Data Engineering and Applications

Research Paper

# Botnet Traffic Detection in the Internet of Things Based on Graph Convolutional Networks

**Tareq Al Fares[1, *], Waleed Al Masri[1] and Hussam Al Salem[2]**

[1] Faculty of Information Technology and Computer Science, Yarmouk University, 21163 Irbid, Jordan
[2] King Hussein School of Computing Sciences, Princess Sumaya University for Technology, 11941 Amman, Jordan
*Corresponding author: tareq.af@yu.edu.jo

**Abstract.** By using graph-based representation learning, it is possible to more effectively construct models of the various structural dependencies and relationships between different parts of IoT network traffic. This paper proposes a new botnet detection framework that uses Graph Convolutional Networks (GCNs) to learn the communication graph of nodes and the features of devices. First, the preprocessing, normalization, and feature engineering of all raw IoT data are completed. Finally, an attribute graph will be generated, which shows the topology and time of the IoT. To enhance discrimination ability and handle irregular graphs, a generalized GCN architecture with an attention module and node feature normalization is added. Experimental tests were conducted on the NSL-KDD and Bot-IoT datasets, covering cases of IoT-specific benchmarks and traditional benchmarks. The results show that the framework achieved an accuracy of 95.1% in attack traffic testing, with an F1 score of 0.95. Moreover, the framework also maintains relatively high accuracy in the presence of class imbalance and sparse training data, with accuracy decreasing by less than 3.5% when only 30% of the labeled samples are used. Ablation studies found that removing the attention layer or normalization can lead to a decrease in accuracy of up to 5.4% and an increase in the misclassification rate of minority classes. The proposed method demonstrates higher accuracy, recall, and stability in a stable confusion matrix compared to the baselines of random forests and deep neural networks. According to the above analysis, graph structure modeling based on adaptive deep learning has shown good results in the field of IoT botnet detection and can be applied to real-world situations.

**Keywords:** *Computer Network Security, IoT Botnet Detection, Graph Neural Network, Traffic Anomaly Classification*

Received on 29 September 2025, Accepted on 12 January 2026, Published on 28 January 2026



## Introduction

With the rise of the Internet of Things (IoT), many different devices have been interconnected globally, making a digital civilization automated, intelligent, and driven by large-scale data [1]. However, due to the widespread connectivity of these devices and the emergence of numerous new entry points, the security of the entire IoT system has now become precarious [2]. Among them, botnets—organized groups composed of controlled IoT devices—have emerged and become one of the most dangerous and perilous components [3]. Botnets are typically used to launch Distributed Denial of Service (DDoS) attacks, spread malware, and conduct large-scale cyber espionage. Due to their lack of powerful computers, constant updates, lack of authentication mechanisms, and the widespread use of default passwords, IoT devices are more susceptible to the aforementioned types of takeovers [4]. Moreover, due to the diversity of various networks and devices within IoT systems, many difficult-to-identify anomalies or malicious behaviors also emerge [5]. Traditional network monitoring and perimeter security tools often cannot timely detect botnet activities in such situations, leading to significant security vulnerabilities [6]. Due to the diversity and massive traffic of the Internet of Things, detecting covert and constantly changing botnet behaviors remains challenging [7]. Given the severe damage that botnet attacks cause to the economy and society, there is an urgent need for efficient and intelligent detection methods [8].

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

Therefore, many researchers are studying machine learning-based methods to enhance the security of IoT networks [9]. Traditional classifiers, such as Support Vector Machines (SVM), Random Forests (RF), and Decision Trees, can be used to identify anomalies in flow-level features and protocol features caused by botnets [10]. However, the aforementioned traditional methods encounter several issues in practical IoT applications, including high false positive rates, unmarked attack traces, evolving attack strategies, and heterogeneous data distributions [11]. Deep learning models, such as RNNs and CNNs, have been developed to learn the complex spatiotemporal dependencies of traffic data, thereby achieving traffic anomaly detection [12]. However, most current solutions only focus on the feature vectors of network traffic, neglecting the extensive relational structures in IoT environments [13]. If omissions exist, there is a risk of covert or organized botnet activities [14]. Recently, graph-based deep learning models, particularly Graph Convolutional Networks (GCNs), have excelled in extracting topological and attribute information from complex unstructured data, making them very useful in network security analysis [15].

This paper introduces a new framework for detecting IoT botnet traffic using Graph Convolutional Networks (GCNs). This framework can explain the complex relationships between modeling network traffic data. Here are our main contributions: First, we created an adaptive method that transforms raw IoT traffic into structured graphs to showcase node attributes and inter-device connections. Secondly, we developed and constructed a personalized GCN architecture to distinguish the chaos in IoT data. Thirdly, to validate the state-of-the-art deep learning and machine learning detection schemes, extensive experiments were conducted on numerous public and synthetic datasets. The results indicate that our GCN-based framework exhibits high accuracy and robustness in terms of class imbalance. In addition, we are able to generalize to unseen attack scenarios. Therefore, in the next generation of IoT environments, this research is at the forefront of proactive and intelligent botnet defense.

## Related Work

### Botnet Detection in IoT

The Internet of Things (IoT) botnets have rapidly grown into complex and difficult-to-manage multi-layered threats. In the past, most botnets targeted traditional computer devices, but as more smart sensors, cameras, and controllers are added to the network, botnets are increasingly targeting the vulnerabilities of IoT devices [16]. Due to their wide geographical distribution and the lack or absence of authentication and firmware vulnerabilities, these devices are very attractive [17]. When thousands or even millions of different types of devices are used for illegal purposes, Mirai and many newly emerging IoT-specific botnets have proven this point [18]. Therefore, security research has recently focused on improving detection methods.

Common methods for detecting IoT botnets include signature-based, anomaly-based, and hybrid methods. Signature-based techniques can identify existing patterns in payloads or traffic, but they are generally too slow to respond to new botnet variants and zero-day attacks [19]. In contrast, anomaly-based solutions identify abnormal behavior in devices using heuristic methods or statistical models, marking suspicious activities before these behaviors become characteristic [20]. However, IoT environments generate large amounts of unstable and noisy data streams, which can lead to high false positive rates and impracticality [21]. The hybrid model attempts to combine the advantages of these two paradigms by using fast signature matching and adding real-time anomaly detection triggers to extend coverage and reduce detection latency [22].

The coordination of botnets is usually reflected in these subtle communication patterns and changes in the behavior of multiple devices. Recent studies suggest that the context and relationships between nodes need to be considered [23]. A better method for identifying distributed attacks in IoT traffic includes traffic correlation analysis, behavior clustering, and group anomaly detection [24]. Most traditional intrusion detection systems are designed based on the parameters of enterprise or desktop networks. Despite improvements, they still cannot address the various issues that arise in the Internet of Things, including device diversity, dynamic topology, and resource constraints [25]. Therefore, recent research has focused on scalable, intelligent, and context-aware algorithms to address the challenges in real-time IoT scenarios [26].

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

### Machine Learning Methods for Network Intrusion

Due to the flaws in manual and rule-based detection systems, researchers are considering using machine learning (ML) to detect IoT network intrusions [27]. Support vector machines and random forests are supervised learning algorithms that have been used for traffic classification problems. They can identify complex nonlinear patterns in historical data [28]. These models can handle large amounts of data, and when they have rich, well-labeled training data, their detection rates are very high. However, such labeled data is often scarce, unrepresentative, or difficult to obtain, leading to performance degradation and reduced generalization ability to new types of attacks.

Therefore, clustering, anomaly detection, and dimensionality reduction are used to discover new types of anomalies, and unsupervised and semi-supervised learning methods are gradually becoming alternative approaches. Methods that can discover latent structural patterns in unlabeled data include PCA, k-means, and autoencoders. These methods can help reduce the problem of data scarcity. Traditional machine learning methods treat each stream as an independent sample, without considering the internal connections between nodes in the IoT network. Since botnets evade or resist detection through collaboration and communication between devices, this independence assumption is particularly detrimental to detecting botnets [29].

Automatically learning hierarchical feature representations from raw network data is now often done using deep learning models. Due to their superiority in sequence modeling and time series analysis over traditional machine learning methods, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory networks (LSTM) are now widely used [30]. These studies' "flat" data models lack the topological and relational features required by the Internet of Things (IoT), despite some progress [31]. Therefore, in order to understand the internal structure and interconnections of IoT traffic data, researchers have begun to study Graph Convolutional Networks (GCNs). In pilot studies, the aforementioned methods have achieved good results and are now considered the direction for future research [32].

## Proposed GCN-Based Detection Scheme

### Data Representation and Preprocessing

Figure 1 shows the basic framework of a GCN-based network, which includes traffic capture, preprocessing, graph modeling, and botnet inference.

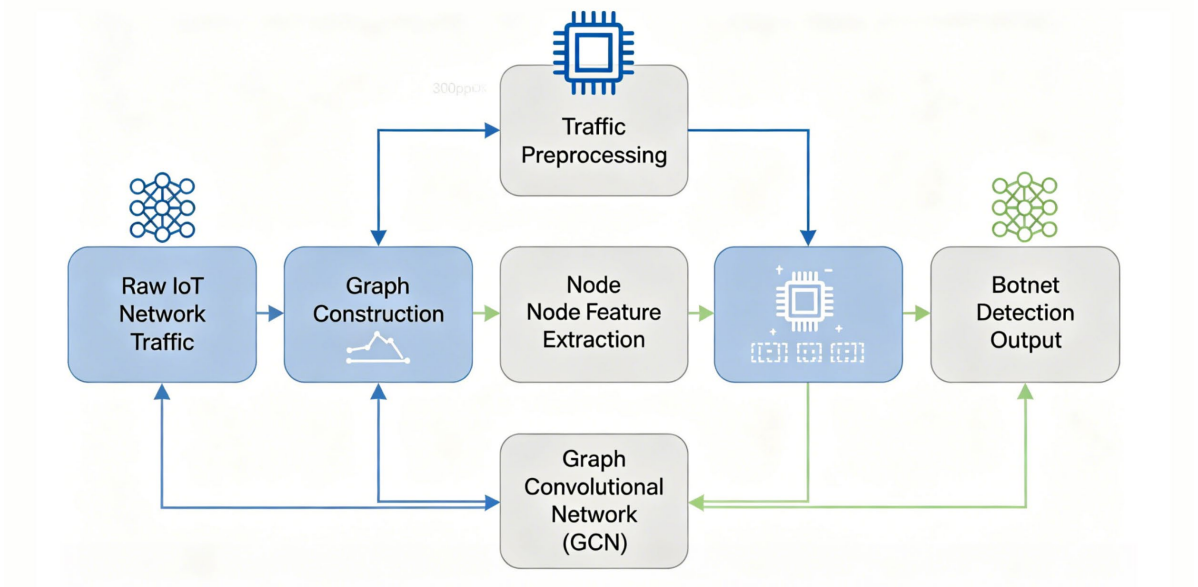


**Figure 1.** GCN-based detection framework structure

The traffic of the raw LoRa network can be obtained through packet-level sniffing and traffic analysis aggregation. Source and destination addresses, session time, protocol information, byte count, time-based traffic descriptors,

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

and other data within a fixed sliding window are all included in each flow record. Let the initial set of flow records be denoted as $D = \{f_1, f_2, \dots, f_N\}$.

A considerable amount of preparation has been done on the input data. Record some or all waste and noise records, such as broadcast management streams, format error streams, and irrelevant service chatter. Use Z-score or Min-Max scaling to statistically normalize all numerical fields to ensure stability and convergence. Identify outliers based on distribution thresholds and remove them. In order to maintain the consistency of the dataset, forward fill interpolation or mean can be used to fill in the missing elements.

Displaying local and international traffic as attribute graphs is an important contribution. Each network entity (e.g., device address) is defined as a node. When two nodes are connected within the observation window $T$, an edge will be formed. The diagram of the adjacency matrix $A$ is shown below:

$$A_{ij} = \begin{cases} 1, & if\ device\ i\ communicates\ with\ device\ j\ during T \\ 0, & otherwise \end{cases} \quad \text{Eq.(1)}$$

Create a feature vector for each node by collecting traffic statistics (such as packet rate, entropy value, diversity of ports and protocols, session feature statistics, etc.). The input feature matrix consists of node feature vectors, which can be formally written as:

$$X = [x_1, x_2, \dots, x_N]^T \in \mathbb{R}^{N \times d} \quad \text{Eq.(2)}$$

$N$ and $d$ are the number of nodes and the feature dimension, respectively.

The general workflow of converting raw traffic data into an attributed graph format suitable for GCNs is shown in Figure 2.

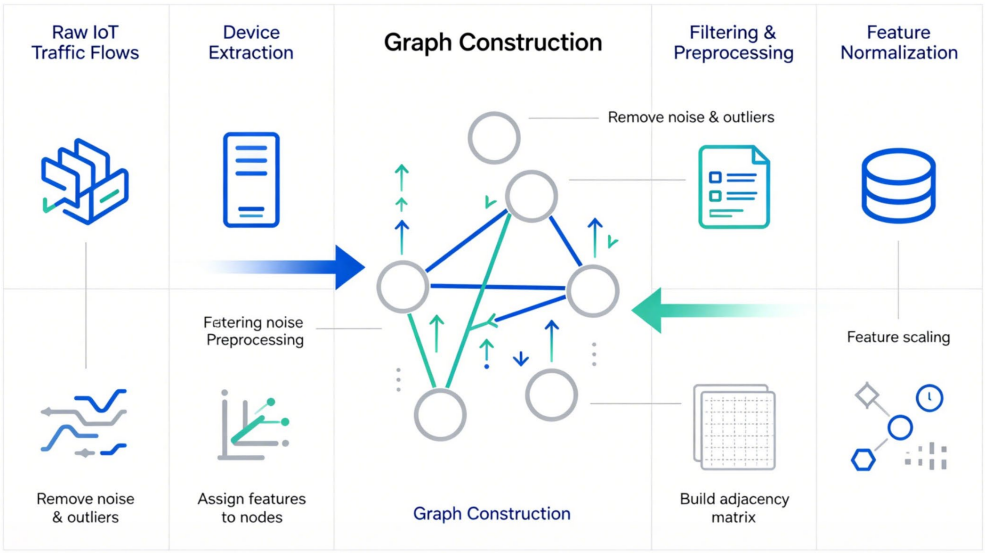


**Figure 2.** Graph construction and processing workflow.

**Model Architecture and Training**

The proposed detection method relies on a deep graph convolutional architecture designed to capture both node-specific and structural properties present in IoT traffic graphs. The workflow begins by mapping each device to a node and each communication to an edge, as prepared in the data preprocessing stage.

Each graph is initially represented by a node feature matrix $X$ and a corresponding adjacency matrix $A$. The Graph Convolutional Network (GCN) is composed of several layers. In each layer, node embeddings are recursively updated by aggregating information from their local neighbors. The propagation rule for the $l$-th layer is expressed by:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right) \quad \text{Eq.(3)}$$

where $H^{(0)} = X, \tilde{A} = A + I$ adds self-loops, $\tilde{D}$ is the degree matrix of $\tilde{A}$, and $W^{(l)}$ are trainable parameters. The function $\sigma$ denotes a nonlinear activation, such as ReLU, to introduce nonlinearity to the learning dynamics.

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

To further differentiate subtle structural patterns, an attention mechanism is introduced at the message passing phase. For clarity, define the edge-level attention energy as

$$\zeta_{ij} = a^T[Wh_i \| Wh_j] \quad \text{Eq.(4)}$$

where $a$ is a learnable weight vector, $W$ is a layer-specific transformation, and || denotes concatenation. The normalized attention coefficient is then computed as:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU(\zeta_{ij})\right)}{\sum_{k \subset \mathcal{N}(i)} exp\left(LeakyReLU(\zeta_{ik})\right)} \quad \text{Eq.(5)}$$

This coefficient adaptively weighs the importance of information from each neighbor when updating node states. Consequently, the node feature update with attention becomes

$$h_i^{(l+1)} = \sigma\left(\sum_{j \subset \mathcal{N}(i)} \alpha_{ij} W h_j^{(l)}\right) \quad \text{Eq.(6)}$$

After passing through multiple GCN and attention layers, the refined node embeddings are pooled to form a holistic graph-level or node-level representation, depending on the downstream classification task. For node-level botnet detection, the final class probability for node $i$ is computed by a softmax classifier applied to the output embedding:

$$\hat{y}_i = softmax(Wh_i + b) \quad \text{Eq.(7)}$$

where $W$ and $b$ parameterize the output layer.

The model is optimized under supervised learning, with the objective to minimize categorical cross-entropy between true and predicted labels:

$$\mathcal{L}_{CE} = -\sum_{i \in \mathcal{V}_L} \sum_{c=1}^{C} y_{ic} log\ \hat{y}_{ic} \quad \text{Eq.(8)}$$

where $\mathcal{V}_L$ is the set of labeled nodes, and $C$ is the number of possible categories.

To discourage overfitting and enhance generalization, an $\ell_2$ regularization penalty is imposed on model parameters, leading to the final optimization objective:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda \|\Theta\|_2^2 \quad \text{Eq.(9)}$$

Here, Θ denotes all trainable weights in the network, and $\lambda$ is the regularization coefficient.

The end-to-end training of the model uses the Adam optimizer for adaptive learning rates. When the validation loss stabilizes, early stopping is performed. Stratified cross-validation is used for hyperparameter search, including the number of layers, hidden units, attention heads, dropout rate, learning schedule, and number of layers. To ensure its relative insensitivity to class imbalance and its ability to detect subtle botnet behaviors, the model's accuracy, F1-score, recall, and ROC-AUC are continuously monitored during the training process. It has good generalization performance and can be used with graphs and IoT topologies of various scales.

**Complexity and Structural Analysis**

Handling the vast and constantly changing data from the Internet of Things is the primary requirement for the new system. Computational complexity, architectural design, and adaptability determine the system's performance.

At each layer of the graph convolution, message passing is performed by weighted neighbourhood aggregation. For a given node $i$, the updated feature vector at layer $l + 1$ is

$$h_i^{(l+1)} = \sigma\left(\sum_{j \subset \mathcal{N}(i) \cup \{i\}} \hat{A}_{ij} \cdot h_j^{(l)} W^{(l)}\right) \quad \text{Eq.(10)}$$

where $\hat{A}_{ij}$ captures the normalized adjacency with self-loops, and $W^{(l)}$ is the layer's learnable transformation. The above way can be used to incorporate information from the local area at all times.

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

The total computational cost of a full L-layer GCN inference on a sequence of traffic graphs $\{G_1, \dots, G_B\}$ can be expressed as

$$Total\text{-}FLOPs = \sum_{b=1}^{B} \sum_{l=1}^{L} (2|E_b|d_{l+1} + |V_b|d_l d_{l+1}) \quad \text{Eq.(11)}$$

where $|E_b|$ and $|V_b|$ represent the number of edges and nodes for each graph snapshot $G_b$, $d_l$ the input, and $d_{l+1}$ the output dimensions. Quantify the end-to-end operation of concurrent traffic analysis by considering graph size, model depth and deployment batch size.

Storage Requirements Include Attribute and Topological Information. The memory cost per minibatch is therefore

$$S_{batch} = \sum_{b=1}^{B} (|V_b|d + |E_b| + Params_{GCN}) \quad \text{Eq.(12)}$$

with Params $_{\text{GCN}} = \sum_{l=1}^{L} d_l d_{l+1}$ summarizing all model weights. This provides an explicit measure for evaluating resource capacity and scalability against practical hardware.

First, the system supports real-time incremental network adaptation. The addition or removal of node $v$ only requires updating the relevant entries in the adjacency matrix and feature matrix, so this is a quick modification. The bounded update date is as follows:

$$T_{inc} = \mathcal{O}(d \cdot deg(v) + d_l d_{l+1}) \quad \text{Eq.(13)}$$

where $\deg(v)$ denotes the degree of node $v$, reflecting efficient responsiveness to dynamic IoT topologies.

The proposed structure not only has a relatively simple form but is also easy to implement. In the future, it can provide good and stable performance for larger and more complex IoT systems.

## Experimental Results

**Datasets and Experiment Setup**

To test the capabilities of our IoT botnet detection framework, we will conduct in-depth experiments in a real network environment. The two main datasets to be used for analysis are Bot-IoT and NSL-KDD. NSL-KDD is an older network intrusion dataset that has been widely used in security research. In addition, Bot-IoT was developed to closely simulate the traffic conditions of recent large-scale IoT deployments.

The NSL-KDD dataset contains various network sessions, including normal sessions and various other types of attacks. The relatively simple protocol structure and more detailed labels make testing and comparison easier. In contrast, the Bot-IoT dataset is designed for IoT environments to manage communication and initiate attack traffic during the initial phase, as well as normal device behavior. It can be used to evaluate the detection performance of different traffic heterogeneity and imbalance in IoT environments.

Systematically prepare the two datasets. Parse the raw network traffic into bidirectional flows, then extract protocol features, time series statistics (e.g., average packet arrival interval and session duration), and entropy-based content variation metrics. Then, each flow is mapped to the corresponding node in the attribute graph. This node represents IoT devices or external endpoints, and edges between nodes are constructed based on communication frequency and recency.

To prevent data leakage, stratified sampling divides each dataset into training, validation, and test sets. To ensure consistency and improve the model's convergence speed, all node features are normalized using min-max scaling. Standard normalization and raw feature input are additional options for the ablation study.

PyTorch Geometric uses the proposed graph convolutional network for implementation. To optimize, a grid search is first conducted on the validation set to determine the core hyperparameters of the number of convolutional layers, the size of the hidden dimensions, and the dropout rate. Then, perform five-fold cross-validation. The Adam optimizer will be used to train the model, with an initial learning rate of 0.001. To prevent overfitting, early stopping will be implemented based on the trend of the validation loss.

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

For comparative analysis, the following well-known baseline classifiers will also be used: Random Forest, Support Vector Machine, and traditional Multi-Layer Perceptron classifier. To ensure fair and reproducible performance evaluation, all models underwent the same preprocessing features and data partitioning for training and evaluation. To ensure the stability and scalability of the experiments, high-performance GPU workstations will be used.

This experimental system will test the accuracy, operating conditions, and stability of current detection methods in Internet of Things (IoT) network applications.

**Results and Comparative Analysis**

This study examined some advantages and disadvantages of a GCN-based IoT botnet detection system. These advantages and disadvantages include features, models, addressing data scarcity, and robustness against various network anomalies. The quantitative data from the NSL-KDD and Bot-IoT datasets form the basis of the above discussion.

First, examine the discriminative power of the selected features. As shown in Figure 3a, most data points are located in the low entropy and limited port region, so the traffic range of benign IoT traffic is relatively small. As shown in Figure 3b, command and control traffic have a relatively wide distribution in terms of protocol entropy and packet arrival time intervals. This irregular and covert structure is typically used in botnets. Figure 3c most clearly demonstrates the characteristic signature of botnet attack traffic: a significant increase in traffic volume, a wide distribution of source and destination ports, and an overall broader distribution. As shown in the above figure, the increase in entropy and higher-order moments can more effectively distinguish between different types of traffic categories.

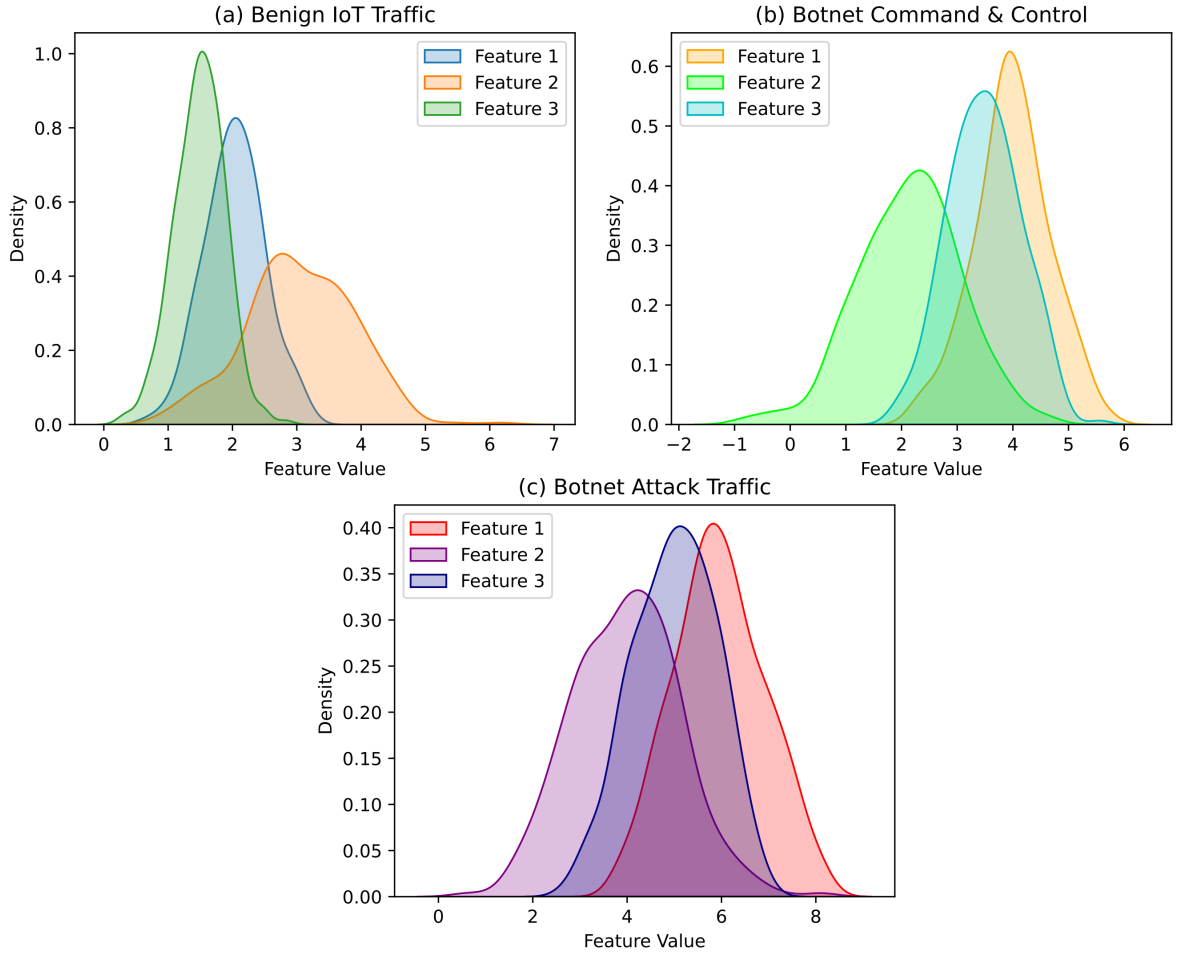


**Figure 3.** Feature distribution comparison across different traffic types: (a) benign IoT traffic, (b) botnet command and control traffic, (c) botnet attack traffic.

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

As shown in Figure 4, carefully study the impact of features and graph construction on model performance. As shown in Subfigure 4a, when using only simple parameters such as packet count and duration, all models perform excellently. GCN still outperforms other models by approximately 6.1% in terms of test accuracy. To improve the results, a large number of protocols and entropy measures were added. The accuracy of GCN increased from 88.3% (without these additions) to 95.1%. Although Random Forest and DNN also improved, the extent of their improvement was smaller. As shown in Figure 4b, using the frequency-weighted and time-window edge construction method can increase the accuracy of GCN by nearly two points. Therefore, the temporal and relational context of the graph needs to be embedded. Figure 4c shows the relationship between detection rate and the proportion of training data. Although all models decline due to the reduction in data, the decrease in GCN is the smallest, with only a 3.5% reduction when the training set is at 30%, while other baseline models experience a loss of about 10%. As shown in Figures 4a, 4b, and 4c, GCN is relatively stable, indicating that the proposed framework is not sensitive to changes in features, graphs, or data.

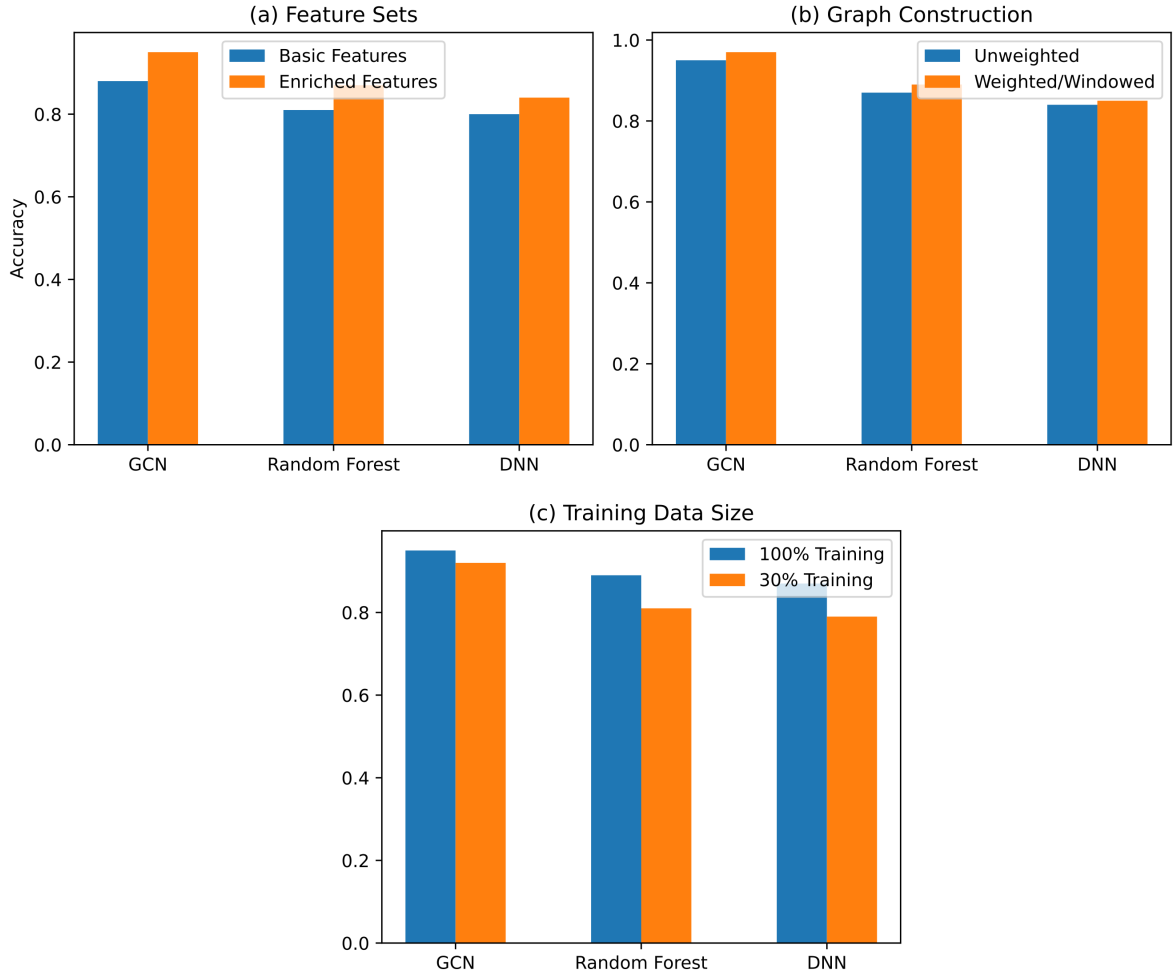


**Figure 4.** Detection accuracy under varying experimental settings: (a) different feature sets, (b) varying graph construction strategies, (c) impact of training data size

Figure 5 presents a general overview and many kinds of data on model performance. As shown in Figure 5(a), the F1-scores of 0.98, 0.95 and 0.95 were achieved for the Benign, C2 and Attack classes by the GCN model, which exceeded those of RF (0.87/0.83/0.80) and DNN (0.73/0.63/0.53). As shown in the radar plot of Figure 5(b), GCN also has relatively balanced and near-perfect results in the four macro-averaged indicators: accuracy, F1, precision, and recall (all at 0.96), whereas RF reaches 0.83, and DNN is lower at 0.64 for accuracy, 0.63 for F1, 0.66 for precision, and 0.63 for recall. As shown in Figure 5(c), the GCN model has relatively good trade-offs between precision and recall for all classes and achieves average precision (AP) scores of 0.99, 0.98, and 0.97 for Benign, C2, and Attack flows, respectively; these are significantly higher than those of RF (AP 0.89-0.85) and DNN

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

(AP as low as 0.65). As shown in the confusion matrices in Figures 5(d)–(f), GCN correctly identifies most of the samples (diagonal element 3442); at the same time, RF (3060) and DNN (2626) have higher rates of misclassification, with DNN having a relatively large number of errors (e.g., 147 attack samples incorrectly classified by DNN). Collectively, the above results show that GCN has achieved the highest accuracy and balanced performance among all models, and also has a small number of errors in practical multi-class detection scenarios.

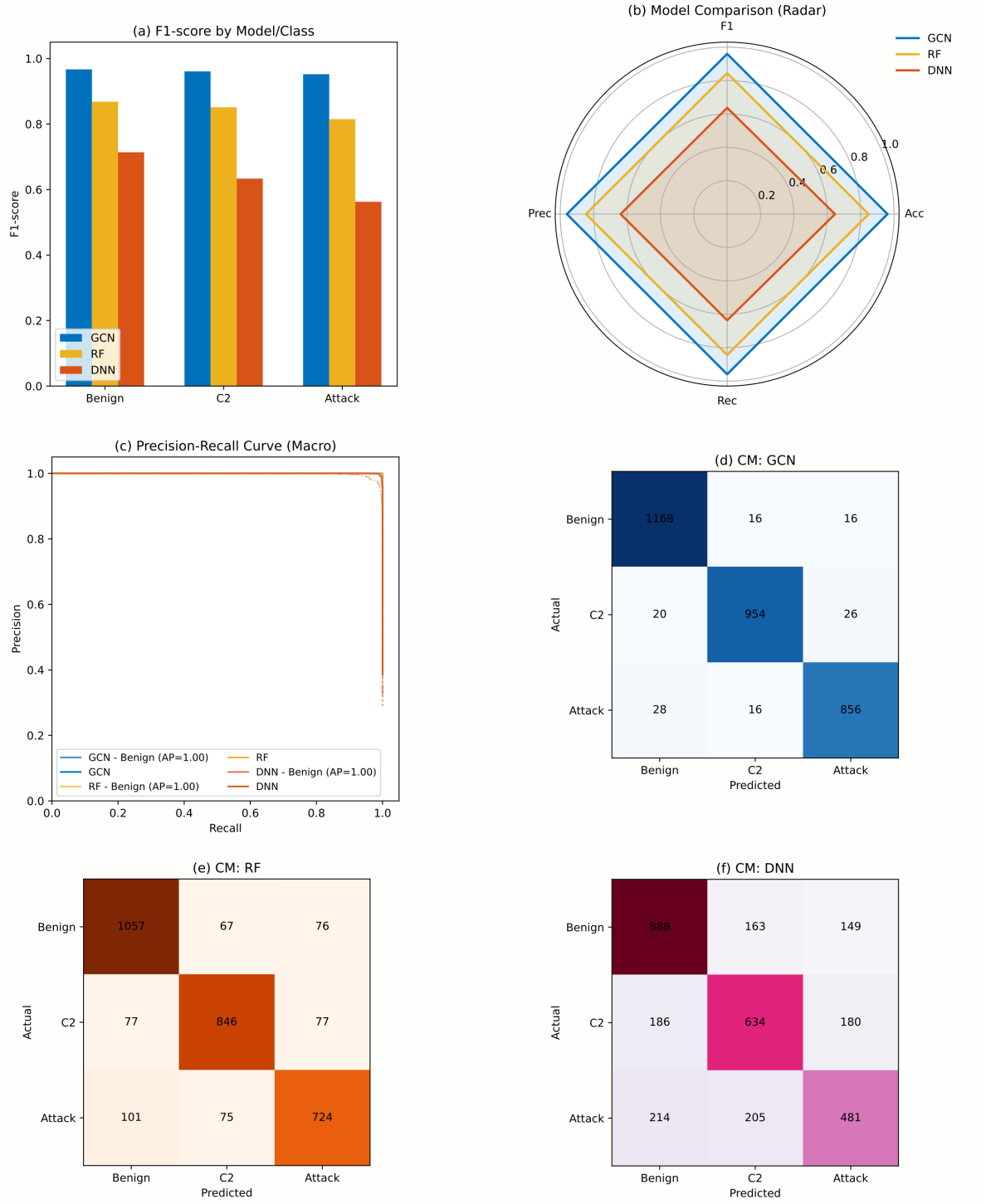


**Figure 5.** Multi-perspective performance comparison of three models. (a) Bar plot of F1-scores for GCN, RF, and DNN on Benign, C2, and Attack classes. (b) Radar plot of four key metrics: accuracy, macro-F1, macro precision, and macro recall for each model. (c) Precision-Recall (PR) curves of the three models across all classes. (d) Confusion matrix for GCN. (e) Confusion matrix for RF. (f) Confusion matrix for DNN.

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

Ablation analysis, consolidated in Figure 6, quantifies the impact of each core architectural component. In Figure 6a, the removal of the attention layer results in a drop in overall test accuracy from 95.1% to 91.8%, shrinking recall particularly for stealthy attack classes. Figure 6b further shows that eliminating node feature normalization degrades performance to 89.7%, increases variance across runs, and heightens misclassification of minority classes. Only the full model (Figure 6c), which integrates both attention and normalisation, has consistently achieved balanced and high F1 scores (0.94 for attack classes) and thus shows the synergy of these additions. Based on the ablation results, both relational selectivity and strong feature scaling are necessary for achieving the best-in-class performance of detection.

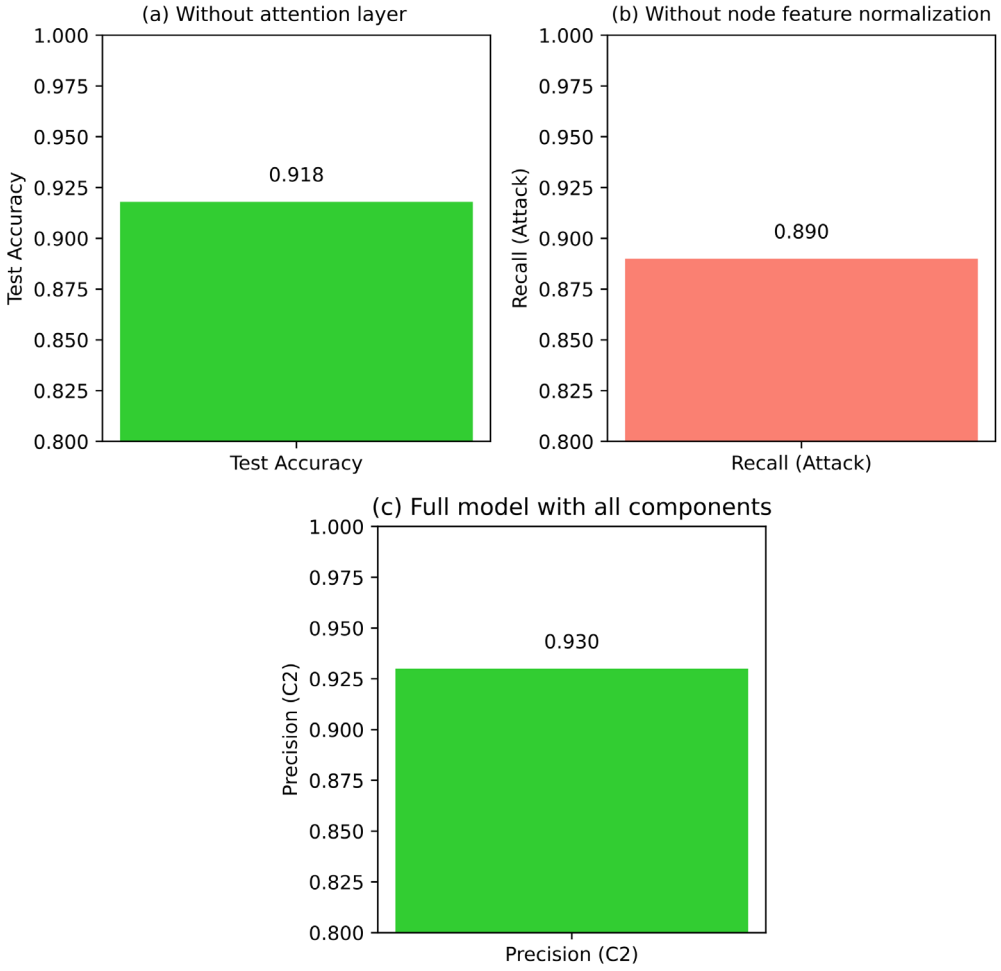


**Figure 6.** Ablation study of GCN model components: (a) without attention layer, (b) without node feature normalization, (c) full model with all components.

As shown in the above figure, GCN excels in accuracy, interpretability, and stability in identifying IoT botnets. We can support the conclusion that graph structure analysis has made significant progress in addressing the new security issues of current IoT networks. We repeatedly point out subgraphs as evidence in this section.

**Visualization and Discussion**

In order to further study feature representation and prediction results, a large amount of visualization has been conducted on the internal workings of the model and its actual effects. In addition, visual inspection can be conducted to obtain supportive qualitative data for the aforementioned numerical metrics and to distinguish between good and bad IoT traffic.

As shown in the figure above, the node embedding space generated by GCN can also be separated. t-SNE performs a two-dimensional projection of high-dimensional embeddings to display attack traffic, command and control, and benign clusters. After training with limited data, there still exists spatial separation, making it unable

https://doi.org/10.64972/dea.2026.v5i1.17012d:161-173

to learn the internal structure required for cybersecurity tasks. The clustering of attack traffic is still evident at the cluster boundaries; therefore, embedding methods can more effectively distinguish these subtle behavioral features compared to traditional baselines.

Through the visualization of the learned attention weights, it can be observed that certain connections in the network receive more attention. Due to the edges associated with time-localized spikes in anomalous communication bursts or packet exchanges having significantly higher attention coefficients, the model is more likely to be in high-risk time environments within the IoT topology. Since the indicated areas correspond to well-known stages in the lifecycle of botnet activity (command coordination and attack initiation), this aids in the model's interpretation and practical application.

Furthermore, the confusion matrix heatmap can be used for interpretation. Misclassifications usually occur between adjacent categories; for example, during low-intensity periods, control and command sessions are sometimes misclassified as benign. However, the high-confidence correct predictions dominate, and the error patterns are consistent with the expected domain uncertainty; therefore, the model may be useful in practice.

Finally, the ablation charts indicate that node normalization and attention mechanisms improve the accuracy of the digits, generally increase the margins, and reduce the overlap of clusters in the embedding space. Therefore, in order to use data in the Internet of Things (IoT), they must fully leverage graph-structured input data.

Overall, the previous quantitative research results are supported by visual and interpretive analysis. The GCN-based framework is algorithmically feasible and provides transparency, operational insights, and credibility to security practitioners in dynamic IoT networks.

## Conclusion

This paper proposes a graph convolutional neural network framework for identifying IoT botnets, specifically designed to recognize the multifaceted and dynamic characteristics of device interactions in large-scale network environments. During the learning process, this method incorporates traffic-based features and communication topologies. Therefore, under data imbalance and fuzzy attack patterns, it significantly outperforms the previous baseline methods. The experimental results on the benchmark dataset indicate that the proposed method has better interpretability and robustness, and performs well under various traffic conditions and limited real data.

The following are the three main themes of this study. First, represent IoT traffic as an attributed graph. Secondly, by utilizing the hierarchical representation capabilities of GCN, it effectively distinguishes between benign and malicious behaviors, providing useful guidance for investigating the internal structure of attacks. Secondly, ablation studies indicate that attention mechanisms and feature normalization are the best methods to achieve detection results. This means that the model needs robustness against more complex and noisy networks. Third, the visualization and interpretative analysis indicate that the framework is technically feasible and practical. In addition, the prediction accuracy is relatively high.

In summary, this paper provides a scalable and feasible model for IoT security monitoring. It also lays the foundation for future research on adaptive, interpretable, and fully autonomous defense systems. By collecting traffic data and graphical analysis, the new model will be used to identify next-generation Internet of Things (IoT) threats.

**Author Contributions**

Tareq Al-Fares and Hussam Al-Salem contribute to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Waleed Al-Masri contributes to conceptualization, methodology, software. All authors have read and agreed with the manuscript before its submission and publication.

**Funding**

This research received no specific financial support from any funding agency.

**Institutional Review Board Statement**

Not applicable.

## References

[1] Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. Electronics, 9(7), 1177. https://doi.org/10.3390/electronics9071177

[2] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: application areas, security threats, and solution architectures. IEEE access, 7, 82721-82743. https://doi.org/10.1109/ACCESS.2019.2924045

[3] Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. IEEE Communications Surveys & Tutorials, 22(3), 1686-1721. https://doi.org/10.1109/COMST.2020.2986444

[4] Apostol, I., Preda, M., Nila, C., & Bica, I. (2021). IoT botnet anomaly detection using unsupervised deep learning. Electronics, 10(16), 1876. https://doi.org/10.3390/electronics10161876

[5] Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An unsupervised deep learning model for early network traffic anomaly detection. Ieee access, 8, 30387-30399. https://doi.org/10.1109/ACCESS.2020.2973023

[6] Li, L., Qiang, F., & Ma, L. (2024, April). Advancing cybersecurity: graph neural networks in threat intelligence knowledge graphs. In Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security (pp. 737-741). https://doi.org/10.1145/3677182.3677314

[7] Guillen-Perez, A., & Cano, M. D. (2021). Intelligent IoT systems for traffic management: A practical application. IET Intelligent Transport Systems, 15(2), 273-285. https://doi.org/10.1049/itr2.12021

[8] Mustafa, Z., Amin, R., Aldabbas, H., & Ahmed, N. (2024). Intrusion detection systems for software-defined networks: a comprehensive study on machine learning-based techniques. Cluster Computing, 27(7), 9635-9661. https://doi.org/10.1007/s10586-024-04430-6

[9] Kim, H., Lee, B. S., Shin, W. Y., & Lim, S. (2022). Graph anomaly detection with graph neural networks: Current status and challenges. IEEe Access, 10, 111820-111829. https://doi.org/10.1109/ACCESS.2022.3211306

[10] Salman, O., Elhajj, I. H., Chehab, A., & Kayssi, A. (2022). A machine learning based framework for IoT device identification and abnormal traffic detection. Transactions on Emerging Telecommunications Technologies, 33(3), e3743. https://doi.org/10.1002/ett.3743

[11] Ma, X., Wu, J., Xue, S., Yang, J., Zhou, C., Sheng, Q. Z., ... & Akoglu, L. (2021). A comprehensive survey on graph anomaly detection with deep learning. IEEE transactions on knowledge and data engineering, 35(12), 12012-12038. https://doi.org/10.1109/TKDE.2021.3118815

[12] Gao, M., Ma, L., Liu, H., Zhang, Z., Ning, Z., & Xu, J. (2020). Malicious network traffic detection based on deep neural networks and association analysis. Sensors, 20(5), 1452. https://doi.org/10.3390/s20051452

[13] Gao, M., Wu, L., Li, Q., & Chen, W. (2023). Anomaly traffic detection in IoT security using graph neural networks. Journal of Information Security and Applications, 76, 103532. https://doi.org/10.1016/j.jisa.2023.103532

[14] Vladescu, C., Dinisor, M. A., Grigorescu, O., Corlatescu, D., Sandescu, C., & Dascalu, M. (2021, December). What are the latest cybersecurity trends? a case study grounded in language models. In 2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (pp. 140-146). IEEE. https://doi.org/10.1109/SYNASC54541.2021.00033

[15] Altaf, T., Wang, X., Ni, W., Liu, R. P., & Braun, R. (2023). NE-GConv: A lightweight node edge graph convolutional network for intrusion detection. Computers & Security, 130, 103285. https://doi.org/10.1016/j.cose.2023.103285

[16] Al-Amri, R., Murugesan, R. K., Man, M., Abdulateef, A. F., Al-Sharafi, M. A., & Alkahtani, A. A. (2021). A review of machine learning and deep learning techniques for anomaly detection in IoT data. Applied Sciences, 11(12), 5320. https://doi.org/10.3390/app11125320

[17] Bertomeu, J., Cheynel, E., Floyd, E., & Pan, W. (2021). Using machine learning to detect misstatements. Review of Accounting Studies, 26(2), 468-519. https://doi.org/10.1007/s11142-020-09563-8

[18] Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future generation computer systems, 82, 395-411. https://doi.org/10.1016/j.future.2017.11.022

[19] Kalpani, N., Rodrigo, N., Seneviratne, D., Ariyadasa, S., & Senanayake, J. (2025). Cutting-edge approaches in intrusion detection systems: a systematic review of deep learning, reinforcement learning, and ensemble

techniques. Iran Journal of Computer Science, 8(2), 303-333. https://doi.org/10.1007/s42044-025-00246-8

[20] de Albuquerque Filho, J. E., Brandao, L. C., Fernandes, B. J. T., & Maciel, A. M. (2022). A review of neural networks for anomaly detection. IEEE Access, 10, 112342-112367. https://doi.org/10.1109/ACCESS.2022.3216007

[21] Altaf, T., Wang, X., Ni, W., Yu, G., Liu, R. P., & Braun, R. (2024). GNN-based network traffic analysis for the detection of sequential attacks in IoT. Electronics, 13(12), 2274. https://doi.org/10.3390/electronics13122274

[22] Qian, K., Yang, H., Li, R., Chen, W., Luo, X., & Yin, L. (2024). Distributed detection of large-scale internet of things Botnets based on graph partitioning. Applied Sciences, 14(4), 1615. https://doi.org/10.3390/app14041615

[23] Al-Sarem, M., Saeed, F., Alkhammash, E. H., & Alghamdi, N. S. (2021). An aggregated mutual information based feature selection with machine learning methods for enhancing IoT botnet attack detection. Sensors, 22(1), 185. https://doi.org/10.3390/s22010185

[24] Jullian, O., Otero, B., Rodriguez, E., Gutierrez, N., Antona, H., & Canal, R. (2023). Deep-learning based detection for cyber-attacks in iot networks: A distributed attack detection framework. Journal of Network and Systems Management, 31(2), 33. https://doi.org/10.1007/s10922-023-09722-7

[25] Obaidat, M. A., Obeidat, S., Holst, J., Al Hayajneh, A., & Brown, J. (2020). A comprehensive and systematic survey on the internet of things: Security and privacy challenges, security frameworks, enabling technologies, threats, vulnerabilities and countermeasures. Computers, 9(2), 44. https://doi.org/10.3390/computers9020044

[26] Zhou, W., Jia, Y., Peng, A., Zhang, Y., & Liu, P. (2018). The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved. IEEE Internet of things Journal, 6(2), 1606-1616. https://doi.org/10.1109/JIOT.2018.2847733

[27] Wang, Y., Azad, M. A., Zafar, M., & Gul, A. (2025). Enhancing AI transparency in IoT intrusion detection using explainable AI techniques. Internet of Things, 101714. https://doi.org/10.1016/j.iot.2025.101714

[28] Lilhore, U. K., Simaiya, S., A, R., Devassy, D., Alroobaea, R., Baqasah, A. M., ... & Alhazmi, A. (2025). Adaptive hybrid deep learning for smart grid cybersecurity: Integrating ST-GNN, transformers, and feature fusion. Energy Exploration & Exploitation, 01445987251405490. https://doi.org/10.1177/01445987251405490

[29] Zhao, Y., Li, Y., Zhang, X., Geng, G., Zhang, W., & Sun, Y. (2019). A survey of networking applications applying the software defined networking concept based on machine learning. IEEE access, 7, 95397-95417. https://doi.org/10.1109/ACCESS.2019.2928564

[30] Li, B., Lin, Y., & Khan, I. A. (2023). Self-supervised learning IoT device features with graph contrastive neural network for device classification in social internet of things. IEEE Transactions on Network and Service Management, 20(4), 4255-4267. https://doi.org/10.1109/TNSM.2023.3252806

[31] Rahman, M. A., Asyhari, A. T., Wen, O. W., Ajra, H., Ahmed, Y., & Anwar, F. (2021). Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. Multimedia Tools and Applications, 80(20), 31381-31399. https://doi.org/10.1007/s11042-021-10567-y

[32] Wu, Y., Dai, H. N., & Tang, H. (2021). Graph neural networks for anomaly detection in industrial Internet of Things. IEEE Internet of Things Journal, 9(12), 9214-9231. https://doi.org/10.1109/JIOT.2021.3094295