ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Research on collaborative edge-cloud intrusion detection system based on spatio-temporal graph neural network for power IOT communication

Shuyang Guo[1,*], *Ning Wang*[1], *Zirui Wang*[1], *Xiaobin Lian*[1] *and Mengying Xiong*[1]

[1] The Information and Communication Branch of Hainan Power Grid, Haikou, Hainan, 570000, China

SUMMARY: *The communication environment of the power Internet of Things (IoT) is characterized by massive heterogeneous nodes, strong spatiotemporal correlations in traffic behavior, and constrained edge resources. Traditional intrusion detection methods struggle to address its complex and dynamic security threats. This paper first employs convolutional neural networks (CNNs) for offline training in the cloud to generate diverse baseline detection models. It then introduces Pearson correlation coefficients for feature selection and utilizes the ST-GCN model to deeply mine the inherent spatial-topological dependencies and temporal dynamics within power communication traffic. The ST-GCN model achieves an accuracy of 98.62% on the test set, significantly outperforming CNN (96.39%) and GCN (97.31%). In specialized DDoS attack detection, ST-GCN achieved an accuracy of 98.63% with a false positive rate as low as 0.01%, while maintaining an average detection latency of only 22.12ms. Its performance comprehensively outperformed multiple traditional machine learning baseline models. Under simulated harsh communication conditions with high packet loss rates (up to 5%), the ST-GCN model maintained an accuracy of 90.29%, demonstrating exceptional robustness. Ablation experiments further confirmed the significant contributions of the feature selection and spatio-temporal modeling modules to enhancing detection performance. The edge-cloud collaborative intrusion detection system based on ST-GCN provides efficient, precise, and reliable proactive security protection for the power Internet of Things.*

KEYWORDS: *Power Internet of Things; Intrusion Detection; Edge-Cloud Collaboration; Spatio-Temporal Graph Convolutional Network; Online Ensemble Learning; Feature Selection*

1 Introduction

Under the conditions of smart grids, the IoT of the power industry is being developed. By making power grids effective, reliable, secure, and intelligent in their operation, the IoT of the power industry ensures an increase in the socio-economic value of power systems [1]. Through the implementation of IoT technology, the reliability and security of power systems have been considerably improved, the efficiency of operation increased many times over, energy consumption minimized, and automation control reached. Thus, the basis for the creation of smart grids has been established [2-4]. As technology develops further, IoT technology has entered all segments of the power industry: from generation and transmission to power distribution and sales, as well as business management. Therefore, the power IoT is an essential direction in the development of the country, which allows for the rational use of national energy resources, the intelligent modernization of power systems, and considerable contributions to

*gsy20230429@163.com
https://doi.org/10.65102/is2026134
Published: 30 April, 2026

the country's development and building [5, 6].

But in fact, the power IoT is a network structure that comprises multiple nodes communicating among themselves to accomplish tasks such as data acquisition and control [7]. The extensive nature of the network, its complex interconnectivity and interdependence, and the massive amount of data being transmitted through the physical, communication, and application layers of the IoT make it extremely susceptible to viruses, Trojan horses, backdoor programs, and other attacks. This would jeopardize the safety of the power system and have serious implications for its overall operation [8-11]. On another front, the consistent exchange of data in the power IoT makes it highly susceptible to data-based attacks, thereby exposing it to the possibility of being hacked. In case of successful hacking, it will have a drastic effect on the overall power system [12, 13]. Hence, there is a pressing need to develop efficient intrusion detection techniques for the power IoT.

Recently, with the development of IoT technology and increasing demand for real-time processing and low latency, the concept of edge-cloud cooperation is becoming a new trend that attracts considerable interest and implementation. Coordinating data processing and computing in both cloud and edge computing can be accomplished by means of the joint effort of both cloud computing and edge computing, resulting in increased efficiency in data processing and reduced network latency [14-17]. Concerning the power IoT application, Gong et al. (2019) proposed a scheme of management and control of security in the ubiquitous power IoT through the collaboration of cloud and edge computing corresponding to State Grid development. This strategy improves the safety of grids and integrity of data processing as well as enhances communication processes and remote backup services [18].

They used trust values extracted through real-time monitoring of node communications and interactions to identify abnormal nodes in the network. They established smart contracts using blockchain technology to monitor anomalous nodes on a blockchain that would be synchronized to all nodes on the network [19]. Chen et al. (2023) applied edge-cloud cooperation technology to perform communication attacks monitoring within Internet of Things devices in cyber-physical systems. It helped increase the efficiency of the hardware devices in the process of detection and enabled real-time processing during detection of large-scale hierarchical communication attacks [20]. In 2023, Yang et al. proposed a collaborative IoT edge-cloud intrusion detection system. The technology uses data dimensionality reduction, temporal convolutional networks to capture long sequences of traffic data features, and incorporates federated learning to improve efficient training. Importantly, the system can detect attacks unknown locally by edge devices [21]. But there exist several barriers to use the concept of edge-cloud collaborative detection in power IoT such as inefficient edge-cloud collaboration and generalization capacity, additional communication overhead in edge optimization, and inference latency problems at cloud optimization [22].

Machine learning, deep learning, and GNN are currently being used effectively in the domain of intrusion detection for IoTs. Amouri et al. (2020) proposed a technique for intrusion detection in mobile IoTs through machine learning techniques. The authors used dedicated sniffers for collecting data from IoTs, which were processed by linear regression after being passed on to supernodes. With this model, they could classify nodes into either intruders or non-intruders with a detection accuracy of over 98% in a scenario with high speeds of nodes [23]. The work of Saheed et al. (2022) presents an IoT intrusion detection system using machine learning approach, where the dataset is scaled, the dimensionality is reduced by applying PCA, then analyzing and classifying the system using machine learning techniques with detection rate reaching 99.9% [24]. Taghavinejad et al. (2020) implemented three decision trees for IoT intrusion detection in smart grids and found out that using the three-decision tree approach outperforms other techniques,this includes techniques based on decision trees, support vector

machines, and k-nearest neighbors [25]. However, according to Abdelmoumin et al. (2021), machine learning techniques that are anomaly based and are used in the IDS yield relatively low detection rates compared to deep learning techniques because they require more data [26]. Also, even though traditional machine learning provides great interpretability, they depend on feature engineering manually, making the models less generalized [27, 28].

In their research, Alkahtani and Aldhyani (2021) proposed a strong intrusion detection model based on IoT for smart grids using CNN, LSTM, and CNN-LSTM models for intrusion detection. Using a particle swarm optimization technique to detect data feature extraction in IoT systems, LSTM attained 99.82% accuracy [29]. Bi et al. (2025) constructed an intrusion detection model of IoT for power grids. Their model mainly depends on deep belief networks for extracting high-dimensional and complex data and reducing its dimension. In their approach, bidirectional LSTM captures event dependence among data points to improve detection accuracy. The bidirectional LSTM+CNN model improves efficiency and robustness [30]. Chen et al. (2025) designed a synaptic-intelligent CNN intrusion detection model in IoT systems. The synaptic intelligent algorithm enhances the efficiency of CNNs by resolving detection forgetting issues and complicated training problems. It uses a new loss function that addresses class imbalance and vanishing gradient problems in IoT devices and implements quantization to improve intrusion detection efficiency in dynamic environments [31]. Asha et al. (2025) applied recurrent neural networks and gated recurrent units to detect cyberattacks and relationships between cyberattacks in smart grid IoT systems, resulting in a superior detection mechanism than conventional machine learning and deep learning approaches [32]. Song & Ma (2024) created a federated CNN with edge support IoT intrusion detection model that can generate detection outcomes despite limited resources. The proposed model achieved an accuracy rate of 99%, while at the same time, lowering CPU and memory usage compared to traditional deep learning models [33]. Nevertheless, deep learning IoT intrusion detection relies heavily on massive amounts of training data and computation power. While deep learning offers automatic feature extraction benefits, detecting instances with insufficient data and computation power is difficult [34, 35].

In the domain of IoT intrusion detection, there are many applications of GNN that involve creating links between different elements of IoT network traffic. The GNN based IoT intrusion detection system introduced by Villegas-Ch et al. (2025) proves to be effective in its functionality and scalable, making it possible to cope with various attack situations encountered in the practical world [36]. Altaf et al. (2023) developed an IoT intrusion detection model based on a cascaded polygonal GNN which combined the capabilities of spectral GNN and spatial GNN in order to improve edge feature extraction. This resulted in very high detection accuracies and precisions with rates reaching up to 99.96% [37]. Likewise, Lin et al. (2025) designed an edge-based IoT intrusion detection system driven by a GNN, using the global attention and local gates approach. In order to improve data feature extraction, graph contrastive learning was used [38]. Musharraf et al. (2025) utilized the power of GNN algorithms to combine edge feature and structural knowledge within the IoT network flow. The method illustrated the attack behavior and deviations from normal network flows to the network managers, solving the problem of the false-negative rate of complicated attacks using machine learning intrusion detection methods and improving generalization performance [39].

Nevertheless, GNNs may ignore the temporal dimension in the detection process and find it challenging to incorporate the spatiotemporal connections in the power system data [40]. STGNNs successfully resolve this challenge. With the capability to concurrently learn the temporal change and spatial relationship within the graph structure, STGNNs have been demonstrated to be flexible in various complex situations [41]. The authors introduced the

federated learning model in the STGNN model to enable cooperative learning between various IoT nodes. The approach limited the need for data transmission but enhanced intrusion detection accuracy in IoT networks [42].

In this paper, we will discuss the important techniques used for intrusion detection systems in edge-cloud collaboration. We will first propose an edge-cloud collaborative intrusion detection model using online ensemble learning techniques. By combining the strengths of offline and online learning methods, we use deep learning algorithms (CNN) for offline learning of historical traffic data within the cloud center to generate robust and diverse base learners. Meanwhile, the model uses online learning to learn the new samples continuously. The proposed model uses time-series performance-based weighted ensemble methods to integrate the output of different online models, which helps to tackle the problem of time variation in attacks and improves the adaptability and reliability of the algorithm. Based on this idea, we can further propose a feature selection technique based on Pearson correlation coefficients. The objective is to extract the most pertinent and significant features associated with network intrusion activities from the power communication data sets in their high dimensions but with redundant and irrelevant features excluded. The approach not only helps in simplifying the problem computationally by reducing complexity in resource-limited edge computing environments, but also helps in minimizing noise contamination. In order to understand the spatiotemporal association of traffic within the power communication network, a Spatio-Temporal Graph Convolutional Network (ST-GCN) architecture is utilized. The authors conduct a detailed analysis of the principle of graph convolution in ST-GCN as well as neighborhood partition schemes, such as unified labeling, distance, and spatial configuration.

2 Edge-Cloud Collaborative Intrusion Detection Based on Spatio-Temporal Graph Neural Networks

2.1 Edge-Cloud Collaborative Intrusion Detection Method

2.1.1 Intrusion Detection Method Based on Online Ensemble Learning

In the current open edge-cloud collaboration environment, the performance overhead of securing massive edge access in this novel architecture has become increasingly prominent. To ensure the trustworthiness of terminal access, developing a lightweight intrusion detection system can not only provide comprehensive guidance for network security but also alleviate network communication transmission pressure while accelerating intrusion detection response times. However, this approach cannot effectively defend against attacks, as hackers employ ever-evolving tactics. Only through continuous updates to the intrusion detection system can effective defense against hacker intrusions be achieved. Therefore, this paper adopts an online ensemble learning method. By leveraging online learning integration technology, it meets the system's requirement for data timeliness, thereby enhancing system reliability and responsiveness.

First, deep learning models are used to train offline data, constructing offline base learners. Subsequently, multiple sets of online base learners are constructed based on continuously supplemented data. These online base learners are derived through incremental online training, with the capabilities of the online base learners enhanced by selecting generations from the offline base learners. Third, integrating the performance of multiple sets of online base learners, a time-series performance-weighted online ensemble scheme is proposed. This approach reduces the variance of the online learning model, thereby improving the reliability and stability of the intrusion detection system.

2.1.2 Offline Learning Scheme Design

To maintain diversity in intrusion detection models, this paper divides offline data into multiple time-series segments and employs deep learning methods to train traffic sequences. This learning process constructs intrusion detection models using historical traffic sequences, with relatively low temporal requirements. Based on feature extraction from each time-series dataset, features are fused to form an offline base learner for the current traffic sequence, ultimately generating an offline intrusion detection model. Thus, deep learning approaches can be employed to train intrusion detection models.

Design of Offline Learning Scheme is represented in Figure 1 below. The offline learning scheme mainly utilizes a CNN to handle huge intrusion data and classify such intrusion data.

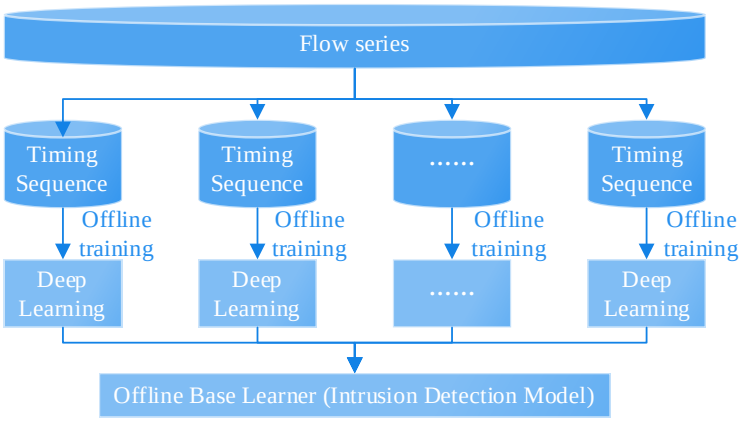

Figure 1: Offline learning program design

The specific steps are as follows:

(1) After obtaining the time-series data, convert it into numerical format and normalize all values to ensure that feature extraction is not influenced by numerical variations.

(2) This paper employs a gated convolution approach to eliminate invalid and redundant signals within the time series. Values obtained from two distinct convolution methods are multiplied to derive the optimal convolution layer, as illustrated below:

$$X = E * W + b \tag{1}$$

$$Y = E * V + c \tag{2}$$

$$g(E) = X \otimes \mathrm{Relu}(Y) \tag{3}$$

Here, E denotes the normalized time series from step (1), W and V represent the weight parameter vectors in different convolutional layers, b and c denote the bias vectors in the neurons, $\otimes$ indicates the multiplication of values obtained from two different convolutions, and the ReLU function serves as the activation function for the convolutional neural network.

(3) After feature extraction from the time series through the convolutional layer, pooling is applied to compress the data features, eliminating a large number of weight parameters requiring training and thereby improving the training efficiency of the detection system.

(4) After aggregation, the aggregated features are then linked to the fully connected layer. This layer combines all data processed x prior to passing it on to the output layer. The output layer makes use of the soft function for classifying the data, taking the class with the highest probability as the output of classification. The mathematical formula for the soft function is:

$$f_\theta\left(x^{(i)}\right)=\begin{bmatrix} p\left(y^{(i)}\right)=1 \mid x^{(i)};\theta\right) \\ p\left(y^{(i)}\right)=2 \mid x^{(i)};\theta\right) \\ \cdots \\ p\left(y^{(i)}\right)=k \mid x^{(i)};\theta\right) \end{bmatrix} \tag{4}$$

Here, $x^{(i)}$ denotes the data from the i nd temporal sequence transmitted to the output layer after integration in the fully connected layer; $y^{(i)}$ represents the activation value of the i th temporal sequence after passing through the Soft function; $p\left(y^{(i)}\right)=1$ indicates the probability that the activation value of the i th temporal sequence belongs to class 1; and θ signifies the parameter within the Soft function.

From steps (1)-(4), it can be seen that after training on multiple time-series datasets, multiple offline intrusion detection models are constructed after the training of convolutional neural network on offline data is completed. The data distributions of different time periods in the edge-cloud cooperative environment generally have large whole differences, and the offline intrusion models of a single time series often suffer from insufficient generalization ability and weak robustness. Therefore, in this paper, on the basis of obtaining the construction of multiple effect line detection models, the weighted average integrated learning is used to aggregate multiple offline detection chess models to construct an offline base learner.

2.2 Feature Selection—Pearson Correlation Coefficient

To further optimize model inputs, we will now introduce feature selection methods, focusing on the application of Pearson's correlation coefficient in power IoT communication data. It considers those attributes that have high correlation to intrusion behavior, thus simplifying the model and improving its performance.

An attribute can either be classified as relevant or irrelevant. Relevant attributes are those attributes that help in classification, whereas irrelevant attributes are those that have no effect at all in classification. Feature selection refers to the selection of features from the targeted dataset. Increasing number of features may not improve the outcomes if there is an adequate number of samples. The objective of feature selection is to get the best possible model performance with fewer features. It solves the problem of computationally intensive modeling brought about by excess dimensions by eliminating irrelevant features.

The most widely used technique of feature selection in this research work is Pearson correlation As a statistical concept, the Pearson correlation coefficient is widely applied for quantifying the linear association between variables, under the assumption that there is monotonic change in the values of the two variables. The Pearson correlation coefficient remains unaffected by changes in the scale or position of variables. It is highly computationally efficient, demonstrating outstanding performance when handling large datasets. The measurement range of the Pearson correlation coefficient spans $[-1,1]$; this interval allows it to characterize a wide variety of relationships. A result closer to 1 indicates a stronger positive correlation between variables, while a result closer to -1 indicates a stronger negative correlation. A result closer to 0 indicates a weaker correlation. In other words, the closer the absolute value of the Pearson correlation coefficient is to 1, the higher the correlation, whereas the closer it is to zero, the lower the correlation. The formula is shown in (5), where S is the correlation coefficient value, $\bar{X}$ is the sample mean corresponding to X_i, and $\bar{Y}$ is the

sample mean corresponding to Y_i.

$$S = \sum^{n}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right) \Big/ \sqrt{\sum^{n}\left(X_i - \bar{X}\right)^2}\sqrt{\sum^{n}\left(Y_i - \bar{Y}\right)^2} \tag{5}$$

In summary, a higher Pearson correlation coefficient indicates a stronger relationship between attributes, while a lower coefficient signifies weaker or even negative correlation. Pearson correlation coefficient could represent the relationship intuitively.

2.3 Spatio-Temporal Graph Convolutional Neural Network Model ST-GCN

There is some special spatio-temporal correlation in power IoT communication data, which makes the traditional modeling method unable to capture it. In order to resolve this problem, another model, known as ST-GCN, will be introduced in this paper. This model is able to work on both spatio-topological structure and time-series, which is quite useful for intrusion detection purposes in collaborative computing platforms.

2.3.1 Spatiotemporal Graph Convolution Operations

Single-frame Graph CNN Model: Within a single frame at time l, there are R_t key points, and the edges connecting these points are defined as N keypoints R_t within a single frame at time l. The edges connecting these points are defined as $E_s(l) = \left\{ r_{t_i} r_{t_j} \mid l = T, (i, j) \in H \right\}$. Based on the definition of convolutions of two-dimensional images or feature maps, the input borders can be considered a 2D grid, while the resulting feature map from the process of convolution can also be assumed to form a 2D grid. As long as the right stride is chosen, the size of the output feature map will be equal to that of the input image. Consider a $K \times K$ convolution kernel, an input image f_{in} with c channels. The output for a single channel at position x is:

$$f_{out}(x) = \sum_{h=1}^{K}\sum_{w=1}^{K} f_{in}(p(x,h,w)) \cdot w(h,w) \tag{6}$$

Here, p denotes the sample function with respect to position x, neighborhood h, and weights. In image convolution, $p(x,h,w) = x + p'(h,w)$; the weight function $w: Z^2 \to R^c$ represents a c-dimensional weight vector in real space, used to compute the inner product of input feature vectors in the c-dimensional sample. The values of the weight variables are independent of the input x's position, allowing filter weights to be shared across input images. Standard convolution in the image domain is achieved by encoding a rectangular grid within $p(x)$. Apply the convolution operation from the above formula to the input features of the spatial graph V_t. Here, the input feature graph $f_{\text{in}}^{t}: V_t \to R^c$ represents any node in the image. To adapt the model for spatio-temporal graphs, first optimize the sampling function and weight function.

In the video image, $p(h,w)$ in the sampling function represents the neighborhood pixels of the central pixel x. In the spatio-temporal graph, a node's neighborhood set can be similarly defined as $B\left(v_{ti}\right) = \left\{ v_{tj} \mid d\left(v_{tj}, v_{ti}\right) D \right\}$ with $D = 1$, i.e., the neighborhood set with distance 1,

where $d\left(v_{tj}, v_{ti}\right)$ denotes the shortest path from v_{tj} to v_{ti}. Therefore, the sampling function $p:\ B\left(v_{ti}\right) \to V$ can be expressed as:

$$p\left(v_{t_i}, v_{t_j}\right) = v_{t_j} \tag{7}$$

Unlike the sample function, the weight function is difficult to describe. The two-dimensional convolution operation inherently involves a strict grid surrounding the center point position, and hence the neighboring points have a certain ordering relationship. The weight function is calculated according to the spatial index tensor (c, K, K). As regards the randomly arranged general graph described above, a way to determine the order has been proposed by Wang et al., which takes into account the labeling of the neighborhood graph around the root node position.Neighborhood nodes are not assigned labels individually. Instead, the neighborhood set $B\left(V_{t_i}\right)$ of a given key node V_{t_i} is partitioned into a fixed number K of subsets. Each subset is assigned a numerical label, with the mapping defined as $l_{t_i}: B(v_{t_i} \to \{0, \cdots, K-1\}$. Therefore, the weight function $w\left(v_{t_i}, v_{t_j}\right):\ B\left(v_{t_i}\right) \to R^c$ can be implemented as a vector indexed by dimensions (c, K):

$$w\left(v_{t_i}, v_{t_j}\right) = w'\left(l_{t_i}\left(v_{t_j}\right)\right) \tag{8}$$

Based on the redefinition of the sample function and weight function, we now apply Equation (1) to graph convolutions:

$$f_{out}\left(v_{t_i}\right) = \sum_{v_{t_j} \in B\left(v_{t_i}\right)} \frac{1}{Z_{t_i}\left(v_{t_j}\right)} f_{in}\left(p\left(v_{t_i}, v_{t_j}\right)\right) \cdot w\left(v_{t_i}, v_{t_j}\right) \tag{9}$$

Among these, the regularization term $Z_{t_i}\left(v_{t_j}\right) = \left|\left\{v_{t_k} \mid l_{t_i}\left(v_{t_k}\right) = l_{t_i}\left(v_{t_j}\right)\right\}\right|$ corresponds to the cardinality of the respective subset, balancing the contribution of different subsets to the output. From Equations (6) to (8), we can derive:

$$f_{out}\left(v_{t_i}\right) = \sum_{v_{t_j} \in B\left(v_{t_i}\right)} \frac{1}{Z_{t_i}\left(v_{t_j}\right)} f_{in}\left(v_{t_j}\right) \cdot w\left(l_{t_i}\left(v_{t_j}\right)\right) \tag{10}$$

When treating the input image as a conventional 2D grid, Equation (10) can be computed as a standard 2D convolution. To apply it to a 3×3 convolution operation, the pixel-centered 3×3 grid neighborhood is divided into nine subsets, each containing one pixel.

After defining the Spatial Graph CNN, dynamic modeling of the skeleton sequence begins in both space and time. Spatial graphs are formed by connecting identical joints across adjacent frames. We now extend the spatial graph to a spatiotemporal graph, simultaneously expanding the neighborhood set to include joints connected temporally:

$$B\left(v_{t_j}\right) = \left\{v_{qj} \mid d\left(v_{tj}, v_{ti}\right) \leqslant K, \mid q - t \mid \leqslant \lfloor \Gamma / 2 \rfloor\right\} \tag{11}$$

In this case, the parameter Γ represents the size of the temporal kernel that dictates the temporal distance between neighborhood graphs. In order to conduct convolutions in the spatio-temporal graph, the sampling function, the weighting function, and the labeling graph l_{ST} are needed. The sampling function is the same as that for the spatial graph. As the time dimension is already ordered, the transformation function for manipulating the labeling graph according to the spatio-temporal neighborhood based on v_{t_i} is:

$$l_{\mathrm{ST}}\left(v_{q_j}\right)=l_{t_i}\left(v_{t_j}\right)+\left(q-t+\lfloor \Gamma/2 \rfloor\right)\times K \tag{12}$$

Here, $l_{t_i}\left(v_{t_j}\right)$ denotes the label graph for the single frame v_{t_i}. Thus, a well-defined convolution operation is established on the spatio-temporal graph.

2.3.2 Partitioning Strategy

Once we have described the convolution operation on spatio-temporal graphs, the design of a suitable partitioning scheme to apply the labeling graph will be just as important. This article investigates three schemes for partitioning. As they can naturally be adapted for the spatio-temporal graphs through Equation (12), we are going to discuss only the case of one frame from now on.

Unified Label: The simplest method is to put all elements of the neighborhood set into one partition. In this case, the inner product of the weights corresponding to any two adjacent nodes is the same. The problem with this partitioning scheme is that when applied to a one frame image, it is identical to taking the inner product of the mean vector of adjacent nodes' features with their respective weights, which means $K=1, l_{t_i}\left(v_{t_j}\right)=0, \forall i, j \in V$.

Distance Partitioning: Another partitioning strategy involves dividing the neighborhood set based on the distance $d\left(\cdot, v_{t_i}\right)$ from each node to the root node v_{t_i}. Setting $D=1$ divides the neighborhood set into two subsets: $d=0$ represents the root node itself, while $d=1$ contains nodes adjacent to the root node. Thus, different distances correspond to distinct weight vectors, enabling modeling of local differential properties (e.g., relative joint movements), expressed as $K=1, l_{t_i}\left(v_{t_j}\right)=d\left(v_{t_j}, v_{t_i}\right)$.

Spatial Arrangement: As the human body is spatially confined, this particular arrangement can be utilized in partitioning. Movements within the body parts may be classified as either concentric or eccentric motions. The neighborhood set of a vertex is thus split into three categories: (1) The root vertex itself; (2) Centripetal cluster: vertices that are nearer to the center of mass of the skeleton compared to the root vertex; (3) The centrifugal cluster. In this case, the center of mass of the whole skeleton is computed by averaging the coordinate positions of all joint vertices in a single frame of a skeleton graph. This can be mathematically stated as:

$$l_{t_i}\left(v_{t_j}\right)=\begin{cases}0 & \text{if } r_j=r_i\\ 1 & \text{if } r_j<r_i\\ 2 & \text{if } r_j>r_i\end{cases} \tag{13}$$

Here, r_i denotes the average distance from the overall skeleton center of mass to joint i across all frames in the video image.

Figure 2 highlights three types of partitioning methods for visualization. In figure 2(a), an input skeleton frame is given. The joints are shown in blue circles, while the receptive field of the filter of $D=1$ is highlighted by the red dashed circle. Figure 2(b) displays the uniform labeling technique, in which all nodes within the set of neighbors are labeled in green circles. Figure 2(c) presents the distance partitioning method, in which there are two subgroups: the root node (in green circles) and its neighboring nodes (blue circles). Finally, figure 2(d) depicts the spatial configuration method, in which the cross in black denotes the center of mass of the entire skeleton. Nodes are labeled according to their distance from the center of mass of the skeleton and the root node (green): nodes close to the center of mass are defined as centripetal joints (blue), whereas centrifugal joints (yellow) are further away from the center of mass than the root node. Partitioning strategies were evaluated using simulations for skeleton-based behavior recognition. It is expected that more sophisticated partitioning methods will provide better modeling and recognition results.

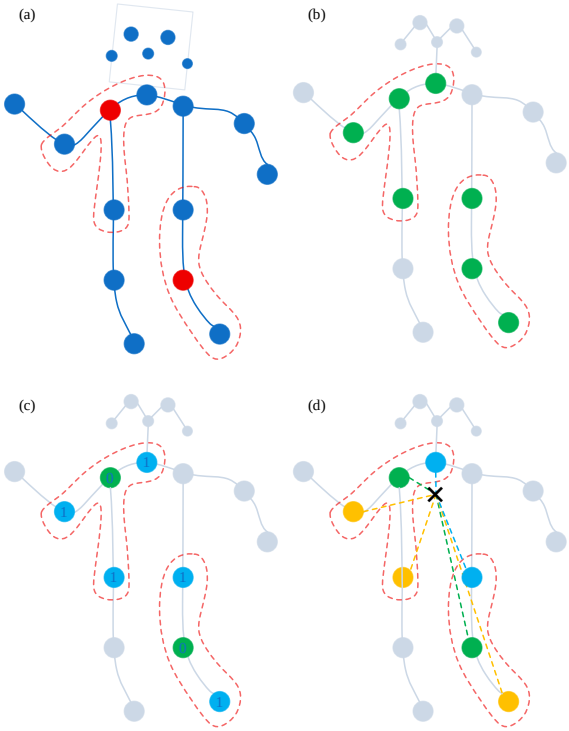

Figure 2: Input skeleton example graph frame (a) and three partition strategies (b~d)

Following the implementation approach used by Hu Zhengping et al., the connections within the body in one single frame of the video can be modeled through the adjacency matrix A and the identity matrix I. When processing a single frame of video imagery, the output result from the first partitioning strategy is computed using the following equation:

$$f_{out} = \Lambda^{-\frac{1}{2}}(A+I)\Lambda^{-\frac{1}{2}} f_{in} W \tag{14}$$

where $\Lambda^{ii} = \sum_j \left(A^{ij} + I^{ij}\right)$, the sum of weight vectors across multiple output channels yields the weight matrix W. In the spatiotemporal domain, the input feature map is represented as a $C \times V \times T \times \Gamma$ dimension tensor. A standard $1 \times \Gamma$ two-dimensional convolution is performed, multiplying the normalized adjacency matrix $\Lambda^{-\frac{1}{2}}(A+I)\Lambda^{-\frac{1}{2}}$ with the resulting tensor.

In both of the above partitioning approaches, which involve partitioning into more than one subset, the implementation methodology is essentially the same, and that involves breaking down the adjacency matrix into several matrices $A_j, A+I=\sum_j A_j$. For instance, in the distance partitioning strategy, $A_0=I, A_1=A$, then equation (14) transforms into equation (15):

$$f_{out}=\sum_j \Lambda_j^{-\frac{1}{2}} A_j \Lambda_j^{-\frac{1}{2}} f_{in} W_j \tag{15}$$

where $\Lambda_j^{ii}=\sum_k\left(A_j^{ik}\right)+\alpha$, and $\alpha=0.001$ is set to avoid empty rows of A_j . For each adjacency matrix, a learnable weight matrix M is associated. Replace the matrix $A+I$ in equation (14) with $(A+I)\otimes M$ and the $A_j.\otimes$ in equation (15) with $A_j \otimes M$, where the element-wise multiplication between two matrices is denoted by the symbol. The mask M is initialized as a full-ones matrix.

3 Experimental Validation and Performance Analysis of an Edge-Cloud Collaborative Intrusion Detection System for the Power Internet of Things

In order to prove the validity and feasibility of the proposed solution, this chapter will perform a number of experiments. In this way, the proposed approach will be comprehensively evaluated from several aspects, such as training efficiency, detection performance, real-time performance, and robustness. In addition, a detailed analysis will be conducted, taking into account the special features of power IoT communication.

3.1 Model Training and Benchmark Performance Comparison

The previous part created a model for intrusion detection based on cooperation between the edge and cloud computing models. Now, the model ST-GCN will be trained, using deep separable convolutions to create the model of ST-GCN for intrusion detection simulations.

3.1.1 Experimental Preparation

The dataset used in the experiments contains 4,738,291 records and the ratio of data for training and testing is 9:1. In the training dataset, there is only one normal label, while there are 23 attack types, such as DoS attacks, response injection (NMRI, CMRI), command injection (MSCI, MPCI, MFCI), and reconnaissance attacks. In addition, there are 15 unknown attack types in the test dataset.

All computations presented in this work have been done using a personal computer with a single-core 2.8 GHz processor and 5 GB of memory. The simulation process employed Python version 3.7.1 and the TensorFlow deep learning framework.

3.1.2 Model Training Loss

The loss plot for the obtained ST-GCN model after repeated training sessions can be seen in Figure 3.

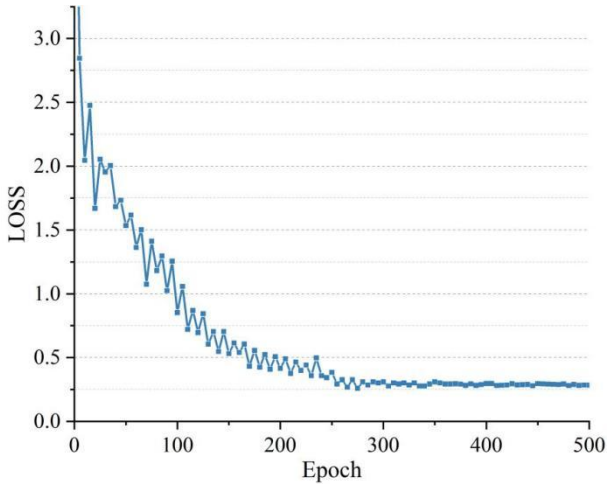

Figure 3: ST-GCN Model training loss curve

It is clear that the loss for the model reduces as the training iteration numbers increase until it stabilizes at 300 training iterations. Because of the availability of large amounts of data, an efficient model, and enough training time, the loss decreases with higher iteration numbers when training, thus confirming the efficiency of the model training procedure.

3.1.3 Model Accuracy Comparison Experiment

To verify the effectiveness of using ST-GCN in terms of accuracy for this study, an experimental comparison was carried out between ST-GCN, CNN, and GCN models. The performance results of the models in terms of accuracy for training and testing datasets are shown in Figures 4 and 5.

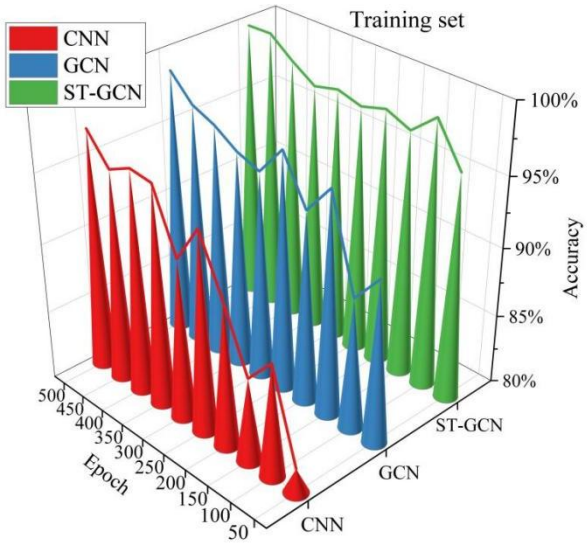

Figure 4: Accuracy rates of each model on the training set

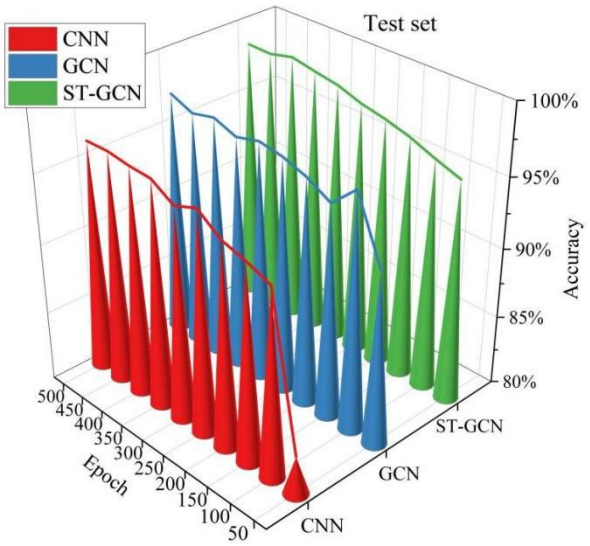

Figure 5: Accuracy rates of each model on the test set

In general, the proposed ST-GCN model performed much better in classification accuracy compared to conventional CNN and GCN models for the majority of the training iterations, including those on the training and testing datasets.

As far as the training set is concerned, the ST-GCN model showed great capabilities in learning powerful features already during the first iterations of training, with the accuracy of 96.15%, surpassing the 81.76% of CNN and 91.78% of GCN. While increasing the number of training iterations, the accuracy of the ST-GCN also kept increasing, reaching its peak value at 99.94% after 450 iterations, thus demonstrating excellent fitting capabilities of the proposed architecture. In turn, the CNN and GCN models also increased their accuracy towards the end of the training process, although this increase was unstable. The stable final accuracy rates of these two models (about 97%) were notably lower than those of the ST-GCN model.

Even the outcomes on the test set proved to be much more persuasive. The model ST-GCN was leading at all times, with its accuracy continuously increasing from 95.72% to 98.98%, which can indicate very strong generalization skills. Particularly, despite the presence of 15 unknown attack types in the test set, the model ST-GCN was able to detect attacks with extremely high accuracy rates. In addition, the performance curve of this model was absolutely smooth and stable, without showing any signs of overfitting. Although both models CNN and GCN showed higher accuracy scores on the test dataset compared to the training one, indicating their natural generalizability, their accuracy was always 1-3 percentage points lower compared to the ST-GCN model. As a result of training, the test accuracy rate of the model ST-GCN was 98.62% compared to 97.31% for GCN and 96.39% for CNN. This indicates the greater ability of ST-GCN to comprehend the intrinsic dynamics and topological connections of attacks due to simultaneous consideration of both.

As one can see, the above tests have successfully validated the advantages of the ST-GCN model over classical approaches. In addition to high training speed and stability, this model shows significantly higher generalization capability.

3.2 Edge-Cloud Collaborative DDoS Attack Detection Experiment

After implementing the training process and comparison experiments related to the ST-GCN model, in order to verify the actual application capacity of this method in specific attack cases, we also carried out targeted detection experiments on DDoS attacks that often occur in the edge-cloud collaboration environment.

3.2.1 Controller Load Variation

An experiment design was carried out to determine the variation in load of the controllers as they were under attacks via the DDoS attacks at varying packet rates. This was done by carrying out the DDoS attacks using hping3 attack packages. This analysis was done for c0, c1, and c2 controllers with packet rates of 1000 packages per second and 2000 packages per second. This is illustrated in Figures 6 and 7.

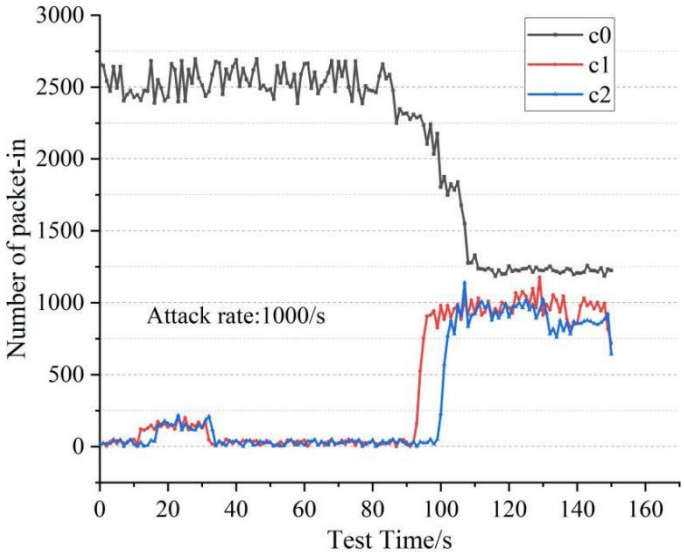

Figure 6: The load of controllers c0, c1&c2 under the attack rate of 1000 per second

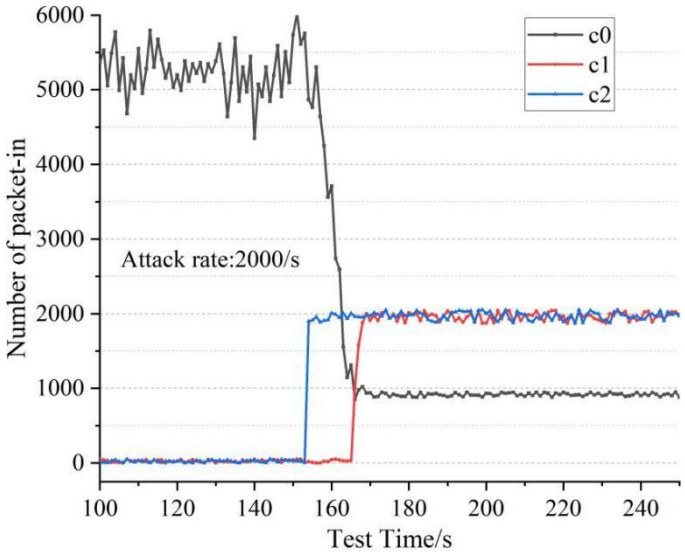

Figure 7: The load of controllers c0, c1&c2 under the attack rate of 2000 per second

Under simulated DDoS attack rates of 1000 packets per second, the load on all three controllers increased in a stepwise manner over time. As the central controller, c0 showed a lot of fluctuations on the load curves, suggesting that it was affected most heavily by attack packets processing. An overloaded inflection point occurred during the middle of the attack. The loads of c1 and c2 did not change much; packet processing at the peak was always kept at about 1000, indicating that the edge-layer attack traffic diversion approach could reduce some loads. With an increased attack rate up to 2000 packets per second, the loads of all the controllers increased

more than 40% compared with Figure 6. From practical considerations, when DDoS attack rates increase, switches are sent a large number of attack packets and then uploaded to the controllers for processing, which is bound to raise controller load.

In general, the loads on c1 and c2 have been increased after load balancing, but the load on c0 was lowered. After load balancing, the distribution of load among all three controllers became even. It can be concluded that the method of spatio-temporal graph convolutional neural network was effective for accomplishing load balancing and defending against DDoS attacks in the network.

3.2.2 Comparison of DDoS Attack Detection Results Across Different Models

To further confirm the effectiveness of the proposed algorithm in detecting DDoS attacks, tests have been conducted by comparing the proposed model with other existing algorithms used in detecting DDoS attacks in SDN.

The baseline models adopted in these experiments were BP neural network, BPNN algorithm, SVM algorithm, KNN algorithm, and Naïve Bayesian algorithm.

Accuracy, Precision, Recall, and False Positive Rate (FPR) are some of the methods employed.

DDoS attack detection experiment results obtained from different models are illustrated in Figure 8. The black dots indicate the accuracy values, while the white dots and gray dots show Recall and FPR, respectively.

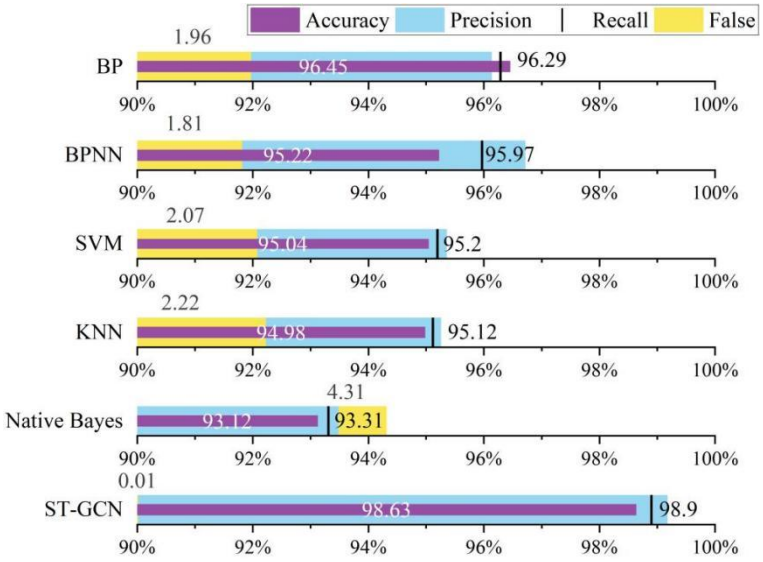

Figure 8: Comparison of DDoS attack detection results for each model

Generally, compared with other machine learning models, the ST-GCN model introduced in this paper has much better performance with respect to all criteria, and is able to effectively resist against the DDoS attack under edge-cloud collaboration in the power IoT.

Firstly, regarding the accuracy, the ST-GCN model reaches 98.63% while BP neural networks only get 96.45%, BPNN reaches 95.22%, SVM obtains 95.04%, KNN scores 94.98% and Naive Bayes achieves 93.12%. These results imply that the classification capability of ST-GCN model is much stronger than others mentioned above. Secondly, the precision of the ST-GCN model is 99.17%, which means the probability of false positives when making real warnings is very low. Lastly, regarding the recall rate, the ST-GCN model still gets 98.90%, far better than other models.

Above all, the false alarms ratio measure reveals that the model has an unusually low ratio

of just 0.01%, while the other competing models have ratios of 1.81% to 4.31%. Thus, the presented results confirm that the application of the proposed solution is highly valuable in practice. The absence of false alarms will ensure that the computer does not spend time processing unnecessary warnings, saving its resources for actual tasks.

Thus, the data provided in Figure 8 clearly illustrates the following fact: the spatiotemporal graph convolution neural network model possesses outstanding properties in terms of detection accuracy and operational effectiveness. This makes it possible to say that the suggested model can efficiently detect DDoS attacks in IoT communication networks.

3.2.3 Time Consumption Distribution in DDoS Attack Detection Process

Apart from evaluating different detection metrics, this paper also tests the real-time performance of the ST-GCN model. Detection of DDoS attacks on flow tables obtained can be considered an inference using spatiotemporal graph convolutional techniques. The experiment performs 500 inference times. The inference time consumption distribution is plotted in Figure 9.

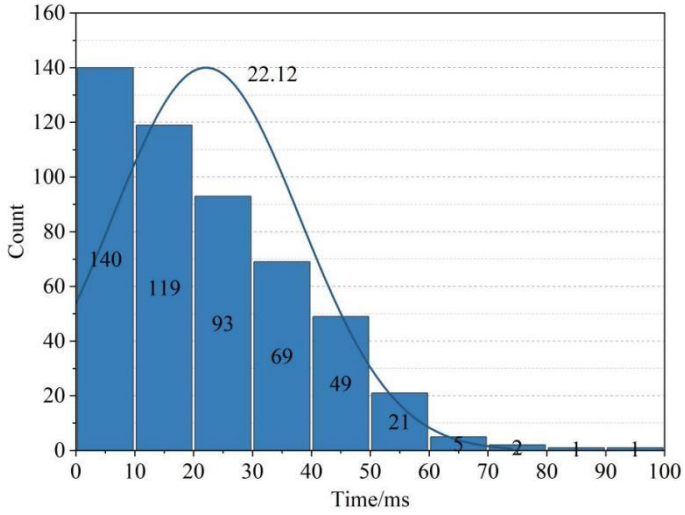

Figure 9: Time consumption distribution of DDoS attack detection of ST-GCN model

From the general distribution, we observe an approximate normal distribution of latency, mainly distributed in the low latency range (0-20 ms), which implies that the proposed ST-GCN model is effective in completing tasks in most cases, thus satisfying the requirement for low latency of power IoT communication. The mean time consumption after 500 inferences was 22.12ms. It shows that a relatively left-skewed distribution (most of the time consumption is less than the average) emphasizes the optimization effectiveness of the model through spatio-temporal graph convolutional techniques and eliminates long tail latency problems. At the same time, it shows the high responsiveness of the model in DDoS attack scenarios due to the small dispersion of distributions.

3.3 Power IoT Communication Environment Simulation Experiment

The problems like unstable channels and data packet losses often occur in real-world power IoT communication scenarios. In this section, the simulation for testing the performance of the model under packet losses is conducted with varied packet loss rates.

For emulating the low stability of data transmissions in real-world power IoT communication systems along with high packet losses, packet losses are randomly introduced into packets of samples from the test set of the dataset mentioned above. The packet loss rate

varies from 0% to 0.05%. The results of packet loss rates on the detection accuracy of CNN, GCN, and ST-GCN models can be observed in Figure 10.

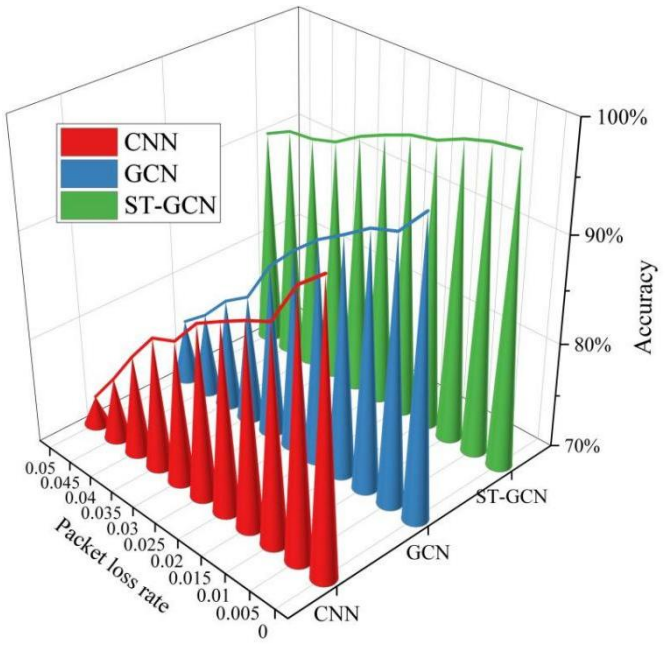

Figure 10: The accuracy of the model under different packet loss rate environments

In the case of the increase in the rate of packet losses to 0.05, the accuracy levels decreased in all models tested. Nevertheless, it was seen that the performance of the ST-GCN model showed a comparatively slow decrease and hence the greater level of robustness compared to others. At the level of packet losses equal to 0.01, the accuracy of the ST-GCN model was 97.51%, while that of CNN and GCN reached only 90.11% and 93.74%, respectively. In the case of increasing packet losses to 0.05, the accuracy rate was still 90.29% in the case of the proposed model, while CNN and GCN showed accuracy levels of 72.70% and 75.73%, respectively. This shows that the model performs better than other ones concerning maintaining its functionality in conditions of real-life situations where the network of power IoT communication can be unstable due to the data losses issue.

3.4 Ablation Experiment

In order to further study the role of each part of the model, this chapter performs the ablation experiment to explore the effect of feature selection module and spatiotemporal modeling module on the classification performance of the detector, so as to prove the necessity and rationality of the design of the model.

The specific settings of ablation experiments for ST-GCN model are:

Model 1: Without feature selection. Starting from the baseline model, the Pearson correlation coefficient-based feature selection module is discarded. All the raw features are used to train the ST-GCN model.

Model 2: Spatio-temporal modeling is excluded. According to the basic model, instead of using the ST-GCN model, use the general graph convolutional neural network (GCN) model. This model can only model the spatial topology of network traffic data but does not have the ability to model the temporal dynamics, thus proving the importance of spatio-temporal modeling.

A comparison experiment was carried out using 1000 randomly sampled samples from the

database. The results from the model's last layer were used for the y-axis value while the number of samples was used for the x-axis value. The results were further classified using a threshold method where each sample was marked as either normal or abnormal based on their labels.

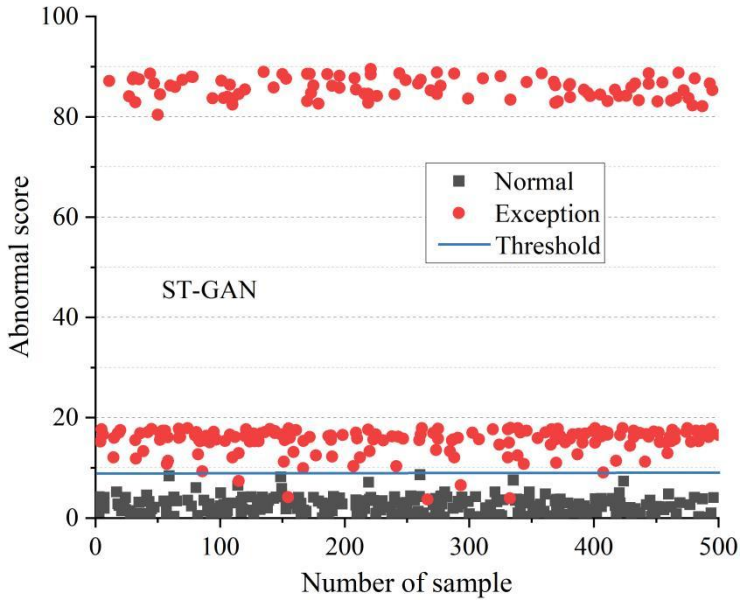

Figure11: The detection effect of ST-GCN on 500 samples

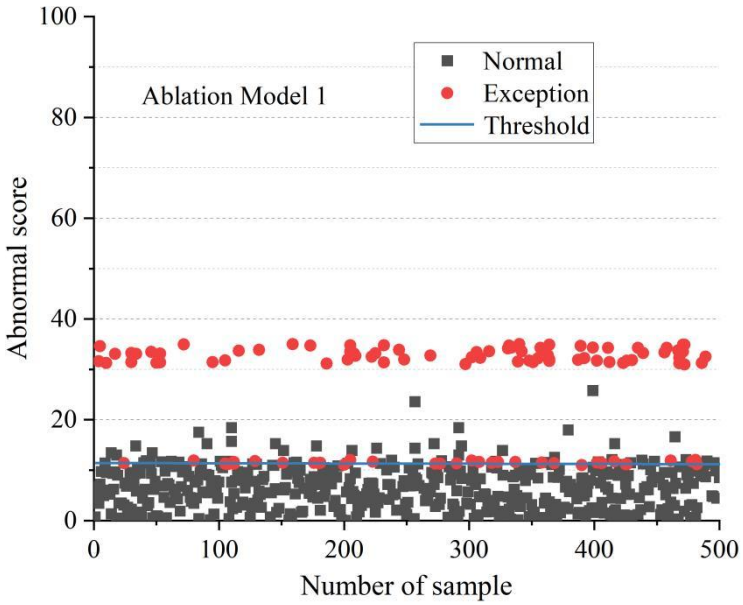

Figure12: The detection effect of Ablation Model 1 on 500 samples

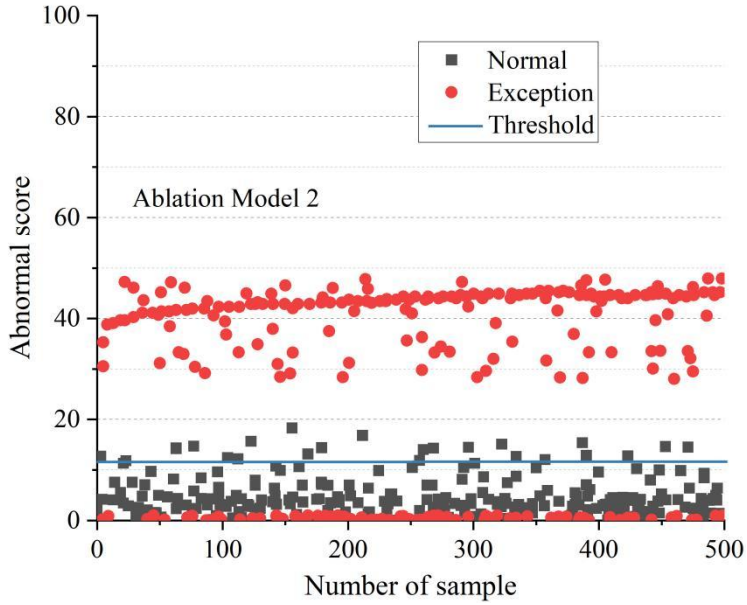

Figure13: The detection effect of Ablation Model 2 on 500 samples

From the experiment results, we can see that in ablation model 1, in which the feature selection module is not used, the score of many abnormal values becomes closer to normal values, while the number of misclassification becomes higher compared to other models. It indicates the importance of the feature selection module in filtering out redundant data, thus avoiding the interference of noise to the model. Without it, the sensitivity of intrusion behaviors decreases. Similarly, for ablation model 2, in which the spatio-temporal modeling module is not used, the number of misclassified samples rises, and at the same time, the clustering of the anomaly score becomes denser than other models. It proves the significance of using spatio-temporal graph convolutions to detect the dynamics of spatio-temporal correlations in attacks. In the case where this module is not used, the problem of dealing with changes over time is not handled well. Unlike the previous two models, the model based on ST-GCN is able to distinguish between anomalies and normal values, in which the number of misclassification is much lower, reaching only 7 samples.

4 Conclusion

This paper highlights the security challenges that arise due to the presence of many edge devices connecting to the power IoT communications domain, along with spatiotemporal attributes in the network traffic, which explores and designs an edge-cloud cooperative intrusion detection system using ST-GCN.

In terms of detection performance, the ST-GCN model exhibits superior detection results. For a test set consisting of 15 kinds of unknown attacks, the ST-GCN model obtains a detection accuracy of 98.62%, performing much better than traditional GCN models (97.31%) and CNN models (96.39%). Especially in terms of DDoS attack detection, this model performs significantly better in all aspects with a detection accuracy of 98.63%, precision of 99.17%, and recall of 98.90%. Meanwhile, it maintains a very low false positive rate of 0.01%, far lower than other compared models (1.81%-4.31%), suggesting that the proposed method is more reliable with fewer false alarm problems in practice.

For real-time performance, the proposed system ensures low latency in the power IoT application scenario. The ST-GCN model takes only 22.12 ms for detecting DDoS attacks on

average, showing an extremely tight inference time distribution, implying that the system could be able to detect attacks online.

Robustness-wise, this model proves itself to be highly adaptive to adverse network conditions. During experimentation under harsh communication channels with a high packet loss ratio, the ST-GCN model shows the slowest decline in performance. Even with a 5% packet loss rate, its attack detection accuracy was still an impressive 90.29%, far surpassing GCN (75.73%) and CNN (72.70%). This thoroughly proves the resilience and robustness of the system when dealing with data incompleteness problems faced in practical situations.

Quantitative tests have confirmed the efficacy of key modules. The results show that omitting either the feature selection or spatio-temporal modeling module would result in a sharp drop in performance and an observable increase in the number of wrongly classified examples, while the full ST-GCN model only made 7 mistakes. This further confirms, from the data perspective, that the use of the Pearson correlation coefficient for feature selection plays an essential part in decreasing data redundancy, and ST-GCN's spatiotemporal joint modeling is indispensable in detecting the essence of attacks.

About the Authors

Shuyang Guo (1993-12) female, Han Nationality, Hainan Haikou, master, engineer, research direction network security.

Ning Wang (1983-5) male, Han Nationality, Hainan Haikou, bachelor degree, associate senior engineer, research interests include ubiquitous network security, security protection construction, attack and defense target field, security operation, etc.

Zirui Wang (2002-3) male, Han Nationality, Hainan Haikou, bachelor degree, trainee, research direction: network security of ubiquitous power information system.

Xiaobin Lian (1999-1) male, Han Nationality, Quanzhou, Fujian Province, bachelor degree, assistant engineer, research direction pan-network information security.

Mengying Xiong (1997-7) female, Han Nationality, born in Fengcheng, Jiangxi Province, master's degree, trainee. Research interest: network security of ubiquitous power information system.

References

[1] Bedi, G., Venayagamoorthy, G. K., Singh, R., Brooks, R. R., & Wang, K. C. (2018). Review of Internet of Things (IoT) in electric power and energy systems. IEEE Internet of things Journal, 5(2), 847-870.

[2] Murdan, A. P. (2023). Internet of Things for enhancing stability and reliability in power systems. Journal of Electrical Engineering, Electronics, Control and Computer Science, 9(3), 1-8.

[3] Fadlullah, Z. M., Pathan, A. S. K., & Singh, K. (2018). Smart grid internet of things. Mobile Networks and Applications, 23(4), 879-880.

[4] Xie, L., Wu, J., Li, Y., Sun, Q., & Xi, L. (2022). Automatic generation control strategy for integrated energy system based on ubiquitous power internet of things. IEEE Internet of Things Journal, 10(9), 7645-7654.

[5] Hossein Motlagh, N., Mohammadrezaei, M., Hunt, J., & Zakeri, B. (2020). Internet of

Things (IoT) and the energy sector. Energies, 13(2), 494.

[6] Jiang, A., Yuan, H., Li, D., & Tian, J. (2019). Key technologies of ubiquitous power Internet of Things-aided smart grid. Journal of Renewable and Sustainable Energy, 11(6).

[7] Wang, Q., & Wang, Y. G. (2018, October). Research on power Internet of Things architecture for smart grid demand. In 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) (pp. 1-9). IEEE.

[8] Sarjan, H., Ameli, A., & Ghafouri, M. (2022). Cyber-security of industrial internet of things in electric power systems. IEEE Access, 10, 92390-92409.

[9] Ko, S. W., & Kim, S. L. (2018). Impact of node speed on energy-constrained opportunistic Internet-of-Things with wireless power transfer. Sensors, 18(7), 2398.

[10] Hou, R., Ren, G., Zhou, C., Yue, H., Liu, H., & Liu, J. (2020). Analysis and research on network security and privacy security in ubiquitous electricity Internet of Things. Computer communications, 158, 64-72.

[11] Yang, Q. (2019). Internet of things application in smart grid: A brief overview of challenges, opportunities, and future trends. Smart Power Distribution Systems, 267-283.

[12] Anand, P., Singh, Y., Selwal, A., Singh, P. K., Felseghi, R. A., & Raboaca, M. S. (2020). Iovt: Internet of vulnerable things? threat architecture, attack surfaces, and vulnerabilities in internet of things and its applications towards smart grids. Energies, 13(18), 4813.

[13] Chin, W. L., Li, W., & Chen, H. H. (2017). Energy big data security threats in IoT-based smart grid communications. IEEE Communications Magazine, 55(10), 70-75.

[14] Li, J., Gu, C., Xiang, Y., & Li, F. (2022). Edge-cloud computing systems for smart grid: state-of-the-art, architecture, and applications. Journal of Modern Power Systems and Clean Energy, 10(4), 805-817.

[15] Zhai, X., Peng, Y., & Guo, X. (2024). Edge-cloud collaboration for low-latency, low-carbon, and cost-efficient operations. Computers and Electrical Engineering, 120, 109758.

[16] Yuan, J., Xiao, H., Shen, Z., Zhang, T., & Jin, J. (2023). ELECT: Energy-efficient intelligent edge–cloud collaboration for remote IoT services. Future Generation Computer Systems, 147, 179-194.

[17] Shi, Y., Yi, C., Chen, B., Yang, C., Zhu, K., & Cai, J. (2022). Joint online optimization of data sampling rate and preprocessing mode for edge–cloud collaboration-enabled industrial IoT. IEEE Internet of Things Journal, 9(17), 16402-16417.

[18] Gong, Y., Chen, C., Liu, B., Gong, G., Zhou, B., & Mahato, N. K. (2019, November). Research on the ubiquitous electric power Internet of Things security management based on edge-cloud computing collaboration technology. In 2019 IEEE Sustainable Power and Energy Conference (iSPEC) (pp. 1997-2002). IEEE.

[19] Jiang, J., Xin, P., Wang, Y., & Zhou, C. (2023, September). Network Security Strategy Based on Cloud-Edge Collaboration in Electric Power Internet of Things. In 2023 6th

International Conference on Information Communication and Signal Processing (ICICSP) (pp. 695-699). IEEE.

[20] Chen, C., Li, Y., Wang, Q., Yang, X., Wang, X., & Yang, L. T. (2023). An intelligent edge-cloud collaborative framework for communication security in distributed cyber-physical systems. IEEE Network, 38(1), 172-179.

[21] Yang, R., He, H., Xu, Y., Xin, B., Wang, Y., Qu, Y., & Zhang, W. (2023). Efficient intrusion detection toward IoT networks using cloud–edge collaboration. Computer Networks, 228, 109724.

[22] Pan, X., Jiang, A., & Wang, H. (2020). Edge-cloud computing application, architecture, and challenges in ubiquitous power Internet of Things demand response. Journal of Renewable and Sustainable Energy, 12(6), 062702.

[23] Amouri, A., Alaparthy, V. T., & Morgera, S. D. (2020). A machine learning based intrusion detection system for mobile Internet of Things. Sensors, 20(2), 461.

[24] Saheed, Y. K., Abiodun, A. I., Misra, S., Holone, M. K., & Colomo-Palacios, R. (2022). A machine learning-based intrusion detection for detecting internet of things network attacks. Alexandria Engineering Journal, 61(12), 9395-9409.

[25] Taghavinejad, S. M., Taghavinejad, M., Shahmiri, L., Zavvar, M., & Zavvar, M. H. (2020, April). Intrusion detection in IoT-based smart grid using hybrid decision tree. In 2020 6th International Conference on Web Research (ICWR) (pp. 152-156). IEEE.

[26] Abdelmoumin, G., Rawat, D. B., & Rahman, A. (2021). On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things. IEEE Internet of Things Journal, 9(6), 4280-4290.

[27] Da Costa, K. A., Papa, J. P., Lisboa, C. O., Munoz, R., & de Albuquerque, V. H. C. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. Computer Networks, 151, 147-157.

[28] Du, K. L., Zhang, R., Jiang, B., Zeng, J., & Lu, J. (2025). Understanding machine learning principles: Learning, inference, generalization, and computational learning theory. Mathematics, 13(3), 451.

[29] Alkahtani, H., & Aldhyani, T. H. (2021). Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms. Complexity, 2021(1), 5579851.

[30] Bi, S., Wang, J., Song, J., Li, P., & Li, L. (2025). Research on the Intrusion Detection Model for Power Internet of Things Combining Deep Belief Network and BiLSTM. Journal of Cyber Security and Mobility, 14(3), 653-672.

[31] Chen, H., Wang, Z., Yang, S., Luo, X., He, D., & Chan, S. (2025). Intrusion detection using synaptic intelligent convolutional neural networks for dynamic Internet of Things environments. Alexandria Engineering Journal, 111, 78-91.

[32] Asha, A., Karthika, S., & PC, S. (2025, February). Securing smart grid IoT system using

a robust RNN based cyberattack detection framework. In 2025 International Conference on Electronics and Renewable Systems (ICEARS) (pp. 617-622). IEEE.

[33] Song, X., & Ma, Q. (2024). Intrusion detection using federated attention neural network for edge enabled internet of things. Journal of Grid Computing, 22(1), 15.

[34] Li, Y., Zuo, Y., Song, H., & Lv, Z. (2021). Deep learning in security of internet of things. IEEE Internet of Things Journal, 9(22), 22133-22146.

[35] Khan, A. R., Kashif, M., Jhaveri, R. H., Raut, R., Saba, T., & Bahaj, S. A. (2022). Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions. Security and communication networks, 2022(1), 4016073.

[36] Villegas-Ch, W., Govea, J., Navarro, A. M., & Játiva, P. P. (2025). Intrusion Detection in IoT Networks Using Dynamic Graph Modeling and Graph-Based Neural Networks. IEEE Access.

[37] Altaf, T., Wang, X., Ni, W., Yu, G., Liu, R. P., & Braun, R. (2023). A new concatenated Multigraph Neural Network for IoT intrusion detection. Internet of Things, 22, 100818.

[38] Lin, L., Zhong, Q., Qiu, J., & Liang, Z. (2025). E-GRACL: an IoT intrusion detection system based on graph neural networks. The Journal of Supercomputing, 81(1), 42.

[39] Musharraf, S. T., Khan, M. H., Khan, U. S., Hasan, M. Z., & Hussain, M. Z. (2025). E-GRAPHSAGE++: ENHANCING GRAPH NEURAL NETWORK-BASED INTRUSION DETECTION SYSTEMS FOR IOT NETWORKS. Spectrum of Engineering Sciences, 3(3), 583-595.

[40] Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., & Wang, K. I. K. (2021). Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet of Things Journal, 9(12), 9310-9319.

[41] Cini, A., Marisca, I., Bianchi, F. M., & Alippi, C. (2023, June). Scalable spatiotemporal graph neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 37, No. 6, pp. 7218-7226).

[42] Wang, Y., Li, J., Han, Z., Cheng, P., & Kumar, R. (2025). FedSTGCN: a novel federated spatiotemporal graph learning-based network intrusion detection method for the Internet of Things. Frontiers of Information Technology & Electronic Engineering, 26(7), 1164-1179.