Hindawi
Security and Communication Networks
Volume 2022, Article ID 5363764, 12 pages
https://doi.org/10.1155/2022/5363764

Research Article

E-minBatch GraphSAGE: An Industrial Internet Attack Detection Model

Jin Lan, Jia Z. Lu, Guo G. Wan, Yuan Y. Wang, Chen Y. Huang, Shi B. Zhang, Yu Y. Huang, and Jin N. Ma

School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China

Correspondence should be addressed to Jia Z. Lu; ljz@cuit.edu.cn

Received 3 March 2022; Revised 30 May 2022; Accepted 16 June 2022; Published 14 July 2022

Academic Editor: Robertas Damaševičius

The Industrial Internet has grown rapidly in recent years, and attacks against the Industrial Internet have also increased. When compared with the traditional Internet, the industrial Internet has a more complex network structure, and the traditional graph neural network attack behavior detection model cannot well adapt to the complex network environment. To make the model better adapt to the complex network environment, this paper proposes the E-minBatch GraphSAG model. First, the application layer source port and source IP address is used as source nodes, the application layer target port and target IP address are used as target nodes, and the remaining traffic information is used as edge information to complete the construction of the graph structure data, and then the constructed graph structure data is presampled to select the edge information that needs to be aggregated next, followed by using the AGG aggregation function to aggregate the information in the domain generated by the presampling process. Finally, the information of two adjacent nodes is aggregated as edge information to classify the edges. Increase the number of IP addresses in the UNSW-NB15 dataset, and then use it for model training and testing. The experimental results show that the accuracy of the model reaches 99.49% in a relatively complex network environment. In this paper, the E-minBatch GraphSAG model is presented in an attempt to solve the problem of attack detection in the complex industrial Internet environment.

1. Introduction

Because the traditional industrial production network is separated from the Internet, and the traditional industrial control protocol does not take into account the security events that may occur during use, most traditional industrial control protocols have security problems [1]. With the rapid development of Internet technology, more and more Internet technologies are used in industrial production processes to achieve the goal of automating industrial production processes and reducing production costs [2], resulting in a new concept-industrial Internet. There is also a big difference between the modern industrial Internet and the traditional Internet. The main difference between the Industrial Internet and the traditional Internet is that the traditional Internet has a close connection with people, while the Industrial Internet has a close connection with things. The architecture of the Industrial Internet is also quite different from the traditional Internet architecture. In the modern Industrial Internet, the enterprise management, the supervisory layer, and the field layer are the main components [3].

The social impact of industrial Internet security incidents is far greater than the social impact of traditional Internet security incidents. In recent years, attacks on the Industrial Internet have gradually increased. Iran's Natanz nuclear enrichment site was attacked by the Stuxnet computer virus in 2010, causing abnormal acceleration of uranium enrichment centrifuges and eventually leading to their destruction [4]. It also opened the curtain for attacks against the industrial Internet. In 2015, the malware BlackEnergy3 [5] hacked into the control center of the Ukrainian power grid and tampered with the control commands of the relays via VPN causing widespread power outages in Ukraine. BlackEnergy3 compromised the network and software of the grid control system, launching a DDoS attack that prevented

the control system from sensing abnormal system conditions, thus preventing power from being restored to the blackout area for a long time. During Black Hat 2017, Dr. Staggsp [6] demonstrated how to hack into a wind farm's control system by physically connecting to an uncontrolled wind turbine in the United States. In 2021, a state of emergency was declared in the United States after the hacker group "DarkSide" attacked the largest fuel pipeline operator in the country [7]. Several security incidents have shown that the industrial Internet faces huge security risks, and artificial intelligence-based attack detection systems can help to provide early warning of attacks and greatly improve the security of the system.

Detecting attacks is a key step in securing the industrial Internet, and alerting to attacks as early as possible can reduce the impact of attacks to a manageable extent. Currently, there are different classification results for different intrusion detection systems based on the classification method [8]. There are two types of data source classification: host-based and network-based. Classification based on the detection technique can be classified as misuse-based approach and anomaly-based approach. Anomaly-based methods, which are currently the mainstream detection methods, can also be classified as statistical analysis-based methods [9], cluster analysis-based methods [10], artificial neural network-based methods [11], or deep learning-based methods [12]. Among them, current studies generally agree that deep learning-based methods for attack detection are more effective than the others [13]. This is because deep learning-based models have better self-learning, self-adaptive capabilities, better generalization ability, and the ability to detect unknown attack behaviors better.

Most attack behavior detection methods focus on finding attack behaviors from the attack traffic itself, while ignoring the correlation between attack traffic. In this paper, we attempt to introduce graph neural networks, a relatively new subfield in the field of deep neural network research, into attack behavior detection in the Industrial Internet.

The scale of the Industrial Internet has begun to grow explosively, and the network structure has become increasingly complex. In order to detect attacks in a complex network environment, this paper proposes an improved method based on E-GraphSAGE algorithm [14], E-GraphSAGE is a variant of GraphSAGE [15], which allows to collect graph edge information and support edge features Perform edge classification to detect malicious network flows. In this paper, the E-minBatch GraphSAGE algorithm is proposed to be able to better adapt to the complex network environments.

The contributions of this paper are mainly as follows.

(i) In this paper, we propose a new GNN model based on E-GraphSAGE, which uses information such as traffic duration and packet size as edge features of the graph, and presamples the points in the graph structure data so that the model can better adapt to the complex network environment.

(ii) This paper applies the new proposed model to industrial Internet attack detection and demonstrates the superiority of the new model by comparing it with traditional machine learning algorithms and deep learning algorithms through experiments.

(iii) The E-minBatch GraphSAGE algorithm proposed in this paper has better results in the detection of three kinds of attacks, namely Shellcode, Reconnaissance, and Exploits.

The rest of this paper is organized as follows. Section 2 discusses related work on industrial Internet attack behavior detection, Section 3 briefly introduces the basics related to GNN and GraphSAGE, Section 4 presents our new GNN model based on GraphSAGE, Section 5 gives experimental results and analysis, and Section 6 summarizes the full paper.

2. Related Works

At present, traditional machine learning or deep learning is mainly used for industrial Internet attack behavior detection. In contrast, there are relatively few researches on attack behavior detection based on graph neural network.

2.1. Traditional Industrial Internet Attack Behavior Detection Algorithms. The label-based attack behavior detection system can accurately detect the known attack behavior, but it is powerless to detect the unknown attack behavior. At the same time, the anomaly-based attack behavior detection system can effectively detect unknown attack behaviors, but an unavoidable problem is: no matter whether the attack behavior is known or not, the anomaly-based attack behavior detection system will have a large false negative rate and false positive rate. In order to enable the model to detect both unknown attack behaviors and known attack behaviors, researchers began to try to combine the two attack behavior detection systems. Khraisat et al. [16] combined the C5 classifier and a class of support vector machine classifiers to design a hybrid intrusion detection system (HIDS) that integrated the advantages of the label-based attack behavior detection system and the anomaly-based attack behavior detection system. The experimental results show that the method has a high accuracy in detecting attack data on the Bot-IoT dataset.

The traffic of attack behavior of the Industrial Internet presents the characteristics of low frequency and multistage. Li et al. [17] designed a bidirectional long-term and short-term storage network with multiple features, and the sequence feature layer and stage feature layer were introduced into the model. The model in the training phase can learn the corresponding attack range from historical data, and effectively detect attacks in different ranges. Suzen et al. [18] proposed a hybrid Deep Belief Network (DBN) attack behavior detection model. Hidden layers are updated via Contrastive Divergence (CD). Experiments show that the hybrid deep belief network model has achieved good accuracy in the detection of industrial Internet attack behavior. A multifeatured data clustering optimization model was used by Liang et al. [19] as the basis of an industrial network intrusion detection algorithm, which classifies the weighted distance and safety factor of the data according to the priority thresholds of the data attribute features of the nodes

in the data. Cluster centers are selected by choosing a node with a high safety factor, and data from around the node is matched into a cluster. In comparison with other algorithms, the experimental results demonstrate that the proposed algorithm has significant advantages in terms of detection rate and processing time. Huang et al. [20] proposed a data-driven intrusion detection method based on time-domain and frequency-domain analysis. The proposed method uses closed-loop controlled sensors, does not consume additional system resources and relies on system models, extracts time-domain and frequency-domain features, uses feature vectors under normal working conditions to build a hidden Markov model, and converts the trained hidden Markov model.

The traffic in the Industrial Internet is very complex and includes not only production networks but also other office networks. About solving the problem of massive data attack behavior detection in hybrid networks, Zhang et al. [21] proposed a data mining algorithm for massive intrusion cluster computing in hybrid networks with feature extraction under specific constraints. Multicomponent cross-detection methods are used to collect information on mixed network massive intrusions and construct models of mixed network massive intrusion signals. Regarding the intrusion interference under the constraint of fixed time-frequency window, Zhang adopts the cascade trap method to deal with it, so as to extract the localized basic volume and main function from a large amount of interference information, and obtain the complete energy distribution spectrum on the time-frequency plane. Data mining for clustering calculations with massive intrusion interference constraints is achieved with the help of the energy distribution spectrum as a guiding function.

The rapid development of the Industrial Internet has led to IoT devices widely deployed, and at the same time, attacks against IoT devices have also appeared in large numbers. IoT devices are ideal springboards for DoS attacks—low security and large numbers make IoT devices the target of many botnets. The attack behavior detection system needs to identify the nodes attacked by DoS in time, and takes measures such as isolation of the infected nodes to ensure the security of the entire industrial Internet environment. Alharbi et al. [22] proposed a Local Global Optimal Bat Algorithm (LGBA-NN) for Neural Networks to select feature subsets and hyperparameters to effectively detect botnet attacks. Experimental results show that LGBA-NN outperforms other variants in detection of multiple botnet attacks. Ali et al. [23] trained on intrusion data, features, and suspicious activity datasets. The data is trained according to different layers of the long- and short-term network to improve the accuracy of attack detection. With the help of training information, the test details are classified by extracting features and forming a sparse matrix construction. In experiments, the model's accuracy reached 99.29%.

The computing power of industrial Internet nodes is relatively poor, and the resources required for the training and deployment of attack detection models are huge. About how to reduce the resources consumed by the deployment node, Wozniak et al. [24] used RNN-LSTM classifier and NAdam optimization algorithm to build the model. Experimental results indicate that the model requires very few resources on deployment nodes.

All the above algorithms have achieved desirable performance in industrial Internet attack detection, but they all only consider the characteristics of the traffic itself or the spatial characteristics of the traffic, and do not consider the correlation between the traffic.

2.2. Industrial Internet Attack Behavior Detection Algorithm Based on Graph Neural Network. Graph neural networks are developing rapidly, and good progress has been made in their applications in many fields. However, the application of graph neural network to network attack behavior detection is still a relatively new field and deserves further research.

Lo et al. [14] proposed a model named E-GraphSAGE based on the GraphSAGE model, which supports edge classification. Taking IP addresses and application-layer ports as nodes, the data flows communicated between hosts are treated as side information, thereby classifying network flows into benign flows and attack flows. According to the experimental comparison, the model proposed by the author is generally better than the traditional attack behavior detection model. However, experiments have shown that with the increase of network complexity, the accuracy of E-GraphSAGE begins to decrease. Our method proposes an improved model based on E-GraphSAGE, which can better adapt to complex network environments.

3. Background

3.1. Industrial Internet Infrastructure. The industrial Internet attack behavior detection model is one of the methods to protect the safety of industrial production equipment and personnel. There are mainly three layers in modern industrial Internet architecture: the enterprise management layer, the supervision layer and the field layer. The enterprise management layer relies on the Internet to enable real-time monitoring and management of industrial processes and assist enterprises in making informed decisions. In addition to collecting data and transmitting it between the enterprise management layer and the field layer, the monitoring layer controls the field devices with specific logic. In the field layer, field information is perceived by the field devices, and data is exchanged between field devices via the field bus. The modern Industrial Internet architecture is shown in Figure 1.

As shown in Figure 1, industrial Internet attack behavior detection systems are generally deployed between the management and the management level of an enterprise, and between the management and the field level control level [3]. There are various attack behavior detection systems, and this paper focuses on GraphSAGE algorithm based on graph neural network.

3.2. Graphical Neural Network. Different attack detection algorithms require different input structures. The input data structure of the CNN-based attack behavior detection algorithm is the grayscale graph corresponding to the traffic. The input data structure of GNN-based attack behavior

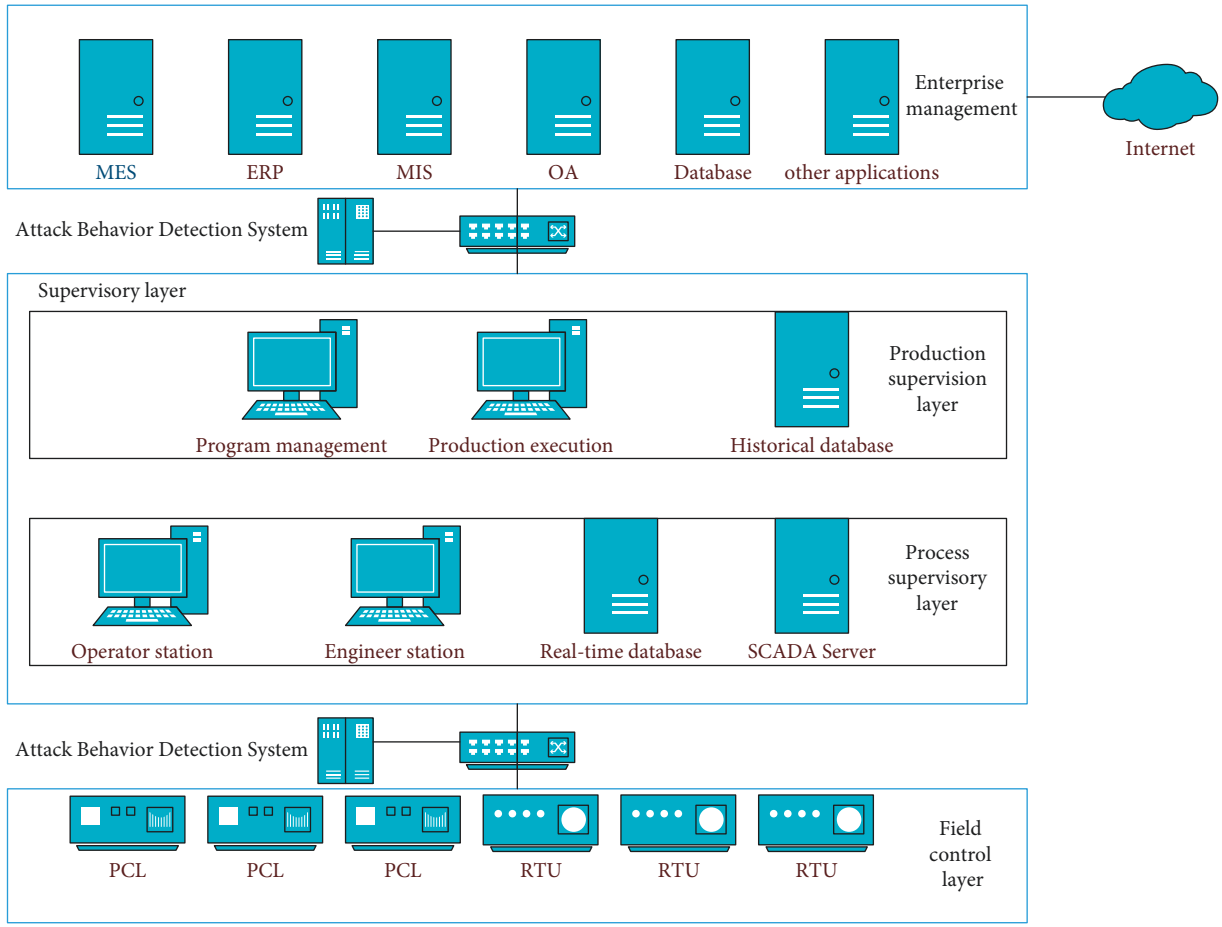

Figure 1: Industrial internet infrastructure.

detection algorithm is the IP address and application layer port as nodes, and the data flow of communication between hosts is treated as edge information, as shown in Figure 2.

Because graph neural networks can utilize data with graphical structure encountered in real-world applications (biology, telecommunications, chemistry, etc.), graph neural networks have received widespread attention since their introduction, and they have grown rapidly in recent years to become one of the fastest growing subfields of artificial intelligence.

The main reason for using GNNs for industrial Internet attack detection is that GNNs can easily exploit important structural information in network data streams. The information in network data streams can be directly encoded into a graphical format. In fact, converting network data traffic into graphical format is a method that has been used earlier, but the process is usually tedious and heavily dependent on manual labor.

3.3. GraphSAGE. GNNs can be considered as a generalization of convolutional neural networks to non-Euclidean data structures [25]. Graph neural networks use the concept of message passing to implement a generalization of the capabilities of convolutional neural networks to the processing of data with non-Euclidean structures. The messages

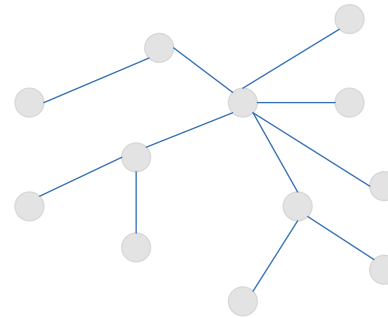

Figure 2: Figure structure.

received by a node are the result of the properties (or attributes) of the neighboring nodes of that node being aggregated. Iteration of the above process is repeated to pass the information from one node to the whole network. If in each iteration, an attempt is made to aggregate all neighboring nodes, unpredictable memory consumption and computational resource requirements occur.

Figure 3(a) shows a simple graph structure data and Figure 3(b) shows two GraphSAGE message passes to the graph. In this example, we assume that the nodes sample all neighboring nodes, i.e., information from all domain nodes

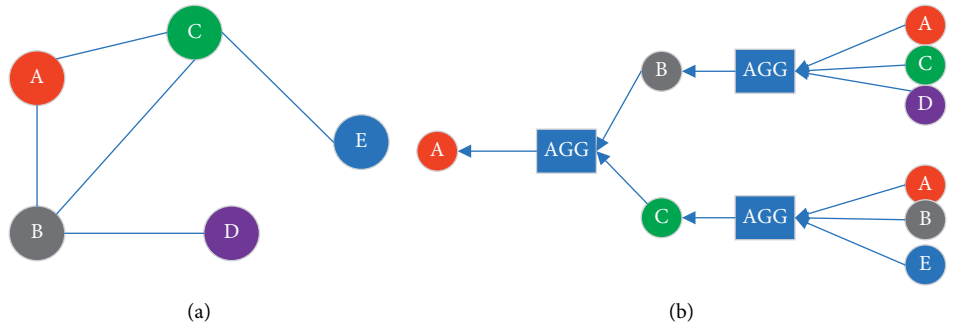

Figure 3: A given graph structure data and the corresponding two-layer fully sampled GraphSAGE algorithm model.

is considered in each iteration. In the face of more complex graph structures, sampling all nodes makes the training time and effectiveness of the model not optimized, so an attempt is made to presample the nodes [26].

Some of the symbols in the graph neural network are defined as follows: $G(v, \varepsilon)$ denotes the data of a graph structure, v is the set of points, and ε is the set of edges. The feature vector of node v is denoted as a vector X_v, and the full set of node feature vectors can be denoted as $\{X_v, \forall v \in v\}$.

In the GraphSAGE algorithm, one of the most critical hyperparameters is the number of convolutional layers K. The role of this hyperparameter is to specify the information of the algorithm's aggregated K-layer neighbor nodes. Considering both the experimental effect and the model complexity, we generally set the number of layers to $K = 2$ in the actual experimental process [26]. On the other hand, GraphSAGE needs to choose a differentiable aggregator function that aggregates the information from the neighboring nodes.

The GraphSAGE algorithm has been used in many fields with good results. However, the algorithm focuses on node classification and does not consider the problem of edge classification. The E-GraphSAGE algorithm proposed by Lo successfully solves the problem of edge classification, but cannot solve the problem of classification in complex network environment architectures. Based on the E-GraphSAGE algorithm, a new node presampling algorithm is proposed to enable the model to better detect attack behaviors in complex networks.

4. E-minBatch GraphSAGE

E-minBatch GraphSAGE is presented in this section, along with its application to detecting industrial Internet attacks.

4.1. E-minBatch GraphSAGE

4.1.1. Forward Propagation Stage. The E-GraphSAGE algorithm, compared with the traditional GraphSAGE, considers not only the node features but also the edge features, while E-GraphSAGE proposes edge embedding. The nodes are presampled in advance so that the E-minBatch GraphSAGE algorithm can adapt to complex network structures, as shown in Algorithm 1.

In comparison to E-GraphSAGE, the algorithm presented in this paper has a larger number of input nodes, which can better represent the complex network environment, and in the face of complex network structure this paper presamples the nodes once to improve the ability of attack behavior detection model to detect attack behavior in complex network environment. As shown in line 1 to 5 of the algorithm, we determine whether a node is a neighbor node of the current node, and if it is, it is directly added to the sampling range. graphSAGE recommends the use of two layers of convolution for the model, and the product of the number of neighbor nodes sampled twice is not greater than 500. The number of samples sampled twice for the model used in this paper is $S1 = 20$, $S2 = 25$(Note: $s1$ indicates that the first layer samples 20 neighbor nodes, and $s2$ indicates that the second layer samples 25 neighbor nodes). As with the E-GrapghSAGE algorithm, this paper still uses the $x_v = (1, ..., 1)$ initialized node features to aggregate the domain edges at the Kth layer.

In the aggregation function in line 9, the difference between E-minBatch GraphSAGE and GrapghSAGE algorithm is that the aggregation is not the information of surrounding adjacent nodes, but the aggregation of surrounding edge information.

$$h^k_{N(v)} = AGG_k\left\{\left\{h^{k-1}_{uv}, \forall u \in N(v), uv \in \varepsilon\right\}\right\}, \tag{1}$$

h^{k-1}_{uv} denotes $N(v)$ the edges in the sampled domain of node u in the k-1 layer and uv denotes the edge $\{\forall u \in N(v), uv \in \varepsilon\}$, $N(v)$ in the sampled domain of node v.

The calculation process in line 10 is the same as the traditional GrapghSAGE algorithm, but the calculation includes the edge information of the previous layer.

Line 11 calculates the node embedding of the kth layer, and the edge embedding Z_{uv} of the nodes in the last layer is the splicing Z_u with Z_v the node embedding, as shown in the following equation:

$$Z_{uv} = \text{CONCAT}\left(Z^K_u, Z^K_v\right), uv \in \varepsilon. \tag{2}$$

4.1.2. Back Propagation. In the back propagation phase, the method used in this paper is updated in the same way as the traditional GraphSAGE algorithm.

Input: Graph $G(v, \varepsilon)$; input edge features $\{e_{uv}, \forall_{uv} \in \varepsilon\}$; input node features $x_v = \{1, .., 1\}$, $x_v \in B$; depth K; weight matrices $W^k, \forall \in \{1, .., K\}$; non-linearity σ; differentiable aggregator functions AGG_K;
Output:Edge emdeddings $Z_{uv}, \forall_{uv} \in \varepsilon$

(1) $B^K \leftarrow B$
(2) **for** $k = K \ldots 1$ do
(3) $\quad B^{k-1} \leftarrow B^k$;
(4) $\quad$ **for** $u \in B^k$ do
(5) $\quad\quad B^{k-1} \leftarrow B^{k-1} \cup N_k(u)$;
(6) $\quad$ **end for**
(7) **end for**
(8) $h_v^0 = x_v, \forall_v \in V$
(9) **for** $k \leftarrow 1$to$K$ do
(10) $\quad$ **for** $u \in B^k$ do
(11) $\quad\quad h_N^k(v) \leftarrow \mathrm{AGG}_k(\{h_u v^k - 1, \forall u \in N(v), uv \in \varepsilon\})$
(12) $\quad\quad h_v^k \leftarrow \sigma(W^k.\mathrm{CONCAT}(h_v^k - 1, h_N(v)^k))$
(13) $\quad$ **end for**
(14) **end for**
(15) $Z_v = h_v^K$
(16) **for** $k \leftarrow 1$to$K$**do**
(17) $\quad z_{u_v^k} \leftarrow \mathrm{CONCAT}(z_u^K, z_v^K)$
(18) **end for**
(19) $z_{uv=} z_{u_v^k}$# k represents the last layer of the model#

Algorithm 1: E-minBatch GraphSAGE edge embedding.

4.2. E-minBatch GraphSAGE Attack Detection Model. As shown in Figure 4, the E-minBatch GraphSAGE attack detection model proposed in this paper first generates a network graph using network stream data, and then presamples the nodes once. After completing the presampling, the data is fed into the model for training. Finally, edge embeddings are created and classification operations are performed on the edges. The next steps are described in turn.

4.2.1. Network Diagram Construction. Network data streams are the fundamental form of data transmission in today's industrial Internet. It is also the most commonly used data format for attack detection models. The data stream contains not only the source and target of the data information, but also the size, duration, and other information of the data stream. In some scenarios, the flow is presented in the form of a graph.

There are different options for using graphs to represent data flows in different usage scenarios. In this paper, the source IP address and application layer port are used to identify the source node, and the target IP and target application layer port are used to identify the target node. The rest of the information is used as information about the edges between the source and target nodes.

Make training data and test data better represent complex network structures, and the original source IP addresses are mapped to random addresses in the range of 10.0.0.0–10.255.255.255 in this paper. A large number of IP addresses can represent the complex network more accurately and make the trained model better adapted to the complex network.

4.2.2. Presampling. In order to adapt to complex network structures, the nodes in the graph continue to be presampled after the conversion of the traffic to graph structure type is completed. In this paper, we use a two-layer convolution process, so each node is presampled twice, the first layer presamples the 20 neighbor nodes of the current node, and the second layer presamples the 25 neighbor nodes of the current node. When the number of neighboring nodes of a node cannot meet the presampling requirement, some of the neighboring nodes are sampled again.

4.2.3. Model Training. The training of the GraphSAGE model generally samples two layers of convolution [27], and similarly the E-minBatch GraphSAGE proposed in this paper uses two layers of convolution. For the aggregation function AGG, the mean value of each edge embedding is simply found, and the defined form is shown in the following equation:

$$h_{N(v)}^k = \sum_{\substack{u \in N(v) \\ uv \in \varepsilon}} \frac{h_{uv}^{k-1}}{|N(v)|_e}. \tag{3}$$

h_{uv}^{k-1} denotes the embedding of the model at layer k-1 and $|N(v)|_e$ denotes the number of aggregated neighbor nodes. In the two-layer convolution, the number of sampling neighbor nodes is $S1 = 20$, $S2 = 25$.

The size of the hidden layer as shown in (3) is set to 128 hidden units, and the nonlinear activation function is chosen as the ReLu function. For improving the model's ability to generalize, a dropout mechanism of 0.2 is set between the

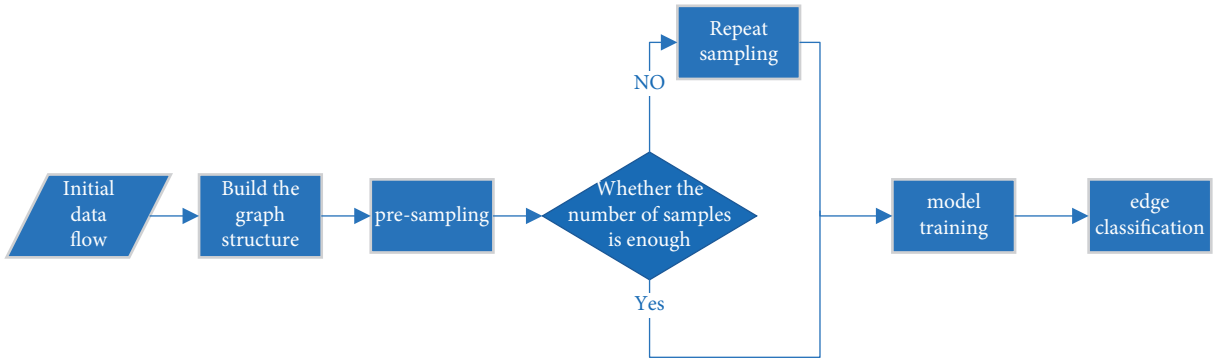

Figure 4: E-minBatch graph SAGE attack behavior detection model flow.

two convolutional layers. When generating the calculation results in the last layer, the embedding of the two nodes is spliced together to get the corresponding edge embedding, and the size of the edge embedding is 256-dimensional at this time, and the edge embedding is passed through a Log Softmax layer, which facilitates the training and optimization of the model parameters.

4.2.4. Edge Classification. After the model training is completed, the effectiveness of the E-minBatch GraphSAGE model is evaluated using the test set. The test set also needs to be transformed into a graph structure as well, presampled, passed through the trained E-minBatch GraphSAGE layer, and finally passed through the Log Softmax layer edge corresponding to the probabilities of different classes, and finally compared with the real class labels to calculate the classification evaluation performance metrics.

5. Dataset and Experimental Results

In this section, this paper presents the datasets selected for training and testing along with the evaluation criteria of the experiments, and finally the experimental results of the model.

5.1. Dataset. The model was pretrained with the UNSW-NB15 [28] dataset, which was generated by the IXIA PerfectStorm tool from the Australian Cyber Security Centre (ACCS) Cyber Scope Lab. The number of various types of traffic included in UNSW-NB15 is shown in Table 1.

5.2. Evaluation Criteria. In this paper, the parameters shown in Table 2 are used to evaluate the selected model and the model proposed in this paper.

In the experiments, two labels are defined for UNSW-NB15, one indicating whether the traffic is attack traffic, and if it is, the other label what kind of attack traffic the traffic is. The first label is used for dichotomous classification and the second label is used for multiclassification. In our experiments, 70% of the traffic data of the UNSW-NB15 dataset is used as the training set, and 30% of the traffic data is used as the test set.

5.3. Experimental Results. Firstly, we compare the accuracy of different models under different training times, as shown in Figure 5.

As we can see from Figure 5, the convergence speed of the graph neural network algorithm is much slower compared to the speed of other traditional neural networks. Because in graph neural networks, along with the increase in the number of network layers, information from more distant nodes needs to be aggregated, which is the reason why using the GraphSAGE algorithm suggests setting the model within two layers. The reason for the slower convergence speed of the E-minBatch GraphSAGE algorithm compared to the E-GraphSAGE algorithm is that the nodes are presampled and require more training times to aggregate the information of surrounding neighboring nodes.

The models E-GraphSAGE [14], CNN [29], RF [27], ResNet50 [30], and the model proposed in this paper are compared in terms of F1-score, ACC, Precision, and Recall.

In a complex network environment, the model proposed in this paper, as shown in Figure 6(a), *F*1score reaches 99.88%, as shown in Figure 6(b), ACC reaches 99.49%, as shown in Figure 6(c), Precision reaches 99.67%, as shown in Figure 6(d), and Recall reaches 99.74%, which is better than E-GraphSAGE. At the same time, the model proposed in this paper is slightly inferior to the current state-of-the-art deep learning model in terms of *F*1-score, ACC, Precision, and Recall, but is currently based on graph neural networks. The research on the network attack behavior detection algorithm is still in the initial stage, and there is room for further research in the future. When the E-GraphSAGE algorithm is used to detect attack behaviors, it not only considers the characteristics of the traffic itself, but also considers the correlation between the traffic. Therefore, in a complex network environment, the effect of the E-GraphSAGE algorithm will decline to a certain extent. The purpose of E-minBatch GraphSAGE proposed in this paper is to make the attack behavior detection method based on graph neural network still has good performance in complex network environment. In the following comparative experiments, the E-minBatch GraphSAGE algorithm proposed by us and the E-GraphSAGE algorithm proposed by Lo are compared.

Table 1: UNSW-NB15 flow type, quantity and profile.

Flow type	Quantity	Introduction
Normal	2,218,761	Normal data traffic
Fuzzers	24,246	Send randomly generated fuzzy data to the target to cause the target to error into a pause state
Analysis	2,677	Port scanning, spam, and html file infiltration
Exploits	44,525	Attacks that exploit vulnerabilities known to exist in the system or software
Worms	174	Attack initiators such as viruses replicate themselves and try to infect other hosts on the network
Shellcode	1,511	A piece of code that exploits a software vulnerability
DoS	16,353	Launch a flooding attack on the target so that it cannot accept new requests
Generic	215,481	Attack against any type of group password
Reconnaissance	13,987	Simulation of information-gathering attacks
Backdoor	2,329	Bypass system defense mechanisms to access sensitive locations and sensitive information

Table 2: Model performance metrics.

Metric	Definition
Recall	$TP/TP + FN$
Precision	$TP/TP + FP$
$F1$-score	$2 \times \text{Recall} \times \text{Precision}/\text{Recall} + \text{Precision}$
Accuracy	$TP + TN/TP + FP + TN + FN$

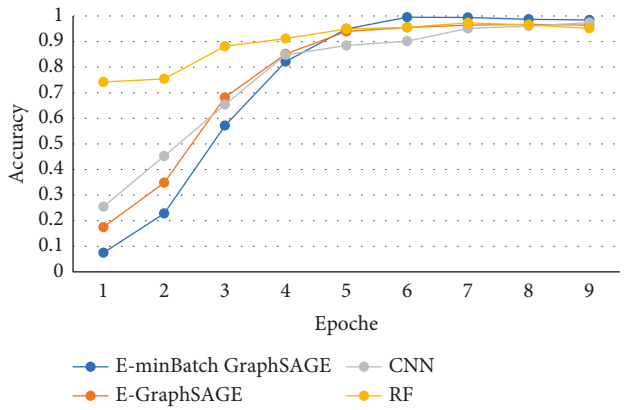

Figure 5: 10 epoche training accuracy.

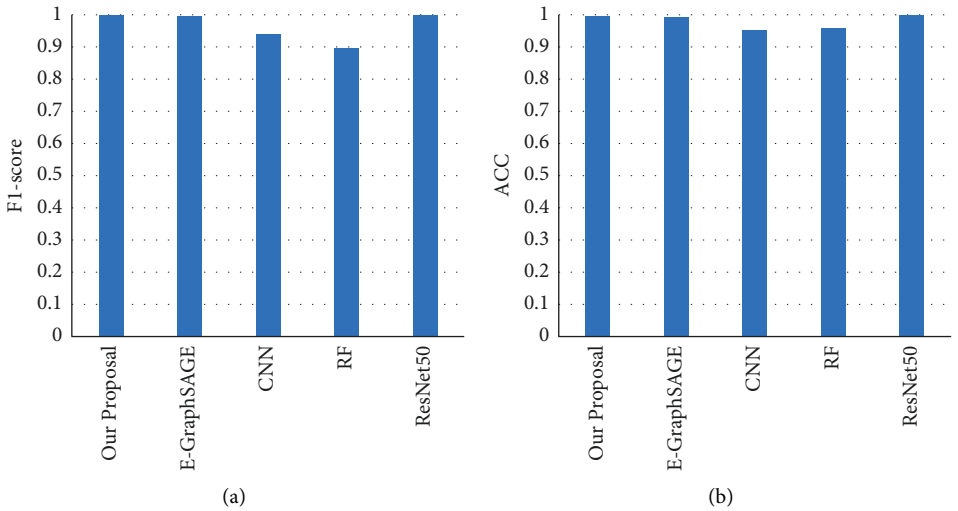

Figure 6: Continued.

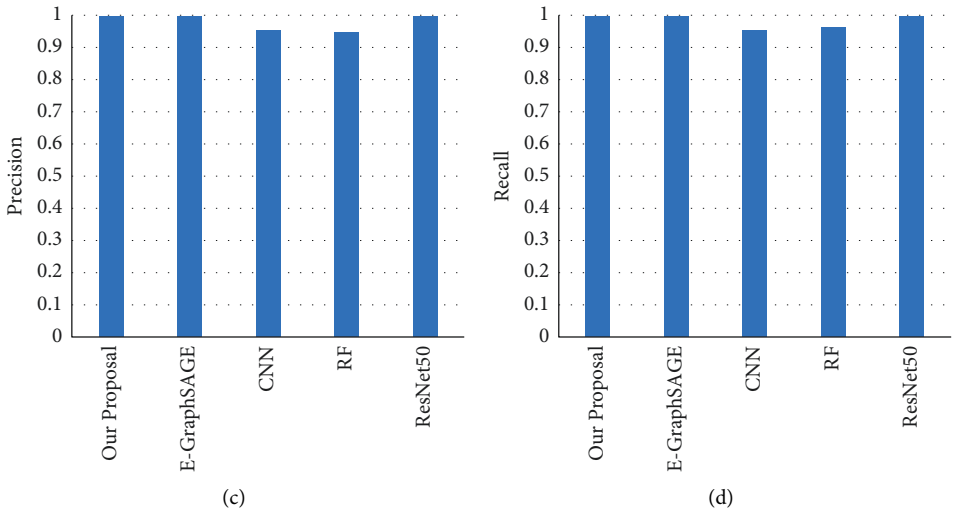

Figure 6: Compare the model proposed in this paper with E-graph SAGE, CNN, RF, ResNet50 in ACC, *F*1-score, precision, and recall. (a) *F*1-score comparison chart. (b) ACC comparison chart. (c) Precision comparison chart. (d) Recall comparison chart.

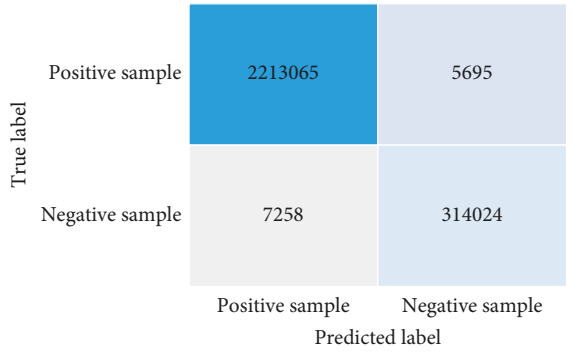

Figure 7: Compare the base confusion matrix for UNSW-NB15 dataset.

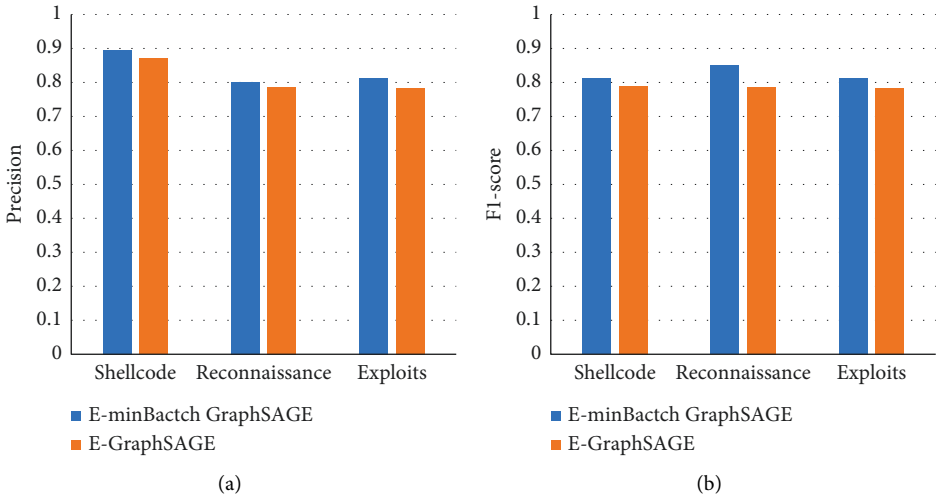

Figure 8: Continued.

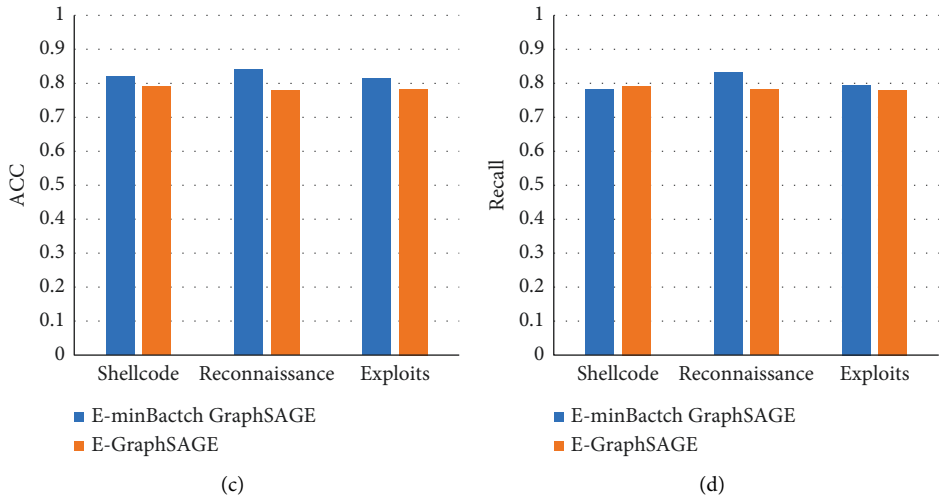

Figure 8: Compare the ACC, *F*1-score, precision, and recall of the model proposed in this paper with E-graph SAGE on three attacks: Shellcode, reconnaissance, and exploits. (a) Precision of some aggressive behaviors-1. (b) *F*1-score of some aggressive behaviors-1. (c) ACC of some aggressive behaviors-1. (d) Recall of some aggressive behaviors-1.

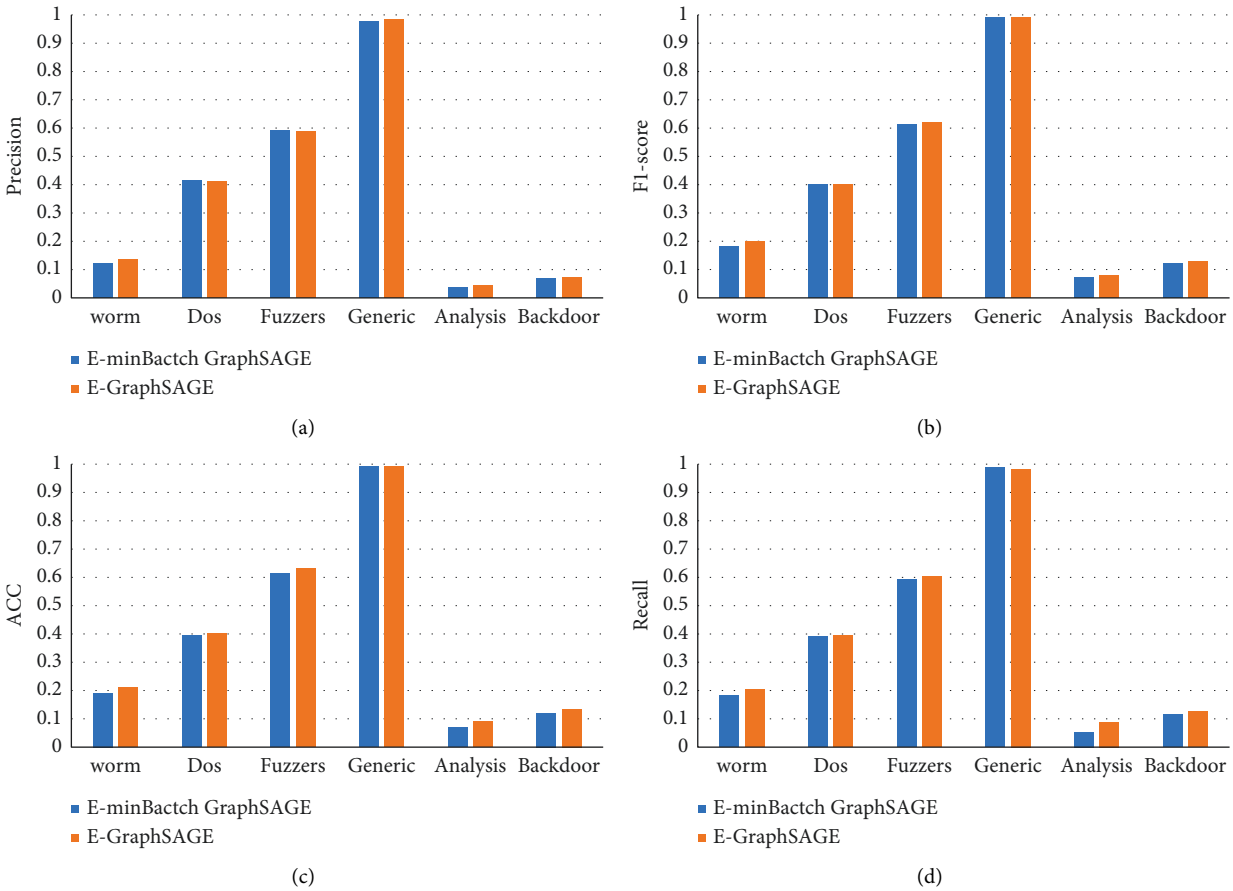

Figure 9: Compare the ACC, *F*1-score, precision, and recall of the model proposed in this paper with E-graph SAGE on three attacks: worm, dos, fuzzers, generic, analysis, backdoor. (a) Precision of some aggressive behaviors-2. (b) *F*1-score of some aggressive behaviors-2. (c) ACC of some aggressive behaviors-2. (d) Recall of some aggressive behaviors-2.

Calculate the confusion matrix to show the effect of the E-minBatch GraphSAGE model. The confusion matrix is shown in Figure 7.

The E-minBatch GraphSAGE model proposed in this paper achieves better results than the E-GraphSAGE model in the detection of three attack behaviors: Shellcode, Reconnaissance, and Exploits. As shown in Figure 8(a), the detection rate of Shellcode attack increased by 2.65%, the detection rate of Reconnaissance attack increased by 1.48%, and the detection rate of Exploits attack increased by 2.83%. At the same time, as shown in Figure 8, the model proposed in this paper still has a certain degree of improvement compared to the E-GraphSAGE model in other metrics (ACC, *F*1-score, and Recall).

To make the model better adapt to the complex network environment, when training the E-minBatch GraphSAGE model, a presampling process is performed, resulting in when the remaining attack behaviors of the UNSW-NB15 dataset are used, and the effect obtained by the model proposed in this paper is similar to that obtained by the E-GraphSAGE model, as shown in Figure 9.

6. Conclusion

This paper proposes a new algorithm-E-minBatch GraphSAGE based on E-GraphSAGE. To make the model better adapt to the complex network environment, the E-minBatch GraphSAGE algorithm presamples the neighbor edges of each node of the model after the graph structure data is constructed. In order to verify the effect of E-minBatch GraphSAGE, experiments are carried out on the UNSW-NB15 dataset. The results show that the algorithm proposed in this paper is comparable to the E-GraphSAGE algorithm in terms of attack behavior detection accuracy and *F*1-score in a complex network environment. In comparison, the model's accuracy and *F*1-score have achieved better results. Compared with the current state-of-the-art deep learning algorithms, the algorithm proposed in this paper is still insufficient in terms of accuracy. At the same time, the algorithm proposed in this paper has great problems in small sample detection, which are worthy of further study.

Data Availability

The data set can be accessed from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant no. 62102049), "Research on Intelligent Depth Detection of APT Attacks for Cyber-Physical Systems," the National Natural Science Foundation of China (no. 62076042), the Key Research and Development Project of Sichuan Province (nos. 2021YFSY0012, 2020YFG0307, and 2021YFG0332), the Science and Technology Innovation Project of Sichuan (no. 2020017), the Key Research and Development Project of Chengdu (no. 2019-YF05-02028-GX), the Innovation Team of Quantum Security Communication of Sichuan Province (no. 17TD0009), and the Academic and Technical Leaders Training Funding Support Projects of Sichuan Province (no. 2016120080102643).

References

[1] K. W. Schmidt and Schmidt, "Distributed real-time protocols for industrial control systems: framework and examples," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1856–1866, 2012.

[2] J. E. Rubio, R. Roman, and J. Lopez, "Integration of a threat traceability solution in the industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6575–6583, 2020.

[3] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 155014771879461, August 2018.

[4] R. Langner, "Stuxnet: dissecting a cyberwarfare weapon," *IEEE Secur Priv2011*, vol. 9, no. 3, pp. 49–51, 2021.

[5] R. M. Lee, M. J. Assante, and T. Conway, *Analysis of the Cyber Attack on the Ukrainian Power Grid*, p. 2, Electricity Information Sharing and Analysis Center(E-ISAC), Washington,DC, 2020.

[6] J. Staggs, *Adventures in Attacking Wind Farm Control Networks*, black hat, San Francisco, CA, 2017.

[7] E. Noonan, *Colonial Pipeline Didn't Have Multifactor Authentication in Place—And Most Defense Contractors Don't Either*Nextgov.com, China, 2021.

[8] R. Singh, H. Kumar, R. K. Singla, and R. R. Ketti, "Internet attacks and intrusion detection system," *Online Information Review*, vol. 41, no. 2, pp. 171–184, 2017.

[9] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," in *Proceedings of the 7th USENIX Security Symposium*, pp. 120–132, San Antonio,TX, January 1998.

[10] L. Khan, M. Awad, and B. M. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," *The VLDB Journal*, vol. 16, no. 4, pp. 507–521, 2007.

[11] E. Hodo, X. Bellekens, A. Hamilton et al., "Threat analysis of iot networks using artificial neural network intrusion detection system," in *Proceedings of the 2016 International Symposium on Networks Computer and Communications*, pp. 1–6, China, May 2016.

[12] A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 282, pp. 761–768, May 2017.

[13] R. Beghdad, "Critical study of neural networks in detecting intrusions," *Computers & Security*, vol. 27, no. 5-6, pp. 168–175, 2008.

[14] W. W. Lo, S. Layeghy, M. Sarhan, and E. GraphSAGE, "A Graph Neural Network Based Intrusion Detection System," 2021, https://arxiv.org/abs/2103.16329.

[15] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 02216, 2017.

[16] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "A novel ensemble of hybrid intrusion detection system for detecting Internet of things attacks," *Electronics*, vol. 8, no. 11, p. 1210, 2019.

[17] X. Li, M. Xu, P. Vijayakumar, N. Kumar, and X. Liu, "Detection of LowFrequency and Multi-Stage Attacks in Industrial Internet of Things," *IEEE Transactions on Vehicular Technology*, vol. 69, 2020.

[18] A. A. Süzen, "Developing a multi-level intrusion detection system using hybrid-DBN[J]," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[19] W. Liang, K. C. Li, J. Long, X. Kui, and A. Y. Zomaya, "An industrial network intrusion detection algorithm based on multifeature data clustering optimization model," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2063–2071, 2020.

[20] D. Huang, X. Shi, and W. A. Zhang, "False data injection attack detection for industrial control systems based on both time and frequency-domain analysis of sensor data[J]," *IEEE Internet of Things Journal*, no. 99, p. 1, 2020.

[21] K. Zhang, C. Shen, H. Wang, Z. Li, Q. Gao, and X. Chen, "Cluster computing data mining based on massive intrusion interference constraints in hybrid networks[J]," *Cluster Computing*, vol. 22, no. 3, pp. 7481–7489, 2019.

[22] A. Alharbi, W. Alosaimi, H. Alyami, H. T. Rauf, and R. Damaševičius, "Botnet attack detection using local global best Bat algorithm for industrial Internet of things," *Electronics*, vol. 10, no. 11, p. 1341, 2021.

[23] M. H. Ali, M. M. Jaber, S. K. Abd et al., "Threat analysis and distributed denial of service (DDoS) attack recognition in the Internet of things (IoT)," *Electronics*, vol. 11, no. 3, p. 494, 2022.

[24] M. Wozniak, J. Silka, M. Wieczorek, and M. Alrashoud, "Recurrent Neural Network model for IoT and networking malware threads detection[J]," *IEEE Transactions on Industrial Informatics*, vol. 1, no. 99, 2020.

[25] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[26] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," *Advances in Neural Information Processing Systems 30 (NIPS 2017*, China, 2017.

[27] K. Xie, Y. Yang, Y. Xin, and G. Xia, "Cellular neural network-based methods for distributed network intrusion detection," *Mathematical Problems in Engineering*, vol. 2015, no. 3, pp. 1–10, Article ID 343050, 2015.

[28] Y. Y. Huang, D. Wang, Y. Sun, and B. Hang, "A fastin tra-coding algorithm for HEVC by join tly utilizing naïve Bayesian and SVM," *Multimedia Tools and Applications*, vol. 79, no. 45, pp. pp33957–33971, 2020.

[29] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Nov.,* vol. 12, pp. 1–6, 2015.

[30] J. Toldinas, A. Venčkauskas, R. Damaševičius, Š. Grigaliūnas, and Morkevičius, "A novel approach for network intrusion detection using multistage deep learning image recognition," *Electronics*, vol. 10, no. 15, p. 1854, 2021.