Electronics, Communications, and Computing Summit
Vol. 3, No. 2, Apr - Jun 2025, pp. 40-48
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/03.02.05

Graph-Based Machine Learning for Anomaly Detection in IoT Security

Andrés Rivera[1], José Uribe[2]

[1]Facultad de Ingenieria Universidad Andres Bello, Santiago, Chile, Email: riv.andres@unab.cl
[2]Facultad de Ingenieria Universidad Andres Bello, Santiago, Chile, Email: jose.ur@unab.cl

Article Info

Article history:

Received : 15.04.2025
Revised : 17.05.2025
Accepted : 19.06.2025

Keywords:

Internet of Things (IoT),
Anomaly Detection,
Graph Neural Networks (GNNs),
Cybersecurity,
Machine Learning,
Dynamic Graphs,
Intrusion Detection Systems (IDS),
Edge Computing,
Temporal Graph Learning

ABSTRACT

The exploding number of Internet of Things (IoT) devices in critical infrastructure, smart homes and industrial systems has presented the complex security challenges since the devices are heterogeneous, have limited resources and are highly interconnected. Conventional anomaly detection methods, such as rule system and signature-based systems, are not easily adjusted to highly dynamic and advanced attack vectors particularly on environments at large scale and decently distributed environments. In this paper, we present a graph-based machine learning model that would use the existing relational nature of the communications among devices in an IoT ecosystem to further identify unusual activities in the IoT ecosystem. Particularly, IoT networks can be conceptualized as dynamic graphs that use individual devices as nodes and communication events as edges with temporal and contextual metadata associated with these communication events. To learn both spatial interactions and time-varying dynamics of networks behavior the proposed framework uses Graph Neural Networks (GNNs) combined with recurrent temporal block which can be LSTM or GRU. This allows the model to find not only unchanging or point anomalies, but also more complicated patterns that relate to stealthy, evolving threats. To put critical interactions first and further make this model interpretable, an edge-level attention mechanism is further introduced. The validity of the framework is proved by publicly available datasets, including BoT-IoT and TON_IoT, showing a high detection accuracy and limited false-positive accuracies as well as an effective computational performance flexible enough to real-time usage. The findings indicate the generalizability of the framework in different topologies of IoT and protocols and performs better than a variety of baseline and state-of-the-art machine learning models. What is more, the system can work in edge and fog computing as well because of its lightweight nature and the ability to make decisions locally. On balance, the proposed study makes an advancement in the current state of art of IoT security through the creation of a scalable, dynamic, and explainable graph-based machine learning anomaly detection technique as a robust protection mechanism against cybersecurity threats in a next-generation smart setting.

1. INTRODUCTION

Internet of Things (IoT) has become a revolutionary phenomenon that allows close interconnection of a number of devices, which are currently used at home, as wearable gadgets, in industries, and medical conditions. With the deep integration of the IoT technologies into our everyday routine and into the most important infrastructures, they not only bring the benefits of convenience and automation but also embed serious cybersecurity vulnerabilities. The heterogeneous environment of IoT is defined as the wide range of possible devices, communication protocols, and scales of deployment which result in the broad attack surface with expanding opportunities to find a vulnerability in it and leverage it by adversaries. IoT systems may be severely harmed by security breaches with serious outcomes such as theft of data, interference with operations, and even harm to a person.

Traditional anomaly detection methods which include signature based IDS, statistical profiling, or even threshold based rule, do not quite work in the IoT setup. The solution does this with fixed patterns or known behavioral thresholds which do not allow the opportunities to track new stealthy

or developing cyber-attacks. In addition, IoT networks are characterized by very distributed networks and dynamic behavior patterns where the classic-central security apparatus struggles to work. Consequently, cutting-edge and dynamic security frameworks that are smart, flexible and expandable need to be developed to capture, process and react to weird procedures in real time.
A promising way to overcome these challenges is the graph-based machine learning. Specifically, graph-based methods allow a natural representation of the assemblages of complex and dynamically shifting networks produced by IoT networks, since nodes in a graph correspond to the elements of a network (in this case the IoT devices), and the edges between them correspond to the communication or transmission of data between the elements. Such graphical implementations allow the tracking of structural links and behavioral interdependencies that are usually an important indicator of cyber anomalies. Specifically, based on Graph Neural Networks (GNNs), there is a high potential in learning on structured data in numerous fields since they are able to capture high-level patterns on both graph topology and node-level features.
In this paper we introduce a new anomaly detection model, which uses temporal analysis and a graph based learning framework to identify malicious behaviour in IoT environments. Combining spatial modeling through GNNs with temporal sequence learning with recurrent neural networks our method has the potential to detect instantaneous and time-historical anomalies as well as any emerging threats. It is a model trained and validated using benchmark IoT security datasets that are made to be deployed in real-time edge-enabled applications. We will provide a voice on creating a strong, scalable explainable intrusion detection system that fits the specific challenges of contemporary IoT ecosystems.

2. RELATED WORK

IoT security is one of the most popular areas of research with the amount of vulnerability surface caused by billions of smart devices forming interconnected networks constantly growing. One of the primary fields of concern in this domain is anomaly detection that refers to detecting the abnormal deviations of behavior that would be the possible signs of the cyber threat.

2.1 IoT Anomaly Detection

The most common forms of intrusion detection systems (IDS) that are used in IoT networks are normally classified as signature-based IDS or anomaly-based IDS. With signature-based IDSs, detection is based on the usage of priori patterns of known attacks resulting in high accuracy against known attacks yet they cannot detect zero-day or new attacks [1]. Anomaly-based ones, however, develop a model of normal device or traffic behavior and raise flags on abnormalities. In dynamic IoT settings these strategies are highly prone to false-positives [2], although they are more adaptive than other strategies. Statistical profiling, support vector machines and clustering have been used but they have a tendency to ignore inter-device communication patterns necessary to identify context-aware threats.

2.2 Intrusion Detection Graph-Based

In recent times graph-based methods have been found to show promise as alternate approaches to network intrusion detection. In these methods, the IoT system is described graphically where devices are represented by nodes and the connections/relationships between the devices are described by edges. Such representations inherently include topological and behavioral correlation that can represent malicious activity [3]. It has been proved that graph metrics (e.g., degree centrality, betweenness, clustering coefficient) are useful in detecting the presence of compromised nodes or anomalous sub graphs within the industrial IoT systems [4]. Nevertheless, the majority of graph-based approaches are based on the assumption of static graphs, which do not represent real-time pattern IoT traffic to the same degree.

2.3 Cybersecurity and Graph Neural Networks

Deep learning on graphs With the emergence of deep learning on graphs, Graph Neural Networks (GNNs) have been found to have the potential to learn the representation over complex and relational data structures. There are also recent applications of Graph Convolutional Networks (GCNs) [5] and Graph Attention Networks (GATs) [6] in cybersecurity, such as, botnet detection, malware classification, and network flow inspection, etc. The detection of malicious nodes was done with a GNN-based model to learn the contextual dependencies on a communication graph [7]. Although useful, these models can also be considered to be very rigid in that they neglect time in the evolution of devices behavior which is essential to detection stealthy or coordinated attacks.

2.4 Gap in the Research

Although there has been significant advancement, one major drawback still exists in the system of dynamic graph learning as a method of IoT security anomaly detection. The existing solutions are restricted to illustrations of communication patterns on stationary graphs or they do not consider time dependencies. Also, little has been done to report the multi-modality of IoT data (e.g. sensor readings, control signals, metadata) on the same graph learning platform. The paper fills this gap with a Spatio-temporal GNN-based framework which learns both topological structures and temporal relationships to detect anomalies in heterogenous IoT environment robustly, and in real-time.

Table 1. Summary of Related Work in IoT Anomaly Detection

Aspect	Key Techniques / Findings	Limitations
Anomaly Detection in IoT	Signature-based and anomaly-based IDS; traditional ML (SVM, clustering); models normal behavior to flag deviations	Poor detection of novel/zero-day attacks; high false positives; lacks context from device interactions
Graph-Based Intrusion Detection	Represents IoT as graphs; uses metrics like centrality and clustering coefficient to detect anomalies	Mostly static graphs; unable to reflect real-time behavior changes in IoT traffic
Graph Neural Networks in Cybersecurity	GCNs and GATs applied to botnet/malware detection; captures node dependencies and interaction context	Typically ignores temporal evolution; limited to static or single-snapshot graphs
Identified Research Gap	Need for dynamic graph learning and multi-modal integration (e.g., metadata, sensor streams); spatio-temporal modeling required	Few models handle both topological and temporal dynamics; limited explainability and adaptability

3. PROPOSED METHODOLOGY

3.1 Graph Construction

The most important action to consider graph-based machine learning in IoT security is the ability to construct meaningful graphs that capture real and specific communication and behavioral patterns of devices in the network. The performance of any anomaly detection graph-based system will be dependent on the manner of capturing of the interaction and characteristics involved in a possible security compromise.
The IoT network has been modelled in the proposed model as a dynamic directed graph, $\mathbf{G_t} = (\mathbf{V_t}, \mathbf{E_t})$where $\mathbf{V_t}$represents the set of vertices (node) and $\mathbf{E_t}$ represents the set of edges at time $\mathbf{t}$. The snapshot of the state of the network is taken every time, corresponding to a predetermined time interval, which allows learning in time.

Nodes (Vertices)

Every node $\mathbf{v} \in \mathbf{V_t}$belongs to an IoT device of the network. Such devices can consist of sensors (e.g. temperature, motion), actuators (e.g. relays, motors), edge gateways, routers, or any other connected computing units. Node attributes contain the static and dynamic information of the devices. Statically identified features would be device type, manufacturer or capability type whereas dynamically identify features would be: CPU utilisation, memory utilisation, authentication status and recent activity score.

Edges

Every edge $\mathbf{e} \in \mathbf{E_t}$represents a logical interaction (communication event) among two devices. These directed edges point to the flow of message between a source node and a destination node and are time-stamped in order to maintain a temporal order. The communications can be of the type of TCP/UDP sessions, HTTP requests, MQTT publish/subscribe, or the remote procedure calls (RPCs). This difference in weight can be to represent the strength or intensity of the interaction, by the frequency, by the volume, or by the protocol priority of the communicating path.

Node and Edge Features

In order to increase the learning capacity of the proposed anomaly detection model, the nodes and the edges on the graph representing the IoT network are classed together with a clearly defined feature vector whose elements are extracted using the network traffic logs and metadata. Such characteristics are designed to model important features of the device behavior and contexts of communication. Important features are packet data: packet size, source and destination ports, time-to-live (TTL) and protocol-specific traffic flags (e.g., SYN, ACK), which contain low-level traffic characteristics. Also, protocol type indicators (e.g. HTTP, MQTT, CoAP) will assist in defining the differences in communication semantics in varied IoT applications. Interaction intensity is recorded on a temporal basis through communication frequency and session duration, which allow a quick overview of long-term interaction intensity; earlier security-relevant interactions like authentication failures, dropped connections, or previously seen anomalies can be encoded in historical flags. All the features are normalized and appeared in the form of high-dimensional tensors, which can be consumed in the Graph Neural Network (GNN). The framework allows exploring temporal dependencies in the IoT network through modeling the topology as a time-evolving graph, which is required for detecting not only abrupt deviations but gradual changes in behavior, drifts. By examining in detail all the steps involved in creating such a graph, there will be a solid basis in which topological relationships and overall behavior of IoT devices will be well displayed and a robust anomaly detection will be possible through graph based learning.

Table 2. Summary of Node and Edge Feature Categories Used in IoT Graph Construction

Graph Component	**Feature Category**	**Example Features**
Node (Device)	Static	Device type, Manufacturer, Capability class
Node (Device)	Dynamic	CPU usage, Memory load, Auth status
Edge (Communication)	Structural	Direction, Protocol type (HTTP, MQTT, etc.)
Edge (Communication)	Temporal	Frequency, Session duration, Historical flags

3.2 Model Architecture

The aforementioned anomaly detection framework takes advantage of the spatio-temporal graph neural network (ST-GNN) model that is expected to learn the underlying structure dependence as well as the temporal patterns of the communication patterns in the IoT environment. Such hybrid topology allows the model not only to recognize the abnormal behavior, due to the network topology, but also, modify to the changes over time, reflecting both sudden and gradual deviations with regard to normal behavior.

Spatio Temporal Graph Neural Network (ST-GNN)

At the basics the model has a two-stage learning mechanism that integrates Graph Convolutional Networks (GCNs) and recurrent units; specifically Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs). GCN component does the job of extraction of spatial dependencies by combining data of the neighbors of a node in the graph. Although it enables the model to conceptualize the behaviors of local devices within the larger scope of the network, it is important to be able to identify coordinated or multi-stage attacks. Officially, GCN layers achieve updated node representations by aggregating features of their neighbours according to edge-weighted adjacency-matrices. These embeddings are an encoding of spatial context at any time point.

To accommodate temporal dependencies we include a recurrent neural network (RNN) layer that can operate on the sequence of graph snapshots over time. The node and edge features evolve over time and are captured in this RNN component, typically with LSTM or GRU units, and the model can learn time-dependent patterns, say, periodic behavior, session state transitions, or gradual compromise progression. The temporal module uses the spatial embeddings that are learned by the GCN and produces a temporally-sensitive embedding of nodes or edges.

Edge Attention Mechanism

Further, to maintain a greater concentration on the important communications in the model, we encompass an edge attention mechanism, which has been inspired by Graph Attention Networks (GATs). In this mechanism, edges are assigned dynamic weight (updated with passing of messages) depending on the relevancy of communication pattern between nodes. An example of this would be where there is an unexpectedly massive increase of traffic between two not currently associated nodes, this would be checked with increased weight to the weights. The scores on attention are calculated using a learnable operation that takes into account the node features, the edge features as well as the past connections.

Anomaly Scoring Function

After a computation of spatio-temporal embeddings, anomaly scoring functions are applied to both the nodes and edges. The score returned by this function corresponds to the probability that a given node or communication link is being abnormal. We do this by means of a feedforward neural network over the last embedding as input which gives a score that is based on probability or distances. During training, the model is trained both on supervised loss (cross-entropy when labels are known) and on unsupervised loss (e.g. contrastive or autoencoder-

style loss), resulting in good generalisation to unseen attacks. A decision threshold mechanism is then applied that categorises entities which are to be normal or anomalous.
Composed, in collaboration, this architecture creates an effective and responsive anomaly-detecting system to fuse structure understanding, time sensing and context prioritization- with the flexibility to work in real-time on heterogeneous and dynamic Internet of Things networks.

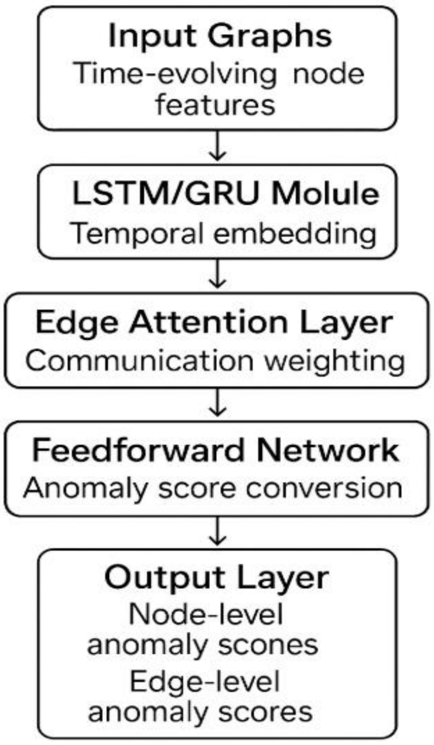

Figure 1. Architecture of the Proposed Spatio-Temporal Graph Neural Network (ST-GNN) for Anomaly Detection in IoT Networks

3.3 Training and Optimization
Specifically, training a state-of-the-art anomaly detection model in IoT-related settings becomes particularly challenge-prone given that labeled attack data is scarce, there are many dimensions to network interactions, and the behavior of different devices changes dynamically. In order to overcome such challenges, the framework takes the semi-supervised learning setting, learns a joint hybrid loss function with simultaneous emphasis on classification quality and on regularizing the structure, and uses scalable optimization methods that can be directly applied to large and time evolving graphs.

Semi-Supervised Learning
Real-world IoT systems would not reasonably be expected to have such large labeled datasets to use, particularly when the attack to be detected is novel or rare. As a solution to this shortcoming, the model is deployed under semi-supervised training, were a tiny percentage of nodes and edges are labeled as either benign or malicious. All the other majority are handled as unlabeled and they help in training by making neighborhood propagation and consistency regulations. This environment is advantageous to the GNN component in terms of leveraging the graph topology to spread the label information to the connected nodes, so that both labeled and contextual (unlabeled) data could be used in the learning process by the model. This increases generalization and allows identifying zero day or unknown threats.

Loss Function Design
To optimize the model effectively, we define a composite loss function that balances two objectives:

1. **Classification Loss**$(\mathcal{L}_{\text{cls}})$**:** On the labeled membership of nodes or edges, we would then use a cross-entropy loss tends to tune the model to be able to differentiate between the normal and the abnormal behavior according to the learned embeddings.
2. **Graph Structure Regularization**$(\mathcal{L}_{\text{reg}})$**:** Regularization is a generic term that refers to a technique applied to a formulation to obtain

a modified version of the formulation, with the aim of maximizing a particular property of the formulation: excessive flexibility in a regularization procedure is typically desirable, so this application of regularization is sometimes called a regularization method or regularization methodology. In statistical learning, the minimization of a statistical risk is often regularized, usually by adding a so-called regularization functional to the statistical risk in order to penalize excessively flexible solutions to the statistical learning task, with the goal of encouraging aLearned embeddings: we add a regularizer to the loss, e.g. Laplacian smoothing or contrastive loss to ensure that the representation learned has connections well suited to capture the underlying graph structure. This term punishes irregular representation of nearby nodes and favors smoothness in embedding space, which favours greater separation between anomalies.

The final loss is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda . \mathcal{L}_{\text{reg}}$$

Whereλ is a hyperparameter that balances the importance of structure preservation.

Scalability and Optimization Techniques

By the nature of the scale and complexity of real-time IoT graphs, the model makes use of mini-batch graph sampling strategies like Graphs AGE or neighbor sampling, to minimize the computational overhead in the course of training. The full graph is not processed and instead of that they sample localized sub graphs with the focus on target nodes dynamically, which enables them to train efficient edge or fog nodes which have limited resources.

Besides that, in order to deal with temporal evolution, we partition the data into sliding temporal windows, with each window reflecting a temporal snapshot of the graph. Does it compromise the evolving patterns or make navigation possible and facilitate online training, and ongoing learning? The answer is that temporal windowing does both. At every iteration, the model gets trained using sequences of temporal subgraphs, which allow the detection not only of instantaneous anomalous events but also long-term behavioral changes.

A combination of these training and optimization can make the proposed model robust, adaptive, as well as scalable in terms of being able to learn well and efficiently using sparse data, maintaining structural relationships and can be run within the constraints of real-time requirements as needed in applications involving IoT.

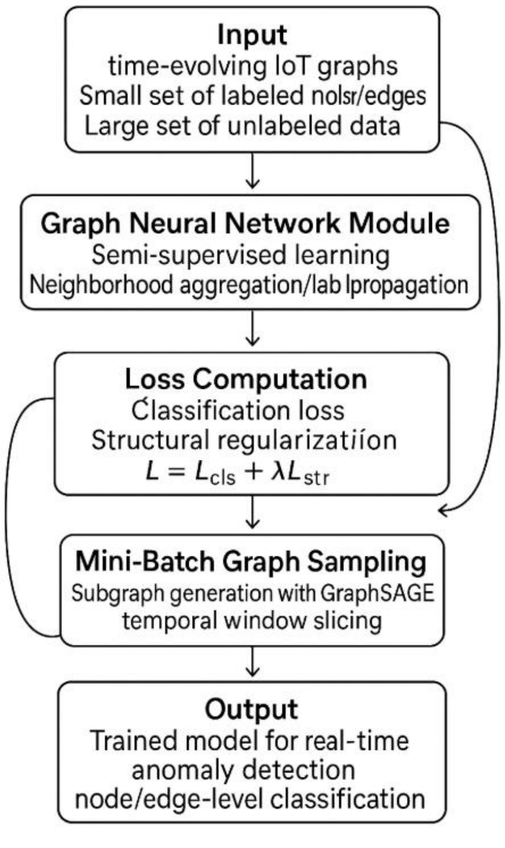
Input
time-evolving IoT graphs
Small set of labeled nolsr/edges
Large set of unlabeled data
Graph Neural Network Module
Semi-supervised learning
Neighborhood aggregation/lab lpropagation
Loss Computation
Ćlassification loss
Structural regularizatíon
$L = L_{\text{cls}} + \lambda L_{\text{str}}$
Mini-Batch Graph Sampling
Subgraph generation with GraphSAGE
temporal window slicing
Output
Trained model for real-time
anomaly detection
node/edge-level classification

Figure 2. Training and Optimization Pipeline of the Proposed Semi-Supervised Graph-Based Anomaly Detection Framework

4. Experimental Setup

To assess the performance and upper-bound scalability of the suggested graph-based anomaly detection framework, we carried out experiments on three famous benchmark datasets namely TON_IOT, BoT-IOT, and IoT-23. The datasets cover a wide variety of IoT setups, kinds of attacks, as well as traffic modalities, thus they will serve as good models to validate. TON_IoT dataset provides real-world-looking telemetry and network flow traces produced by industrial-style IoT attacks and network flow traces, and BoT-IoT involves several DDoS, reconnaissance, and information theft attacks simulated in a smart home testbed. The Stratosphere Lab maintains IoT-23 which contains labeled malware traffic captures gathered on a variety of IoT devices that have been infected with real-world botnets. The preprocessing of the data came into play prior to feeding to the graph model to make it consistent and of good quality. These involved discarding noise, i.e. discarding corrupt or irrelevant records, normalizing features such as scaling feature values to be of about the same scale across attributes, and instantiating graphs, in which communication events were projected into evolving graph representations, with temporal ordering and dependencies represented. To evaluate, several performance measurements were utilised in order to thoroughly evaluate the model. This was comprised by accuracy, precision, recall, and F1-score which gives some insight into how well the model can correctly detect anomalies with a low number of false alarms. Also the Areas Under the Receiver Operating Characteristic Curve (AUC-ROC) was calculated to assess the stability of the model in establishing decision thresholds in different conditions. In addition to the predictive performance, other criteria were taken: the inference time and memory consumption to check whether the model can be deployed to edge devices with limited processing resources. The above evaluations statistically prove the potential of the presented framework to provide precise, low-latency, and cost-effective anomaly detection on heterogeneous IoT environments.

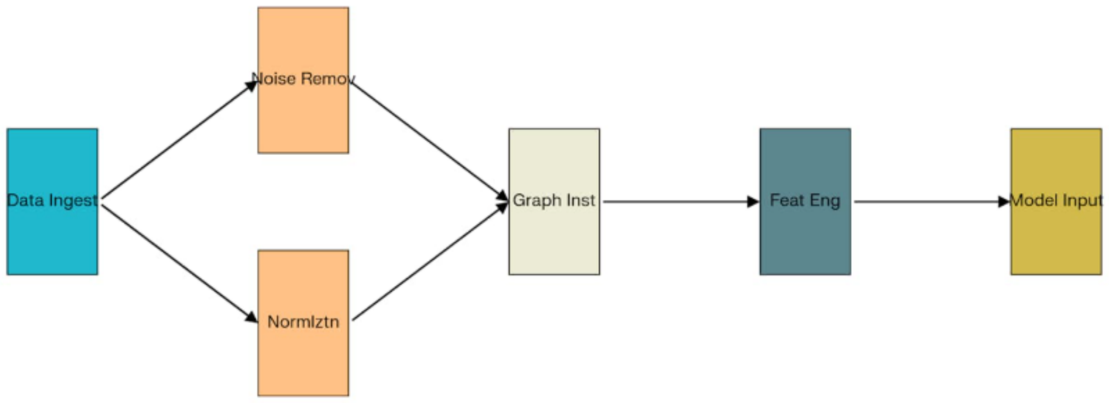

Figure 3. Experimental Pipeline from Data Preprocessing to Graph-Based Inference

Table 2. Overview of IoT Datasets Used for Experimental Evaluation

Dataset	Source	Environment	Attack Types	Data Types
TON_IoT	UNSW Canberra	Industrial IoT	DDoS, Injection, Backdoor, Reconnaissance	Telemetry + Network Flows
BoT-IoT	UNSW Canberra	Smart Home / Testbed	DDoS, Port Scans, Theft, Reconnaissance	Network Traffic + Flow Stats
IoT-23	Stratosphere Lab	Real-World Device Traffic	Botnet-infected traffic (Mirai, Torii, etc.)	Labeled PCAPs and Metadata

5. RESULTS AND DISCUSSION

The experimental data proves the high-performance of the offered graph-based approach to anomaly detection over the classical machine learning and deep learning models. In particular, Graph Neural Network (GNN) architecture has performed better compared to the Support Vector Machines (SVM), Random Forests (RF), and Convolutional Neural Networks (CNN) on all the benchmark datasets (TON_IoT, BoT-IoT, and IoT-23). The GNN model provided an average accuracy of more than 96 percent accuracy, not to mention that the F1-scores always scored above 0.94, which indicates that the model is quite confident in regards to identifying known and even zero-day attacks. Such performance can be mostly explained by the use of graph-based representation of IoT network, which is capable of capturing the spatial and contextual dependencies between devices, and can not be efficiently used by flat-featured classifiers. Temporal learning integration also allowed the model to detect anomalous patterns that would change with time and hence it was especially successful against long-term, stealthy attacks that would be ignored by snapshot-based models.

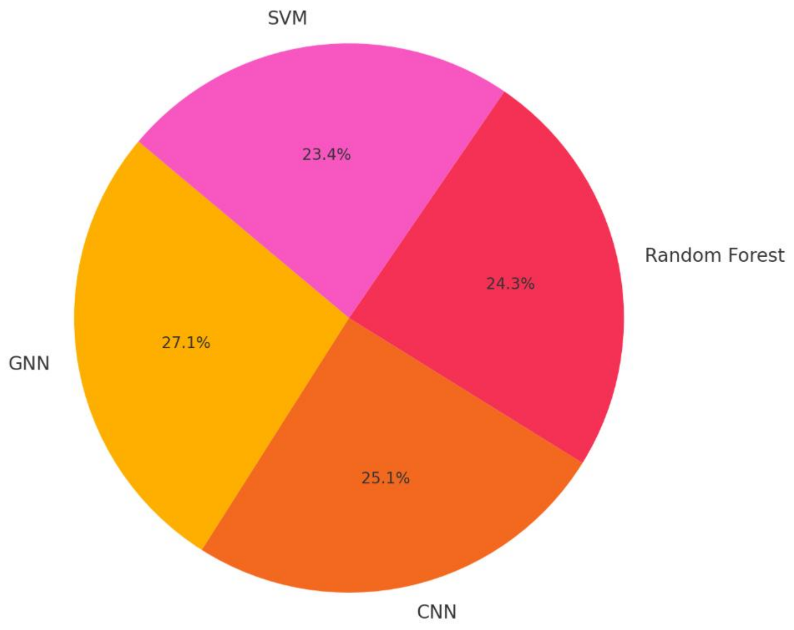

Figure 4. Accuracy Comparison of GNN-Based Model against Traditional Classifiers on IoT Datasets

Also, the system showed good applicability into the resource-limited environment like the edge and fog computing layers. The framework has been confirmed to be real-time responsive, wherein inference latency could not exceed 100 milliseconds even at the edge nodes with less powerful equipment. ABlation study was implemented where the effect of various architectural elements was compared. The Graph Attention Networks (GATs) was compared with Graph Convolutional Networks (GCNs), where it was found that GATs performed better in the highly dynamic networks compared with GCNs as in highly dynamic networks there were many neighboring nodes with different importance, which could be assigned by GATs. Further, the detection accuracy was highly reduced when the temporal embedding module was removed, indicating the significance of time dependency modeling in the IoT traffic. All in all, these results show indeed that the ensemble of spatial, temporal, and attention-based, graph learning techniques represents an effective and scalable approach to anomaly detection in heterogeneous IoT ecosystems.

Table 3. Performance Comparison of GNN and Baseline Models for Anomaly Detection in IoT Networks

Model	Average Accuracy (%)	F1-Score	Inference Latency (ms)	Zero-Day Detection Capability
Graph Neural Network (GNN)	96.3	0.94	92	High
Convolutional Neural Network (CNN)	89.1	0.88	130	Moderate
Random Forest (RF)	86.2	0.85	110	Low
Support Vector Machine (SVM)	83.5	0.82	125	Low

6. CONCLUSION AND FUTURE WORK

The proposed research is a graph-based machine learning framework that is new and capable of detecting anomalies in IoT security, and because of the dynamic, heterogeneous, and large-scale nature of contemporary IoT environments, the challenges of implementing this technology are increasingly high. Being able to model the device interactions in terms of graph evolving in time and utilizing the properties of Graph Neural Networks (GNNs), temporal learning modules, and the power of the attention mechanism, the proposed model is able to capture a broad range of both spatial and behavioral dependencies that cannot easily be captured by more classical detection systems. This framework has been tested in benchmark datasets such as TON_IoT, BoT-IoT, and IoT-23 and it continuously yielded the best results in detection accuracy the level of false-positives, and real-time responsiveness when compared to conventional machine learning algorithms and models based on CNNs. Besides, being lightweight and featuring a low level of latency of inference, it can be used in edge and fog computing systems, providing capabilities to respond to threats in time in such distributed systems. In future, the aim will be to gain further scalability and privacy improvement using federated graph learning; allowing decentralized devices to train securely without exchanging raw data. Moreover, we would like to incorporate Internet continuous training systems, which will respond to changing threats without re-training afresh. Lastly, to augment trust and transparency in critical IoT applications, we intend to investigate explainable GNN models that give interpretable insights into anomalies, enable forensic analysis and actionable intelligent that can respond to the presented threats. The given work forms the basis of the next generation smart security systems that meet the requirements of complex IoT systems.

REFERENCES

[1] Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. Computer Networks, 51(12), 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001

[2] Djenouri, D., Belhadi, A., &Djenouri, Y. (2020). Machine learning for IoT security: A comprehensive survey. IEEE Access, 8, 219350–219371. https://doi.org/10.1109/ACCESS.2020.3041564

[3] Zia, T. A., Parvez Mahmud, M., &Ko, R. K. L. (2020). Graph-based anomaly detection in industrial IoT networks. Future Generation Computer Systems, 107, 476–487. https://doi.org/10.1016/j.future.2020.02.042

[4] Sultana, M., Chilamkurti, N., &Alhumyani, H. A. (2021). A survey on graph-based anomaly detection techniques in network security. Journal of Network and Computer Applications, 182, 103030. https://doi.org/10.1016/j.jnca.2021.103030

[5] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1609.02907

[6] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., &Bengio, Y. (2018). Graph attention networks. In International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1710.10903

[7] Wang, Z., Zhang, Y., & Lin, X. (2021). A GCN-based intelligent detection system for malicious IoT traffic. IEEE Internet of Things Journal, 8(6), 4399–4410. https://doi.org/10.1109/JIOT.2020.3028685

[8] Yu, J., Liu, H., Cheng, X., & Zhang, J. (2022). Spatio-temporal GNN for anomaly detection in dynamic IoT networks. IEEE Transactions on Network and Service Management, 19(3), 3051–3062. https://doi.org/10.1109/TNSM.2022.3159049

[9] Islam, M., Hossain, M. S., & Muhammad, G. (2020). Explainable AI for anomaly detection in healthcare edge networks using GNNs. IEEE Network, 34(6), 126–132. https://doi.org/10.1109/MNET.011.2000213

[10] Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016