] et al. An intrusion detection model suitable for Industrial Internet of Things is designed, which uses deep learning techniques. Dutta [16] et al. A meta-classifier was constructed using deep learning techniques. The data were first pre-processed, and then the features were solved using a deep sparse self-encoder (DSAE). The method could handle the most complex datasets well with high accuracy. Jahromi [17] et al. designed a framework for defending against cyber-attacks for cyber-physical systems. The framework can resist existing network attacks well. Yang [18] et al. proposed a data fusion method based on the characteristics of traffic data. Consequently, the accuracy of anomaly detection was improved. Their algorithm worked well in wireless sensor networks. Basmati [19] et al. proposed a lightweight intrusion detection model based on the node characteristics of network attacks, which can solve the problem of insufficient computing power of some nodes in the industrial Internet. Hua [20] et al. desgin Network Attack identification method for industrial control networks based on the RNN-GBRBM feature decode, which is based on the manual selection of features in the original data packets, followed by the osPCA measurement to identify the traffic features, and the detection of the traffic at last.

Now, the graph neural network can be applied to various occasions, and has achieved good results. Luo [21] et al. based on the GraphSAGE model, edge information is added. More features and information of the graph can be obtained, making it more widely used. Arno [22] et al. using the characteristics of graphs and neural networks, an end-to-end network attack detection method is proposed. Yang [23] et al. desigined a multi-classification method using binary tree SVN and OOA division. This method outperforms other methods in terms of accuracy. Sahoo [24] proposes a method of sequential generalization that improves training speed and detection accuracy. Khammmassi [25] proposed a method based on machine learning and feature fusion detection, which has a good performance in public data. Wang et al. [26] studied the public DDoS attack SDN dataset, which includes 23 characteristics of ICMP, UDP, TCP normal traffic and attack traffic. They detect with multiple machine learning classifiers, and these methods effectively detect DDoS attacks in SDN.

Existing intrusion detection methods have their advantages[27–30,33]. Liu et al. established a TLTD model framework to enhance the robustness of IoT systems[31], and also demonstrated that the robustness of artificial intelligence algorithms is related to the complexity of the model and dataset[32]. Network intrusion detection can detect the initial stage of the attack, during which it can be combined with malicious code detection for double detection. Network intrusion warnings and intranet scanning can prevent some penetration attacks. Using deep learning can fully detect the packet content and header, and it can also help us detect some 0-day attacks. However, none of these methods can defend against the attack with abnormal samples (attack samples whose partial feature information deviates from normal). When the attacker modifies the traffic feature value, detection performance based on machine learning or deep learning will be greatly reduced.

For complex and changeable network attacks, we improve the original message-passing neural network and add new traffic spatiotemporal characteristics. An intrusion detection model based on message-passing neural network and traffic space-time tailoring is proposed, which results in an accuracy comparable to that of traditional machine learning. At the same time, the detection performance of the traditional machine learning model is significantly degraded when adversarial attack behaviour is encountered, our method alleviates such a significant reduction.

3 Intrusion Detection Model Constructions

In our intrusion detection model, the existing MPNN is improved by introducing traffic spatiotemporal features, instead of just relying on a few basic

characteristics of network traffic. Supervised learning methods are then used to train on public datasets, and finally, the experiment is conducted on real network attacks.

3.1 Extraction of Traffic Spatiotemporal Features

In our previous work [34–37], we used the spatiotemporal features of traffic for anomaly detection, such as temporal features, combined features, and protocol features. In our model, we not only use the features proposed by our previous work (such as the average number of packets of upstream and downstream traffic, the total size of upstream and downstream data packets, traffic duration, etc.), but also some new features proposed according to different network attacks (such as TCP handshake flag features, such as ACK, FIN, SYN), in a total of 29 features, as shown in Table 1. It should be noted that among the features we selected, the five tuples of traffics are associated with upstream and downstream traffic.

These features are chosen because some network attacks require automatic commands sent by programs. This automatic sending of commands some kind of regularity at a certain time. The phishing email transfer process uses the fixed protocols, the attacker can send traffic packets in multiple formats and very small size. So, the process of establishing the connection could be very smooth, which means the phishing email in the TCP handshake state is the same as the normal email data. However, in botnets and malware, the attackers need to control the target host and carry out the next attack, this will cause the TCP handshake to fail with a high probability.

In TCP handshake, ACK can be used simultaneously with SYN and FIN. If there is only one SYN, it means the connection is established without confirmation, and most unreachable attacks have a single SYN. Most firewalls detect SYN/FIN packets when

Table 1 Flow spatiotemporal characteristics

Traffic Characteristics
Source and Destination Port, Source and Destination IP, Protocol
The minimum value of downstream packets
The average amount of packets for upstream traffic
Total upstream packet size
The average size of upstream packets
The standard deviation of upstream traffic size
The minimum time interval of upstream data packets
The minimum traffic time interval
Average traffic interval
Standard Deviation of Downlink Traffic Size
Upstream substream size in bytes
Duration
Standard deviation between two streams
Minimum time for stream activity
Average time of stream activity
Downstream continuous data packet average time
Downstream substream size in bytes
Packets transmitted per second
Number of packets with ACK
Upstream data contains the number of PSH flags
Number of packets with SYN
Handshake Situation
Valuse of ACK, URG, FIN, and RST
Repeatedly responds when ACK = 1 in destination IP
SYN = 1,FIN=1 and ACK = 1 in destination IP

there is a FIN and RST but no SYN, and when such packets appear in the network, it may indicate that the network is under attack. At the same time, ACK/FIN data indicates that a TCP connection is completed, so normal FIN data packets are always marked with the ACK flag. A value of NULL indicates a packet without any TCP flags. For normal sending packets cannot appear TCP packets with any of the above-combined flags.

3.2 Graph Construction and Data Selection

In the graph structure we constructed, the traffic is firstly converted into a graph structure form, and the source *IP* node *S* and destination node *D* are converted into points in the graph structure. Secondly, given a stream $f \in E$, each stream is represented as a node on the graph structure. At the node *S* and the node *D*, two undirected edges are created, one from the $S \rightarrow f$, and the other from $f \rightarrow D$.

The structure focuses on the structural features between traffic, which can distinguish and correlate upstream and downstream traffic. Besides, the graph structure can clearly represent the relationship between different traffic connected to the same source node and destination node. The graph structure is shown in Fig. 1. After two information transfer processes (such as $S2 \rightarrow f2 \rightarrow D3$, $D3 \rightarrow f1$), $f1$ can get the relevant information of $f2$, and $f2$ can also get the relevant information of $f1$. Traffic with the same source node or the same destination node is more likely to be the same type of traffic. The message-passing neural network can use the same IP node to transmit the information of different traffic nodes to each other, thus, it improves the accuracy of traffic node classification.

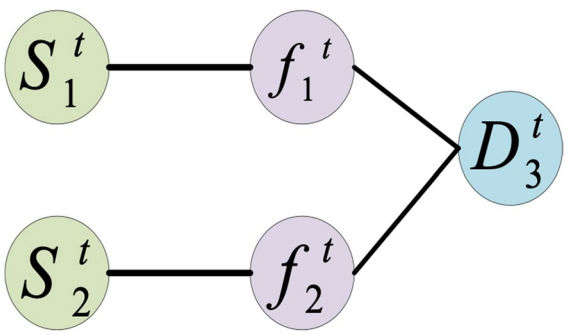

Fig. 1 The graph structure used in our method

3.3 Construction of Intrusion Detection Model

Our intrusion detection model has a graph structure as input, which is based on a non-standard message-passing neural network. Here, h_i^t is the hidden state of node i in the K-th layer, the initial state of the hidden state is the X_i. In the process of initializing the data, the flow node and the IP node are initialized to different states. Assuming that the initial feature of the flow is $X_i = [x_1, \ldots\ldots x_k]$. So the flow node i represents the initial hidden state:

$$h_i^0 = \left[x_1, \ldots, x_k, 0, \ldots, 0\right]. \tag{1}$$

The hidden state vector length is larger than the number of elements in the initial feature vector, then fill the vacant part with 0. In reality, the basic characteristics of traffic cannot be used as the basis for intrusion detection to judge attack behaviour. Therefore, we initialize the feature vector to 1 for the *S* and *f*. Then, the nodes in the graph structure are divided into two categories, one is IP-encoded nodes, and the other is traffic-encoded nodes. The role of IP nodes is to aggregate information from different traffic flows. The role of traffic nodes is to distribute messages. Our information transfer process is shown in Fig. 2.

For the message-passing flow, we first set a learnable parameter δ_{type}. Then the connection of the hidden information of the two nodes is constructed, as shown in Eq. 2. δ_{type} contains two possible functions, δ_{sf} represents an edge $(S \rightarrow f)$, δ_{sd} represents an edge $(f \rightarrow D)$, and then uses an aggregate function to calculate the message on each node:

$$\alpha_i^t = \frac{1}{|\,N(i)\,|} \sum\nolimits_{j \in N(i)} \delta_{type}\left(h_i^t \parallel h_j^t\right). \tag{2}$$

Finally, we compute the hidden state of node i in the next layer and apply the function δ_{type} to the aggregated messages and the node, as shown in Eq. 3. Similar to the message-passing process, δ_{type} includes two different learnable functions ($\delta_h$and δ_f), $\delta_h$is used to update the hidden state of destination and source nodes, and $\delta_f$is used to update the hidden state of flow nodes:

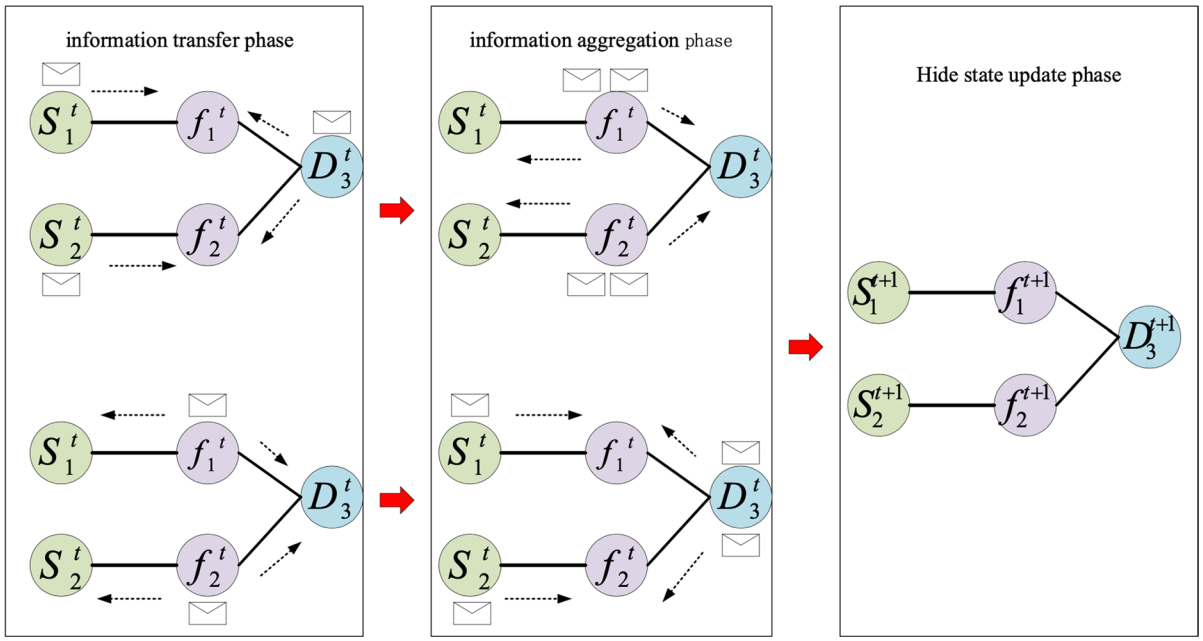

Fig. 2 The information on messages transfer, aggregation and hiding

$$h_i^{t+1} = \delta_{type}\left(h_i^t \parallel \alpha_i^t\right). \tag{3}$$

The Readout function is defined as follows:

$$y_i = r\left(h_i^T\right). \tag{4}$$

The function $r(\bullet)$ takes in the final state of each flow and outputs the predicted class.

Since there are lateral penetration and detour attacks in network attacks, it is difficult to detect the path and traffic of network attacks. Also, correlation analysis cannot be performed well. Therefore, in our a model, we added an additional connection layer to become a four-layer fully connected layer, and the activation function is still ReLU. The traffic correlation model we propose is shown in Fig. 3. After the original data traffic is constructed as graph-structured data, two message-passing processes are firstly completed, and then four fully connected layers are accessed to train the model. The final output is the corresponding training result. In the message-passing process of the MPNN layer, the *S* and *f* pass information to the upstream or next flow node. Then the flow node transmits message to the *S* and *f*. If a single-layer MPNN is used, it is impossible to transmit the flow message of the same source or the same destination to each other, so we choose a two-layer MPNN. In the iterative process of the fully connected graph neural network, all nodes are the sum of their features, and the aggregated representation of a node does not contain its features, which is the feature aggregation of adjacent nodes. At the same time, to better aggregate the features of adjacent nodes, a four-layer fully connected neural network is sampled.

$\left(S_1^t \rightarrow f_1^t\right)\&\left(D_1^t \rightarrow f_1^t\right)$ indicates that the *S* and *f* transmit messages to the flow node, and $f_1^t \rightarrow S_1^t \& f_1^t \rightarrow D_1^t$ indicates that the flow node transmits messages to the source node and the destination node. In $H^{(l+1)} = \sigma\left(\tilde{A}H^{(l)}W^{(l)}\right)$, A represents the adjacency matrix. In $\tilde{A} = A + I$, I represents the identity matrix. W represents a trainable matrix. $H^{(l)}$ is the i-th feature matrix. $H^{(l+1)}$ is the $l+1th$ feature matrix. σ is the activation function. In the $r(\bullet)$, the activation function is the Softmax in formula (5). All possible outputs are represented by one-hot encoding, and Softmax is used for weighting and normalization. $\theta(i)$ is the input obtained from Softmax. It computes a normalized value for each class, where the numerator is the index value of the class, and the denominator is the sum of the index values for all classes.

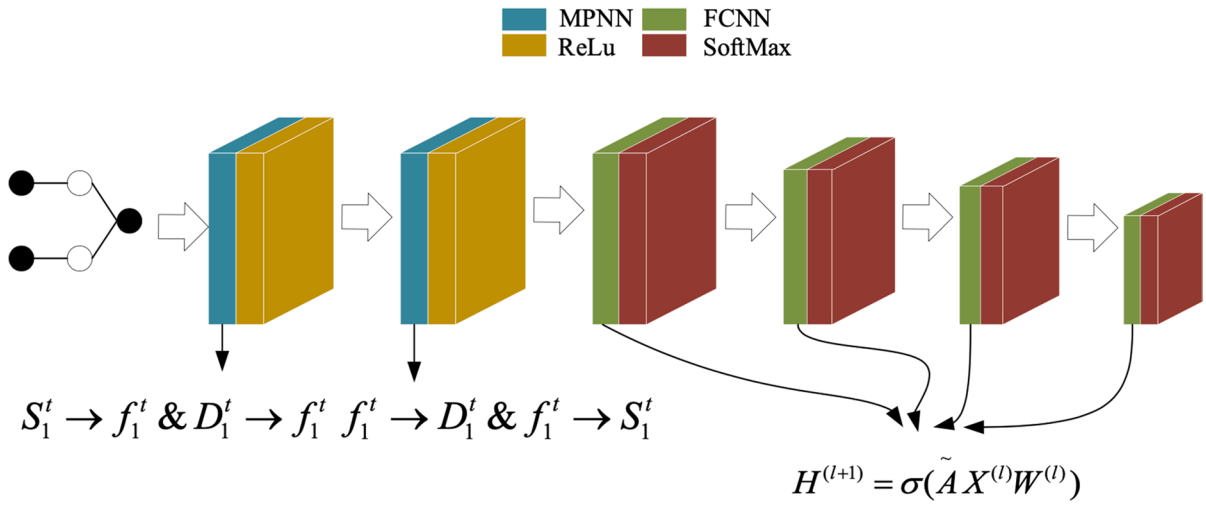

Fig. 3 Message Passing Neural Network Applied to Network Attack Behavior Detection Model

$$P\left(y = j|\theta^{(i)}\right) = \frac{e^{\theta^{(i)}}}{\sum_{j=0}^{k} e^{\theta_k^{(i)}}}, \tag{5}$$

where

$$\theta = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_kx_k = \sum_{i=0}^{k} w_ix_i = W^TX,$$
$$L = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_{i=1}\sum_{c=1} y_{ic}\log\left(\mathrm{p}_{ic}\right) \tag{6}$$

M is the number of categories, y_{ic} is the sign function (0 or 1), and the category of sample i is equal to c=1, otherwise c=0. P_{ic} is the predicted probability of sample i is the category c. In the case of binary classification, it is as formula (7):

$$L = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_i -\left[y_i \bullet \log\left(p_i\right) + \left(p_i\right) + (1-y) \bullet \log\left(1-p_i\right)\right] \tag{7}$$

where $y_i$represents the label of the i sample, the positive class is 1, and the negative class is 0. $P_i$represents the probability that sample i is predicted to be a positive class.

3.4 Intrusion Detection Model

Figure 4 shows an overview of the intrusion detection model we proposed based on MPNN. First, the network data flow is reconstructed into graph-structured datas. Then use the graph structured data during training. Finally, the traffic nodes are classified.

The network data flow is reconstructed into a graph structure: the network data flow is the basic data transmission form in the network, and most currently intrusion detection systems are also based on this structure. The data flow not only includes the source IP, port, destination IP and port of the data information, but also includes the size, duration and other information of the data flow.

In order to better show the correlation between flows, in the MPNN-based intrusion detection system, the S node is first encoded as the source node, the f node is encoded as the target node, and the other information is encoded as the flow node. In the process of encoding nodes, information such as IP cannot be used as the information to judge the attack behavior in the actual intrusion detection system, so the feature vectors of the S and f are initialized as vectors with all 1s. During the encoding process of the stream node, the insufficient part of the stream node is filled with 0.

Model training: we proposed the MPNN-based intrusion detection system proposed , the message passing process is completed twice, and then the four fully connected layers are connected, and finally the flow nodes are classified. The hidden layer is 128. To improve the

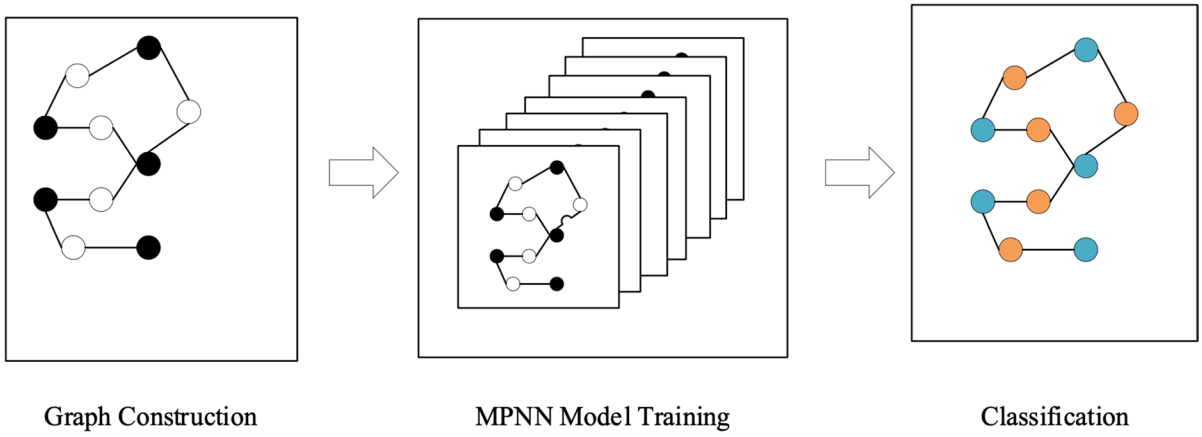

Fig. 4 Structure of MPNN-based intrusion detection system

Table 2 The experimental conditions

Operating system	Windows10
Memory	32G
CPU	AMD Ryzen3950X
GPU	NIVIDIA 3090
Video memory	32G
Hard disk	SAMSUNGMZ-VLB512H-BJQ-00007

generalization ability of the model, we set a dropout value of 0.3 between each layer. In our model, the activation function is the Softmax function, and the nonlinear activation function is the ReLu function.

Classification: We use the public set to evaluate the model performance. First, convert the graph structure of the test set to be consistent with the training set, and then use the trained MPNN model to assign the flow nodes of the Softmax layer to make the corresponding probabilities different. Finally, use the real class labels to evaluate the model.

4 Experiments

4.1 Experimental Conditions

Our experimental conditions include operating system, memory, GPU, CPU, video memory and hard disk, in Table 2.

Table 3 CIC-IDS207 data set traffic types and quantities

Traffic type	quantities
BENIGN	395413
Bot	1959
DDoS	127186
DoS-GoldenEye	10242
DoS-Hulk	229927
DoS-Slowhttptest	5475
DoS-slowloris	5770
FTP-Patator	7907
Heartbleed	11
Infiltration	36
PortScan	157940
SSH-Patator	5866
Web Attack Brute Force	1498
Web Attack Sql Injection	21
Web Attack XSS	650

4.2 Dataset

The data set contains 25 types of user behaviors, which use a variety of protocols, such as HTTP, FTP, etc. At the same time, it also includes 15 kinds of network attacks, and the specific attacks are shown in Table 3. In our comparative experiments, we also use the CICIDS2017 dataset. The CICIDS2017 dataset contains benign and up-to-date common attacks similar to real-world data (PCAPs). It also includes the results of network traffic analysis with CICFlowMeter, using tagged flows based on timestamp, source

and destination IP, source and destination port, protocol and attack (CSV file).

4.3 Evaluation

We use TP, TN, FN and FP to evaluate our model. In the experiment, we define two labels for the data set. The first label indicates whether the traffic is abnormal or not. The second label represents the traffic category, which is empty for normal traffic, and its classification for abnormal traffic. We use the first label for binary classification and the second label for multi-classification. In the experiments, we use 80% data set for training and 20% data set for testing. The biggest problem in training is that the normal data set and abnormal data are not equal. In reality, the amount of normal data accounts for about 95% of the total data. Therefore, we need to eliminate normal data, 15% of the normal samples are randomly selected to participate in the calculation of the message-passing process, and the remaining normal traffic is discarded.

4.4 Analysis

To demonstrate the superiority of our model in accuracy and robustness. We compare with the MLP method proposed by Arnaud [22], the MBT-SVM method proposed by Yang [23], the SGK method proposed by Sahoo [24], and the NSGA2-LR method proposed by Khammassi [25], as shown in Figure 5. Our proposed model achieves 99.81% accuracy on the CIC-IDS2017 dataset and surpasses the accuracy of MLP by 1.46%.

The proposed model can also achieve an accuracy that is better than or similar to traditional machine learning in the detection of specific attack behaviours. As shown in Figures 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 and 17, we compared the precision of five algorithms on multi-classification.

The following experiments will analyse the robustness of the our model for common adversarial attacks. A DDoS attack uses multiple hosts to access the target host at the same time, resulting in the target server being unable to function normally. To launch an attack on the target host faster and reduce the overhead of the attacking host, the data packets of the DDoS attack traffic will not have too much payload, but only need to achieve the minimum packet

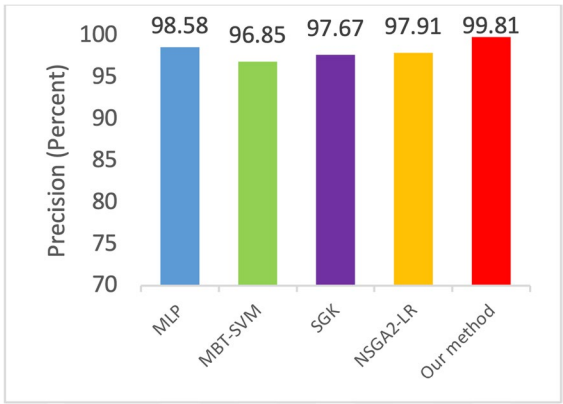

Fig. 5 The accuracy of our method compared with other methods

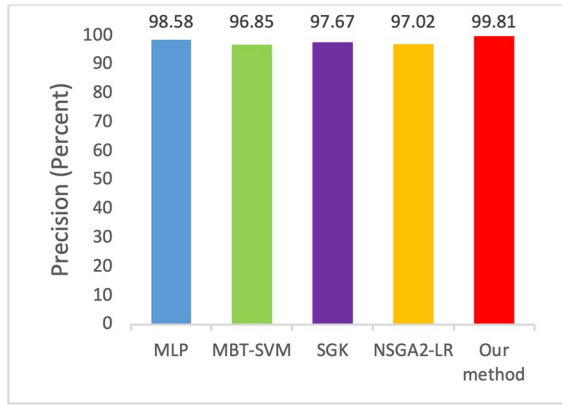

Fig. 6 The precision of Benign

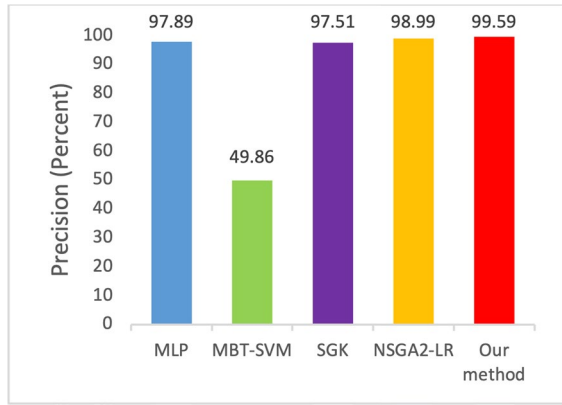

Fig. 7 The precision of Benign SSH-Patator

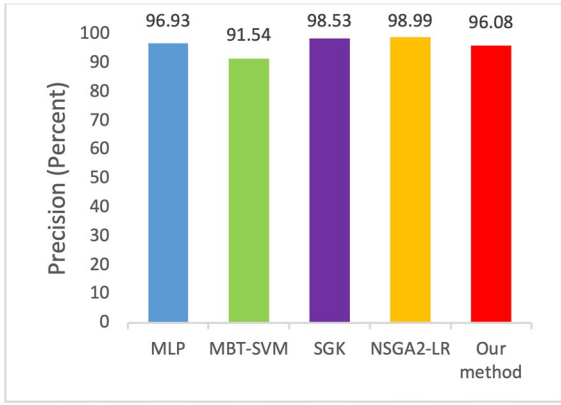

Fig. 8 The precision of DoS GoldenEye

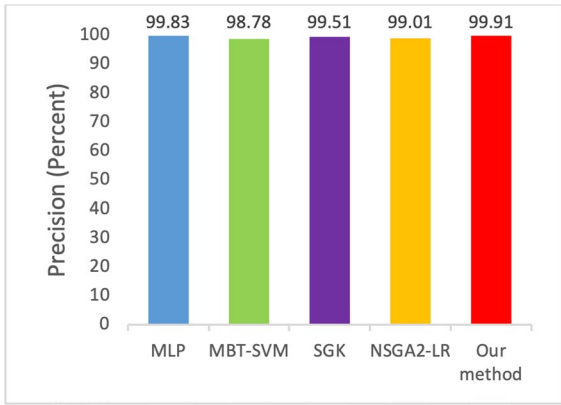

Fig. 9 The precision of PortScan

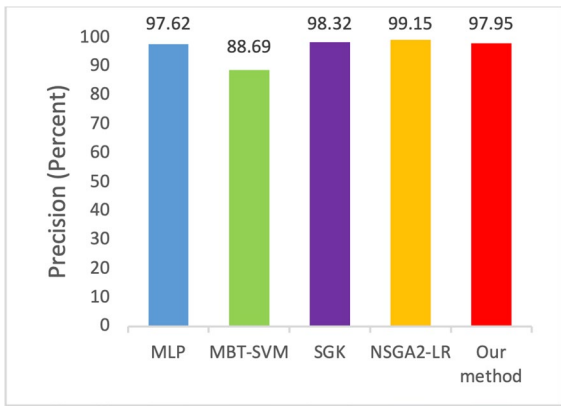

Fig. 10 The precision of DoS Slowhttptest

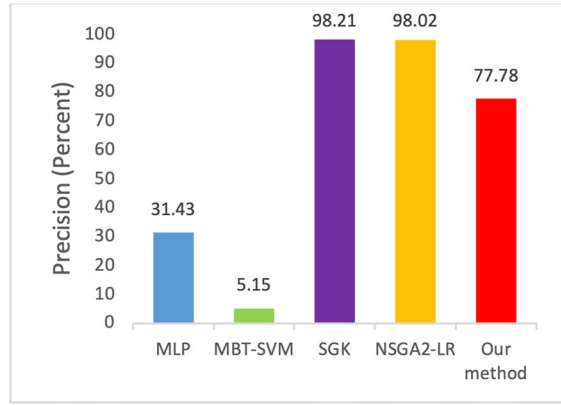

Fig. 11 The precision of Web Attack Brute Force

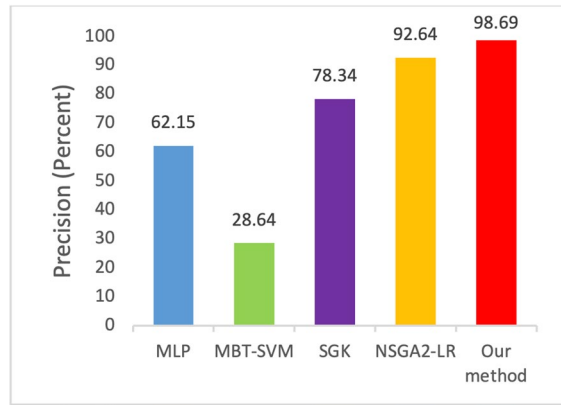

Fig. 12 The precision of Bot

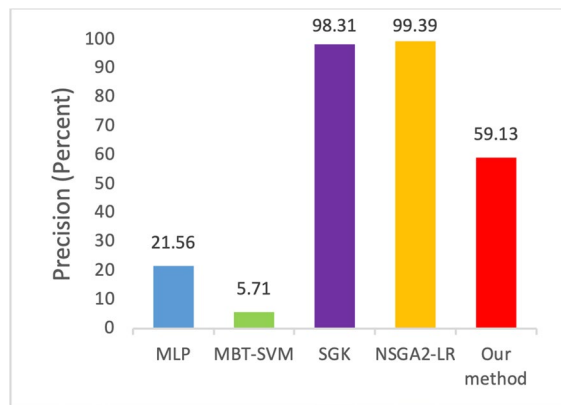

Fig. 13 The precision of Web Attack XSS

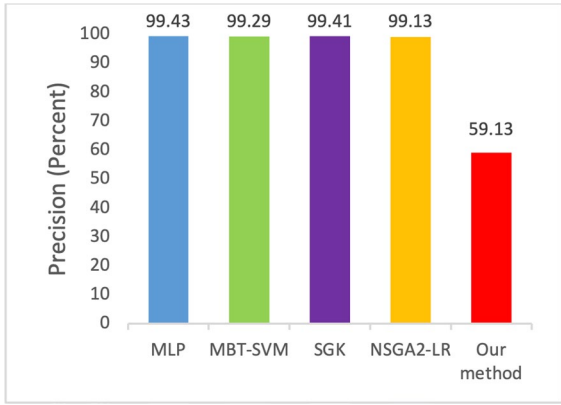

Fig. 14 The precision of DDoS

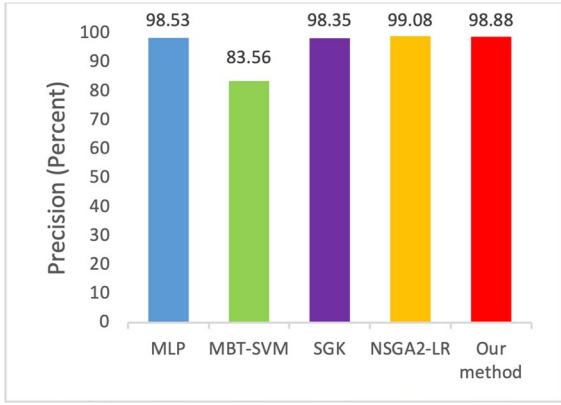

Fig. 15 The precision of DoS slowloris

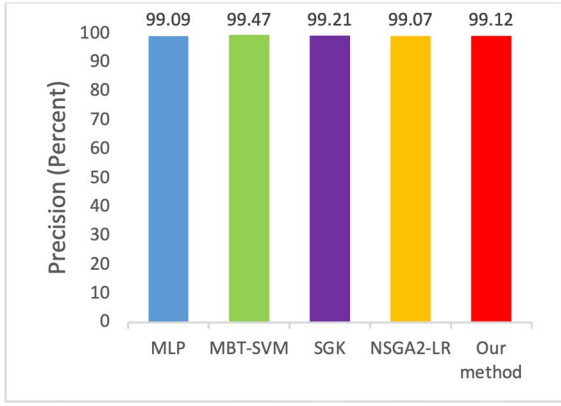

Fig. 16 The precision of FTP-Patator

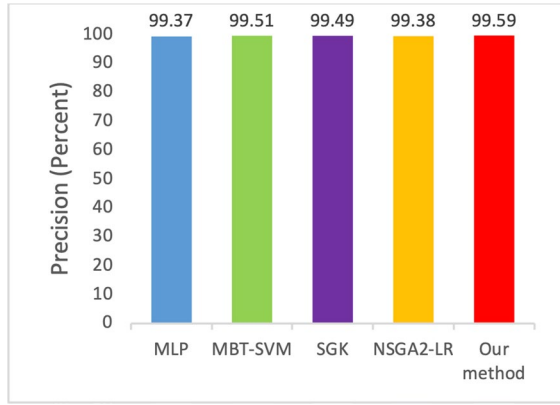

Fig. 17 The precision of DoS Hulk

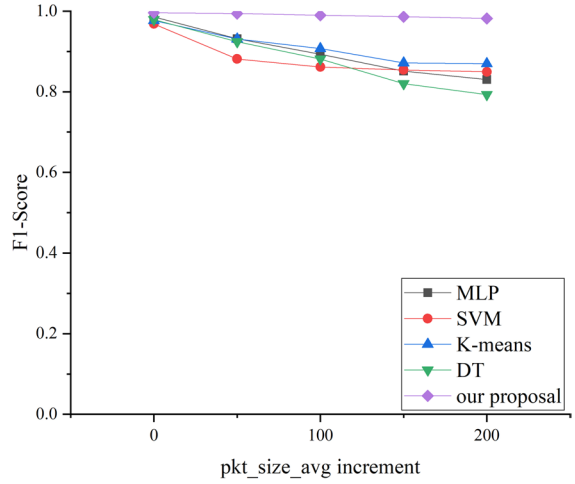

Fig. 18 F1-score changes with packet size

result of network transmission. The purpose of DDoS is to consume the bandwidth resources of the target host. As detecting the packet size of a single flow can usefully identify some network attacks, especially in DDoS attack traffic, the packet size is one of the most important detection indicators. In the first adversarial attack experiment, we manually modified the attack packet, using the byte size of [0-300] to the original packet size. The experimental results are shown in Figure 18.

For traditional machine learning, the data packet size is a key indicator for identifying attack traffic, and the accuracy of the model will decrease after being artificially modified. However, our proposed intrusion detection system based on MPNN and

traffic spatiotemporal features can maintain good accuracy even under adversarial attacks.

The principle of the DoS Slowhttptest attack is to find a way to make the server wait. When the server keeps the connection waiting, the server will always consume a part of the limited resources to maintain the current process. When machine learning detects DoS Slowhttptest attacks, the inter-packet arrival time will be used as an important judgment indicator. We modify the inter-packet arrival time of the test set by [0-5] seconds, and then compare the changes in the accuracy of different models, in Fig. 19.

In the inter-packet arrival time, other methods show a significant decrease in accuracy. Inter-packet arrival time is an important condition for machine learning models to judge traffic. When adversarial attacks are conducted against inter-packet arrival time, the accuracy of the machine learning model will be significantly affected. When launching a brute force attack, different attack tools may have different sizes of packets. But when using the same attack software, the packet size will not fluctuate too much. At the same time, in the normal communication process, the size of the packet changes randomly, there is no statistical law, and the fluctuation is obvious. We try to modify the average packet size to compare the accuracy between different model s, in Fig. 20. From the above three experiments, the traditional intrusion detection model has

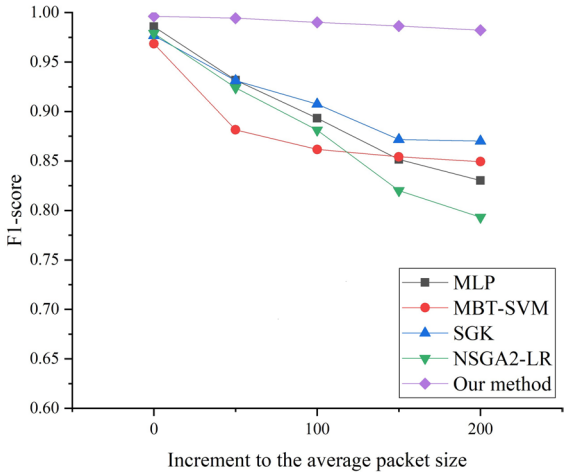

Fig. 19 The relationship between F1-score and inter-package arrival time

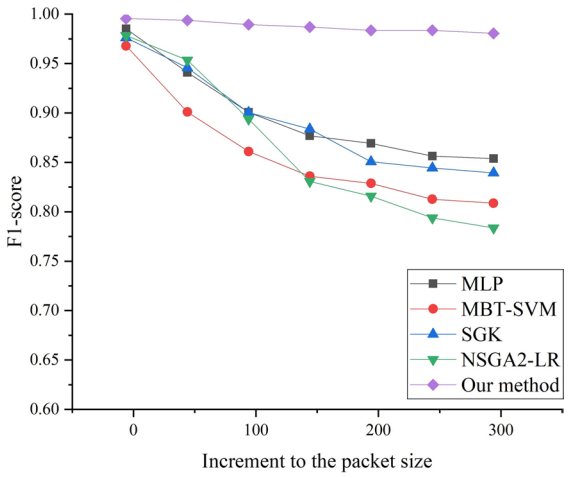

Fig. 20 The relationship between F1-score and inter-package arrival time

less robustness towards adversarial attacks, but our model still maintains good accuracy.

5 Conclusion

Currently, the three major problems restricting the performance of intrusion detection model s are small sample detection, adversarial attacks and real-time detection. Our model is not just relied on a few basic characteristics of network traffic but utilises the traffic spatiotemporal characteristics. Our intrusion detection model is bettet than the other machine learning models in detecting botnets and can still maintain good robustness when the model is confronted with adversarial attacks, although the detection effect of Web Attacks is slightly inferior to other machine learning models. When the adversarial attack is carried out against the model, the accuracy of the machine learning model drops significantly, but the accuracy of the model proposed in this section does not change significantly. The model we proposed has achieved better results than traditional machine learning models in terms of robustness, but it still cannot solve the problem of small samples. In the future, it is an area worthy of further exploration to try to use the combination of graph neural network model and machine learning model to improve the robustness of the model and also improve the ability of the model to detect small sample attacks.

Acknowledgments The authors would like to acknowledge the support of the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No.SKLACSS-202110). Thanks are also expressed to the National Natural Science Foundation of China (Grant No.62102049), the National Key R&D Program of China (Grant No.2017YFB0802300) and the Natural Science Foundation of Sichuan Province(Grant No.2022NSFSC0557).

Authors' Contributions Jiazhong Lu and Jin Lan wrote the main manuscript text, Yuanyuan Huang prepared figures 1-20. Maojia Song check the manuscript, and Xiaolei Liu support the experiment and fund.

Funding 1. Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No.SKLACSS-202110).

2. National Natural Science Foundation of China (Grant No.62102049, 62102379)

3. The National Key R&D Program of China (Grant No.2017YFB0802300)

4. Natural Science Foundation of Sichuan Province(Grant No.2022NSFSC0557).

Declarations

Competing interests The authors declare no competing interests.

References

1. Ferrag, M.A., Shu, L., Friha, O., et al.: Cyber Security Intrusion Detection for Agriculture 4.0: Machine Learning-Based Solutions, Datasets, and Future Directions. IEEE Journal of Automatica Sinica. IEEE. **9**(3), 407–436. (2021). https://doi.org/10.1109/JAS.2021.1004344
2. Gao Y, Chen J, Miao H, et al. 2022. Self-Learning Spatial Distribution-Based Intrusion Detection for Industrial Cyber-Physical Systems. IEEE Transactions on Computational Social Systems. IEEE, 1-10. https://doi.org/10.1109/TCSS.2021.3135586
3. Mubarak, S., Habaebi, M.H., Islam, M.R., et al.: Industrial Datasets with ICS Testbed and Attack Detection Using Machine Learning Techniques. Intelligent Automation And Soft Computing. Tech Sci Press. **31**(3), 1345–1360 (2022). https://doi.org/10.32604/iasc.2022.020801
4. Such, F.P., Sah, S., Dominguez, M.A., et al.: Robust spatial filtering with graph convolutional neural networks. IEEE J Select Top Sign Process. IEEE. **11**(6), 884–896. (2017). https://doi.org/10.1109/JSTSP.2017.2726981
5. Kapoor A, Ben X, Liu L, et al. 2020. Examining covid-19 forecasting using spatio-temporal graph neural networks. arXiv:2007.03113. Retrieved from https://arxiv.org/abs/2007.03113
6. Li, X., Xu, M., Vijayakumar, P., et al.: Detection of low-frequency and multi-stage attacks in industrial internet of things. IEEE Transactions on Vehicular Technology. IEEE. **69**(8), 8820–8831. (2020). https://doi.org/10.1109/TVT.2020.2995133
7. Süzen, A.A.: Developing a multi-level intrusion detection system using hybrid-DBN. Journal of Ambient Intelligence and Humanized Computing. Springer. **12**(2), 1913–1923 (2021). https://doi.org/10.1007/s12652-020-02271-w
8. Gao Y, Chen J, Miao H, et al. 2022. Self-Learning Spatial Distribution-Based Intrusion Detection for Industrial Cyber-Physical Systems. IEEE Transactions on Computational Social Systems. IEEE, 1-10. https://doi.org/10.1109/TCSS.2021.3135586
9. Liang, W., Li, K.C., Long, J., et al.: An industrial network intrusion detection algorithm based on multifeature data clustering optimization model. IEEE Transactions on Industrial Informatics. IEEE. **16**(3), 2063–2071 (2019). https://doi.org/10.1109/TII.2019.2946791
10. Khraisat A, Gondal I, Vamplew P, et al. 2019. A novel ensemble of hybrid intrusion detection system for detecting internet of things attacks. Electronics. MDPI, 2019, **8**(11): 1210. https://doi.org/10.3390/electronics8111210
11. Huang, D., Shi, X., Zhang, W.A.: False data injection attack detection for industrial control systems based on both time-and frequency-domain analysis of sensor data. IEEE Int Things J. IEEE. **8**(1), 585–595 (2021). https://doi.org/10.1109/JIOT.2020.3007155
12. Abdel-Basset, M., Chang, V., Hawash, H., et al.: Deep-IFS: intrusion detection approach for industrial internet of things traffic in fog environment. IEEE Transactions on Industrial Informatics. IEEE. **17**(11), 7704–7715 (2020). https://doi.org/10.1109/TII.2020.3025755
13. Kim B J, Kim I K. 2005. Machine learning approach to realtime intrusion detection system. In Proceedings of the Australasian Joint Conference on Artificial Intelligence. Springer, Berlin, Heidelberg: 153-163. https://doi.org/10.1007/11589990_18
14. Wang C, Wang B, Liu H, et al. 2020. Anomaly detection for industrial control system based on autoencoder neural network. Wireless Communications and Mobile Computing. Hindawi. https://doi.org/10.1155/2020/8897926
15. Awotunde J B, Chakraborty C, Adeniyi A E. 2021. Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wireless communications and mobile computing. Hindawi, 2021. https://doi.org/10.1155/2021/7154587
16. Dutta V, Choraś M, Pawlicki M, et al. 2020. A deep learning ensemble for network anomaly and cyber-attack detection. Sensors. MDPI, **20**(16): 4583. https://doi.org/10.3390/s20164583
17. Jahromi, A.N., Karimipour, H., Dehghantanha, A., Choo, K.-K.R.: Toward Detection and Attribution of Cyber-Attacks in IoT-Enabled Cyber–Physical Systems. IEEE Internet of Things Journal. IEEE. **8**(17), 13712–13722 (2021). https://doi.org/10.1109/JIOT.2021.3067667
18. Yang T, Hu Y, Li Y, et al. 2019. A Standardized ICS Network Data Processing Flow With Generative Model in Anomaly Detection. IEEE Access. IEEE, **2019**, 8: 4255-4264. https://doi.org/10.1109/ACCESS.2019.2963144

19. Basati, A., Faghih, M.M.: DFE: efficient IoT network intrusion detection using deep feature extraction. Neural Comput Appl. Springer. 1–21 (2022). https://doi.org/10.1007/s00521-021-06826-6
20. Zhang H, Zhu S, Ma X, et al. 2017. A novel RNN-GBRBM based feature decoder for anomaly detection technology in industrial control network. IEICE TRANSACTIONS on Information and Systems. IEICE, E100.D(8): 1780-1789. https://doi.org/10.1587/transinf.2016ICP0005
21. Lo W W, Layeghy S, Sarhan M, et al. 2022. E-graphsage: A graph neural network based intrusion detection system. In Proceedings of the NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium. IEEE, Budapest, Hungary, 1-9. https://doi.org/10.1109/NOMS54207.2022.9789878
22. Rosay, A., Riou, K., Carlier, F., et al.: Multi-layer perceptron for network intrusion detection. Annals of Telecommunications. Springer. **77**(5), 371–394 (2021). https://doi.org/10.1007/s12243-021-00852-0
23. Yang, X., Yu, Q., He, L., et al.: The one-against-all partition based binary tree support vector machine algorithms for multi-class classification. Neurocomput. Sci. **113**, 1–7 (2013). https://doi.org/10.1016/j.neucom.2012.12.048
24. Sahoo, S.K., Makur, A.: Dictionary training for sparse representation as generalization of k-means clustering. IEEE Signal Processing Letters. IEEE. **20**(6), 587–590 (2013). https://doi.org/10.1109/LSP.2013.2258912
25. Khammassi, C., Krichen, S.: A NSGA2-LR wrapper approach for feature selection in network intrusion detection. Computer Networks. ELSEVIER. **172**, 107183 (2020). https://doi.org/10.1016/j.comnet.2020.107183
26. Wang, Y., Wang, X., Ariffin, M.M., et al.: Attack detection analysis in software-defined networks using various machine learning method[J]. Comp Electrical Eng. **108**, 108655 (2023)
27. Rababah M, Maydanchi M, Pouya S, et al. Data Visualization of Traffic Violations in Maryland, US[J]. arXiv preprint arXiv:2208.10543, 2022.
28. Abedi M, Tan X, Klausner J F, et al. A comparison of the performance of a data-driven surrogate model of a dehumidifier with mathematical model of humidification-dehumidification system[C]//AIAA SCITECH 2023 Forum. 2023: 2329.
29. Malmir, M., Momeni, H., Ramezani, A.: Controlling megawatt class WECS by ANFIS network trained with modified genetic algorithm[C]//2019 27th Iranian Conference on Electrical Engineering (ICEE). IEEE. 939–943 (2019)
30. Aghakhani, S., Larijani, A., Sadeghi, F., et al.: A Novel Hybrid Artificial Bee Colony-Based Deep Convolutional Neural Network to Improve the Detection Performance of Backscatter Communication Systems[J]. Electronics. **12**(10), 2263 (2023)
31. Liu, X., Zhang, X., Guizani, N., et al.: TLTD: a testing framework for learning-based IoT traffic detection systems[J]. Sensors. **18**(8), 2630 (2018)
32. Liu, X., Hu, T., Ding, K., et al.: A black-box attack on neural networks based on swarm evolutionary algorithm[C]// Information Security and Privacy: 25th Australasian Conference, ACISP 2020, Perth, WA, Australia, November 30–December 2, 2020, Proceedings 25. Springer International Publishing. 268–284 (2020)
33. Liu, X., Zhang, X., Zhu, Q.: Enhanced fireworks algorithm for dynamic deployment of wireless sensor networks[C]//2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST). IEEE. 161–165 (2017)
34. Lu, J.Z., Chen, K., Zhuo, Z.L., et al.: A temporal correlation and traffic analysis approach for APT attacks detection. Cluster Computing. Springer. **22**(3), 7347–7358 (2017). https://doi.org/10.1007/s10586-017-1256-y
35. Lu, J.Z., Lv, F.M., Zhang, X.S.: Integrating Traffics with Network Device Logs for Anomaly Detection. Security and Communication Networks. Hindawi. **2019**, 5695021 (2019). https://doi.org/10.1155/2019/5695021
36. Lu, J., Zhang, W., Deng, Z., et al.: Research on information steganography based on network data stream[J]. Neural Computing and Applications. **33**, 851–866 (2021)
37. Lan, J., Lu, J.Z., Wan, G.G., et al.: E-minBatch GraphSAGE: An Industrial Internet Attack Detection Model[J]. Security and Communication Networks. **2022**, (2022)