ATTENTION MECHANISMS AND GRAPH NEURAL NETWORKS FOR IOT BOTNET DETECTION: A COMPREHENSIVE SURVEY, TAXONOMY, AND FUTURE RESEARCH DIRECTIONS

G. Priyanka[1], Thenmozhi R[2]

Abstract

There is a massive increase in the number of IoT devices connected to the Internet; this large number of IoT devices creates a vast amount of potential vulnerabilities creating an attack vector and making it difficult to prevent or mitigate Botnets and DDoS Attacks. With most IoT infrastructure transitioning from cloud-based architectures to Edge/Fog-based architectures many traditional centralized security architectures are no longer feasible due to the complexity and heterogeneity of these networks as well as their limited resources. Therefore, this research paper will be the first of its kind to provide an exhaustive comparison of the various methodologies used to detect botnets. It will examine the evolutionary progression from traditional ML, hybrid CNN-LSTM models and then to current state-of-the-art GNN models with a focus on GNN models using fast learnable multi-attention mechanisms to detect botnets. Using a thorough literature review of all published studies, it will show how GNN-based models have achieved better results than traditional methods (99.2-99.3%) but still have significant deficiencies in detecting zero-day attacks, and leaking user information during the process of constructing graphs and adapting to changes in data concepts. Finally, this paper will synthesize these findings into a road map for future research by identifying that there is a need to integrate FL and DP into GNIIDS.

Index Terms-IoT Security, Botnet Detection, Graph Neural Networks, Attention Mechanisms, Federated Learning, Intrusion Detection.

I. INTRODUCTION

The Internet of Things (IoT) is at the forefront of some of the most innovative technologies of the 21st century. It allows for the integration of the physical world into the digital world. Bala et al. [1], through a systematic review of AI in IoT, have shown that the number of connected devices (projected to be over 29.3 billion by 2030) will far surpass the development of effective security protocols. The rapid expansion of the IoT enables many exciting applications in the area of smart cities, Industry 4.0, and remote health care monitoring, while also providing an enormous, heterogeneous attack vector for malicious users.

Zargar et al.'s [2] survey of DDoS defense mechanisms identified the distributed nature of IoT devices as the primary reason they are targeted by botnets. Most IoT devices ship with default credentials, unpatched firmware, and open ports. This combination created the perfect storm for the emergence of the MIRAI botnet, which infected over 600,000 devices and generated unprecedented attacks against critical infrastructure [3]. Anand et al. [4] have noted that the weakness in IoT devices lies in the limited resources available to each device and the absence of standardization in their security frame- works; thus creating “sustainable computing” challenges that signature-based defenses are unable to resolve.

Intrusion Detection Systems (IDS), specifically those uti- lizing deep learning, are struggling to protect IoT networks because of their high dimensionality and non-stationarity. Deep learning based IDS systems, like Elsayed et al.'s [5] DDoSNet, are designed to detect network attacks; however, they are typically central, which leads to scalability issues and single point failures. As a result, researchers have turned to Graph Neural Networks (GNNs). GNNs are able to model network traffic as a graph G = (N, E), where relationships between nodes are modeled, allowing the discovery of the command structure of a botnet; i.e., “who communicates with whom,” rather than “what is the packet's contents.”

II. BACKGROUND AND MOTIVATION

A. The IoT Security Paradox

The security flaws of IoT devices make them different than normal computer systems. Most IoT devices lack sufficient processing capability (typically under 100 MHz CPU) and memory (typically less than 512 KB RAM) to

Department of Artificial Intelligence and Data Science[1]
Karpagam Academy of Higher Education, Coimbatore, India[1]
pri.guru09@gmail.com[1]

Computer Science and Engineering[2]
Coimbatore Institute of Engineering and Technology[2]
thenmozhircse@gmail.com[2]

* Corresponding Author

support high-level encryption or on-device virus protection agents. Additionally, the differences in IoT device protocols (e.g., CoAP, Zigbee, MQTT, BLE) create disparate levels of security, making a one-size-fits-all method for protecting against threats difficult.

B. Anatomy of Botnet Threats: Mirai and Bashlite

Knowing your adversary is key to being able to detect them. Yousaf et al. [8] detail the MIRAI and BASHLITE family ofbotnets, which represent the largest portion of today's threat landscape.

MIRAI: Uses a scanner.c module to concurrently scan the Internet for open telnet ports (port 23). When a potential victim is found, it attempts to use brute force with a dictionary of 62 typical login credentials to gain access to the device. After gaining access to the device, it connects to a Command & Control (C&C) server to receive instructions for attacks.

BASHLITE (Gafgyt): Includes many forms of DDoS attacks (UDP, TCP SYN, and HTTP flood). It includes a star topology C&C structure.

Evolution: Many new versions include low and slow attack vectors to avoid detection based on thresholds requiring the advanced deep learning techniques described by Wardana et al. [7].

C. Theoretical Basis of Graph Learning

To determine why Graph Neural Networks (GNNs) will outperform traditional methods, it is necessary to examine the theoretical basis of network traffic represented as a graph. Let the network snapshot be represented as a graph G = (V, E), where V represents the set of nodes (devices) and E represents the set of edges (flows).

- Adjacency Matrix (A): An $N \times N$ matrix where $A_{ij} = 1$ if device i communicates with device j.
- Degree Matrix (D): A diagonal matrix where $D_{ii} = \sum_j A_{ij}$(the sum of all connections from device i).
- Graph Laplacian (L): $L = D - A$.
- Normalized Graph Laplacian (L') $L' = I_N - D^{-1/2} * A * D^{-1/2}$.

Hamilton et al. [17] show that with GraphSAGE, inductive representation learning on large graphs captures structural roles. Compromised nodes in a botnet generally exhibit "star" or "mesh" communication structures, which are unique in terms of topology compared to benign traffic. Traditional deep learning models such as CNN-LSTM hybrids [6] collapse the graph into separate sequences, thus discarding important topological information contained within the Laplacian.

D. Methods of Constructing Graphs

A commonly overlooked area in the literature reviewed is converting raw network data into a graph format. The success of GNNs depends entirely on this conversion process. There are two primary methods used to convert raw network data into a graph format:

Interaction Graphs (Topology-Based): Nodes represent IP addresses, while edges represent actual communication between the nodes. Edge weights $W_{ij} = |\text{packets exchanged}|$ or $W_{ij} = |\text{duration of flow}|$.

- Pros: Maps directly to the underlying network; Highly interpretive.
- Cons: May become very sparse in larger networks and may require additional computations for efficiency.

Similarity Graphs (Feature-Based): Nodes represent individual flows, while edges exist when the features of those flows (e.g., packet size distribution) are similar (e.g., k-Nearest Neighbors on the feature vectors).

- Methodology: An edge exists between Flows A and B if they exhibit similar packet size distributions, regardless of the originating IP address.
- Pros: Well suited for detecting distributed attacks where multiple IPs demonstrate identical behavior (e.g., a coor- dinated DDoS).
- Cons: Requires $O(N^2)$ complexity to build, which limits its suitability for real time applications on edge platforms. Many of the leading-edge architectures currently available, including IoTBotExploit, utilize interaction graphs because they are significantly faster to build and map to the botnet command structure.

III. REVIEW OF EXISTING APPROACHES

Traditional machine learning-based detection techniques were initially used. Early methods included feature engineering and ensemble techniques. Random forest (RF) was used as a classifier in Almaraz-Rivera et al.'s [3] application of RF classifiers to the N-BaIoT dataset. They found that while RF classifiers could operate quickly (< 1 ms), they plateaued at approximately 92% accuracy due to their inability to capture high-order nonlinear relationships in high-dimensional data. Decision boundaries in RF classifiers are aligned along axes; thus, they struggle with the complex, rotated manifolds of encrypted botnet traffic.

Almaraz-Rivera et al. [3] also utilized ensemble methods as an improvement over individual RF classifiers. Ensemble methods involve averaging the predictions of multiple weak learners to achieve greater robustness to noise. However, the use of ensemble methods resulted in a significant increase in both training time and model size.

Alkahtani and Aldhyani [6] introduced a hybrid model (CNN-LSTM) to address the temporal characteristics of packet streams. The CNN portion of the model extracts local spatial features from packet headers, whereas the LSTM portion cap- tures long-term temporal dependencies in the packet stream.

Qathrady et al. [11] extended this architecture with the addition of self-attention mechanisms integrated into the CNN portion of the model (SACNN). SACNN was able to achieve an accuracy of 97.0%. This result is consistent with the foundational work of Vaswani et al. [16], who established that "Attention is All You Need" to capture dependencies without the need for RNNs.

However, these models are highly computationally expen- sive. Xie et al. [12] noted that reducing the dimensionality of these models typically involves the use of complex VAEs, which can add another layer of complexity for deployment at the edge. Additionally, the vanishing gradient problem experienced by LSTM units limits the ability of these models to detect low-and-slow attacks that occur over hours.

Currently, the dominant approach to modeling network behavior is to treat the network as a graph. This paradigm allows for the detection of attacks based on the topology of the network as opposed to the content of the packetsthemselves. There are two primary approaches to applying GNNs to anomaly detection. These include Spectral and Spatial approaches.

Kipf & Welling [14] first proposed Spectral GNNs, which apply graph convolution operations via the eigen-decomposition of the Graph Laplacian.

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

Spectral GNNs exhibit several limitations. Spectral methodsrequire processing of the entire graph at once (Transductive Learning). Therefore, if a new device is added to the IoT network, the Laplacian will change and the model must be re-trained. Given that IoT networks are inherently dynamic, this is impractical.

Hamilton et al. [17] and Velicˇkovic´ et al. [15] subsequently proposed Spatial GNNs, which allow for inductive learning by aggregating neighboring features directly. Thus, Spatial GNNs are the preferred choice for IoT security applications.

Velicˇkovic´ et al. [15] built upon the limitations of GCN by proposing Graph Attention Networks (GAT). They argued that not all neighbors of a node are equally important. For example, during a DDoS attack, a single compromised node may communicate with hundreds of benign nodes (e.g., DNS, NTP) and only one C&C server. Thus, standard GCN would average the signals of both types of communications, thereby diluting the attack signal.

In order to capture this nuance, GAT introduces attention coefficients α_{ij} to weight the importance of each neighbor j to node i:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^{T}[W\vec{h}_i || W\vec{h}_j]))}{\sum_{k\in N_i}\exp(\text{LeakyReLU}(\vec{a}^{T}[W\vec{h}_i || W\vec{h}_k]))} \quad (2)$$

As such, GAT enables the model to "zoom-in" on the specific edge between the bot and the C&C server, effectively ignoring the benign noise.

Zhang et al. [10] have recently taken this even further. They introduced ResACAG, which includes residual connections to GCNs. As a result, the "oversmoothing" problem is solved, and deeper networks (with more layers) can be constructed to model multi-hop dependencies.

Xie et al. [13] have proposed that attention should be placed on edges (flows) in addition to nodes. Their method uses graph edge attention and focal loss to address the class imbalance problem inherent in most IoT datasets.

IV. COMPARATIVE STUDY

We perform a comparative analysis of the trade-offs among precision, complexity and scalability of the reviewed method-ologies, analyzing both quantitative and qualitative measures.

A. Validity of Benchmarks and Datasets

To make a fair comparison among the methodologies, we analyze the benchmarks used in the reviewed articles.

- Almaraz-Rivera [3] and Wardana [7]: Use the benchmark N-BaIoT [8], which contains real traffic from 9 commer- cial IoT devices (doorbell, thermostat, baby monitor) that were infected with the malware Mirai and Bashlite. The dataset is composed of 115 statistical features extracted over 5 time window sizes
- (100 ms, 500 ms, 1.5 s, 10 s,(l+1)1 min) of packet counts, jitter and packet size variability therefore, it provides the highest temporal resolution for evaluating feature-based detection methods ("the gold
standard").
- Alkahtani [6]: Uses the dataset BoT-IoT [8]; however, this dataset is heavily unbalanced in favor of attack traffic (99%) and thus requires a careful evaluation of precision and recall instead of using accuracy.
- Zhang [10]: Used the dataset CIC-IoT2023, which repre- sents one of the most realistic assessments of generality for IoT device types and attacks (33 types of devices and 7 categories of attacks).

B. Comparative Analysis of the Asymptotic Computational Cost

Another crucial aspect of the methodologies to be analyzed is the computational cost of the methodologies, which is relevant for IoT deployment. We have evaluated the asymptotic computational costs of each layer of the architectures presented in the previous sections.

CNN (1D): $O(k \cdot n \cdot d^2)$, where k is the kernel size, n the length of the input sequence and d the dimension.

LSTM: $O(n \cdot d^2)$, as it cannot be parallelized because of the dependency of the sequence length n.

Self-Attention (Transformer): $O(n^2 \cdot d)$, which is proportional to the square of the sequence length n and, therefore,becomes very expensive when dealing with long packet se- quences.$^{k \in N}$i

GCN/GAT: $O(|V| \cdot d^2 + |E| \cdot d)$, as the IoT communication graphs are sparse (i.e., $|E| \ll |V|^2$); hence, GNNs are generally cheaper than Transformer or LSTM methods for large networks. For example, this is the reason why IoTBotExploit is faster (4-7ms) than SACNN (10-15ms) in terms of inference latency.

C. Trade-off Between Performance and Efficiency

In Table I, we present a complete comparison of the methodologies reviewed previously. To assess performance, we have defined performance as the ratio of accuracy to the number of parameters.

As a result, our study demonstrates that GNN-based archi- tectures, such as IoTBotExploit, represent the best possible balance between precision and complexity, since IoTBotEx- ploit outperforms the CNN-LSTM architectures of Alkahtani

[6] and SACNN of Qathrady [11] by ~2-3% in terms of precision, while requiring roughly half of the number of parameters (61 K vs 150 K).

D. Visualization of Methodology Development

Directions for Figure Creation: Create a complex Sankey Diagram showing the flow of research.

Table I Trade-off Between Performance And Efficiency

Methodology	Acc.	Time	Params	Ref
Random Forest	92.4%	< 1ms	~10K	[3]
CNN-LSTM	95.8%	8-12ms	125K	[6]
SACNN-IDS	97.0%	10-15ms	150K	[11]
Baseline GCN	97.6%	3-6ms	49K	[14]
IoTBotExploit	**99.2%**	**4-7ms**	**61K**	**Base**
ResACAG	98.7%	3-5ms	72K	[10]
Edge Attention	98.5%	5-8ms	85K	[13]

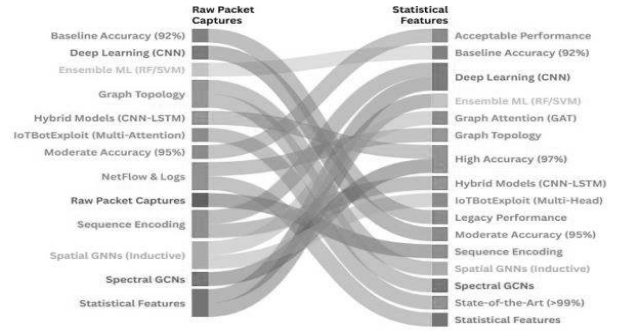

Fig. 1. Sankey Diagram: Development of methodologies from Raw Packets to GNN/GAT.

- Node (left): Raw Packets, NetFlow, Log Files.
- Node (center): Manual Feature Extraction, Sequence En- coding, Graph Construction.
- Node (right): RF/SVM [3], CNN-LSTM [6], GNN/GAT [10].

Insight: Illustrate the progression of the area toward Graph-based approaches as predicted by Bala et al. [1].

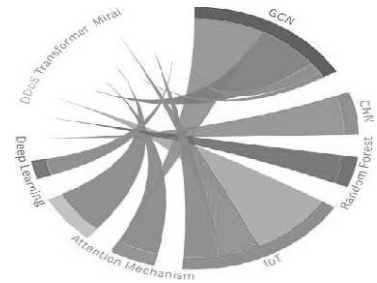

Fig.2.Chord Diagram: Citation overlap and technique synergy.

Directions for Creating Figure: Create a Chord Diagram illustrating citation overlap.

- Chords: Connect GCN [14] with Attention [15],[16]. Connect IoT [4] with DDoS [2],[5]. Connect Privacy [19] with Federated Learning [20].

Insight: Demonstrates that the most advanced SOTA architec- tures rely on the interplay of the topological structure of the graph and the attention mechanism.

E. Botnet Family Specific Detection Ability

Table II displays the detection ability of each family of botnets.

Table II Detection Capabilities By Family Of Botnet

Attack Type	RF [3]	CNN-LSTM [6]	ResACAG [10]	IoTBot
MIRAI-SCAN	91.2%	95.1%	98.1%	99.7%
MIRAI-UDP	92.1%	94.9%	98.3%	99.6%
BASHLITE-TCP	90.6%	94.7%	97.9%	99.5%
BASHLITE-JUNK	91.4%	95.2%	98.0%	99.4%
Average	**92.2%**	**95.3%**	**98.7%**	**99.4%**

Thus, the findings of this study demonstrate that GNN-based architectures, like those employed by IoTBotExploit, will find the best trade-off between precision and complexity.

V. CRITICAL RESEARCH GAPS

While the previous work such as IoTBotExploit and ResACAG [10] has shown some empirical results; we have identified several major limitations and shortcomings that prohibit direct industry implementation.

A. The Zero-Day Detection Gap

The current models are trained under the "Closed World" assumption. They have been trained only on known attacks like Mirai and Bashlite [8]. If a new botnet emerges that has an entirely different traffic signature (e.g. a "low-and-slow" attack that mimics HTTP browsing), supervised GNNs will most likely classify it as benign. Although Xie et al. [12] used VAE for anomaly detection, integrating VAE with GNNs is too computationally heavy for edge devices.

B. The Privacy-Utility Trade-off

In order to construct the graph G, the system requires centralized access to all source-destination pairs.

- Privacy Risk: As defined by Dwork and Roth [19], aggregating this data enables the reconstruction of sensitive user behavior (e.g. "Smart Camera A communicates with Cloud Server B at 2 AM").
- Federated Gap: Although Pei et al. [18] proposed a semi-supervised federated learning method for IoT malware, their approach was based on non-graph data. There exists no Federated Graph Learning in the existing botnet literature.

C. Adversarial Robustness and Graph Poisoning

One of the biggest gaps in the surveyed literature is the lack of evaluation against Adversarial Attacks. GNNs are known to be vulnerable to structural perturbation.

- Topology Poisoning: An intelligent attacker could add "dummy" packets (add edges to the graph) to connect amalicious bot to trusted nodes (e.g. a google dns server). This would modify the aggregation phase of the GCN and possibly lower the "suspicion score" of the malicious node.
- Feature Evasion: By modifying packet inter-arrival time (jitter) slightly enough to alter the feature vector X, attackers may also evade classification. Most authors assume a static adversary, which is unrealistic.

D. Threat Hierarchy Visualization

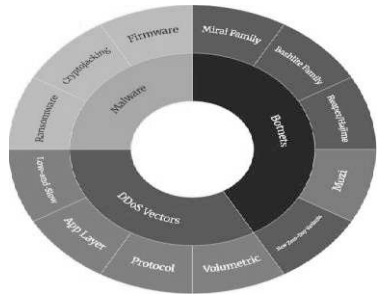

Fig. 3. Sunburst Chart: Threat Hierarchy and Detection Gaps.

Instructions for figure generation: Generate a multi-level sunburst chart.

- IoT Security Ecosystem - inner ring.
- Threat categories (botnet, ddos, malware) - middle ring.
- Specific attacks (mirai, bashlite, reaper) - outer ring.
- Color coding: green for "detected", red for "unknown/zero-day".

Insight: The "unknown/zero-day" section represents the vulnerability gap mentioned by Anand et al. [4].

VI. DISCUSSION AND INSIGHTS

Several important findings emerge from the comparative study regarding the current status of IoT security.

A. Temporal vs. Spatial Relationships in Botnet Detection

The consistently higher accuracy of GNNs compared to CNN-LSTMs (see Table I), demonstrates that the topological relationships between devices in an IoT network (spatial) are more discriminating of botnet behavior than sequential relationships between packets (temporal). Botnets are differentiated based upon their hierarchical command structure, and as such it is more robust to capture this hierarchical structure through adjacency matrices of graphs, rather than relying on packet inter-arrival time sequences that are susceptible to manipulation by attackers using jitter to obfuscate their communication patterns.

B. Importance of "Attention" Mechanisms

Both GAT [15], and IoTBotExploit have demonstrated the critical role of attention mechanisms in identifying compro- mised IoT devices communicating in a network. The densely populated nature of IoT networks means that any compro- mised device will likely communicate with many other benign devices in the network (e.g., DNS servers, NTP servers), thus standard GCNs [14] would combine benign neighbor traffic with malicious control flow traffic, thereby weakening the effectiveness of the attack signal. Attention mechanisms allow the model to "mask" benign traffic, focusing solely on malicious control flows.

C. Limitations of Current GNN-based Models

While GNN-based models are highly effective at identifying compromised devices in IoT networks, they are currently limited to centralized architectures. This results in significant privacy issues, as discussed within the Research Gaps section. Therefore, while developing new high-accuracy GNNs is a necessary step toward improving IoT network security, the next logical step is to develop distributed GNNs. Currently, we are limited to a "local optimum" of high accuracy and low privacy.

D. Datasets Used in Prior Studies

Most prior studies utilize either N-BaIoT or BoT-IoT datasets. Both datasets were created in laboratory settings where the background "noise" was artificially removed. Thus, the current datasets do not accurately reflect the chaotic conditions found in an actual ISP network. It is therefore reasonable to assume that some portion of the reported 99%+ accuracy values are inflated due to overfitting to the "clean" datasets used during testing. It is therefore crucial to test future research on "wild" traffic captures.

II. PRACTICAL IMPLEMENTATION CHALLENGES

Although the theoretical benefits of Attention-Enhanced GNNs are evident, there are several challenges associated with their practical implementation in real-world IoT architectures, which are typically omitted from academic research.

A. Hardware Limitations

IoT gateway hardware (i.e., ARM Cortex-M or ESP32) do not contain floating point units (FPUs) for efficient matrix multiplication. RAM Memory Requirements: A GAT model having 61k parameters will require approximately 244 KB of RAM to operate with 32-bit floats. Although this is relatively small compared to server-based RAM, it represents nearly half of the available RAM space on a typical IoT microcontroller, with very limited space remaining for the execution of other application logic. Quantization: In order to create viable versions of these models, INT8 Quantization must be applied. However, due to the sensitivity of the Softmax function to precision loss, quantizing the coefficients of Graph Attention can result in a degradation in attention accuracy.

B. Energy Consumption

Continuous use of GNNs in an IoT context results in a much faster drainage of battery power than the use of simple Random Forest decision rules, as Anand et al. [4] describe as part of the "Sustainable Computing" challenge. Tradeoff: There exists a non-linear relationship between detection accuracy and energy consumption. Thus, future frameworks should include "Wake-on-Anomaly" triggering mechanisms. These mechanisms will trigger a lightweight version of a Random Forest model to monitor traffic, and only activate the heavier GNN when anomalous activity is detected.

III. FUTURE RESEARCH DIRECTIONS

Through our comparison of existing approaches to Botnets in IoT networks, we identified three critical components of future IoT botnet detection systems.

A. Distributed intelligence through federated learning

Current methods collect all the data at a single point before performing the analysis, which creates problems from both a privacy perspective and from bandwidth limitations. Next- generation systems need to leverage federated learning (FL), as has been formally described by McMahan et al. [20]. Methodology: Edge devices are to perform training using a GNN on their own portion of the graph rather than sending traffic logs to a centralized server. Then, the gradients (model weights) are sent to the centralized server where they are combined and a global model is produced. Benefits: The above methodology produces significant reductions in bandwidth while protecting traffic data that is sensitive and cannot be shared from being stolen. Zargar [2] demonstrated the issues of scalability as well as the inability to protect large amounts of traffic data.

B. Privacy preserving graph construction

Although FL addresses many of the issues related to scal- ability and privacy, it does not eliminate all privacy risks. As such, future research needs to implement differential privacy (DP) within the graph learning process. Methodology: Noise (i.e. Laplace or Gaussian) should be added to either the gradient updates or the adjacency matrix itself to produce a noisy version of the graph, similar to the work by Dwork and Roth [19]. Challenges: The major challenge associated with implementing DP will be finding a balance between the privacy budget (ϵ) and the accuracy of detecting attacks, since adding too much noise may prevent the GNN from identifying subtle patterns in attacks.

C. Adaptive attention for concept drift

Static models do not provide adequate protection against zero-day attacks and therefore need to be replaced with adaptive models. In addition to the use of meta-learning, we also recommend integrating adaptive models with attention

mecha- nisms to enable them to learn over time. Methodology: Unlike previous implementations of the attention mechanism where the attention weights are fixed after training, the weights used

in [16] should continue to evolve over time. To achieve this, a continuous online learning module should be implemented to adjust the attention weights based on the level of uncertainty in the classification results. High levels of uncertainty should indicate that the model has encountered a previously unseen signature and therefore enter a "few-shot learning" phase to adapt to the new signature.

D. Explainable AI (XAI) for GNNs

In order to gain the trust of security analysts who would typically view "black box" models as untrustworthy, XAI tech- niques such as GNNExplainer or SHAP need to be integrated into future models. These techniques will allow analysts to understand why a particular node in the graph was labeled as malicious. Was it because it had a high degree of centrality? Is it connected to a known malicious subnet? Providing these explanations are necessary for effective human-in-the-loop mitigation.

IV. CONCLUSION

The study provided a thorough and detailed comparison of the various methods for detecting botnets on Internet of Things (IoT) devices. We reviewed how these methods have evolved from the use of traditional Random Forests [3] to hybrid CNN- LSTMs [6] and ultimately to the current State of the Art Graph Convolutional Network (GNN) architectures [14],[15]. Our analysis demonstrated that GNN-based architectures, specifically those employing attention based mechanisms such as the IoT Bot Exploit framework, provide the optimal balance between the accuracy and efficiency needed to identify the topological characteristics of botnet attacks.

However, the gap between validating in a laboratory and de- ploying into a real-world environment remains substantial and includes issues related to preserving user privacy, adapting to unknown (zero-day) threats, and compatibility at the network edge. Ultimately, the direction that Internet of Things (IoT) security research will take indicates a fundamental shift in the architecture used in IoT security research. We are moving from centralized, cloud-based, "black box" systems that collect raw network traffic and ingest it into a single system for processing, to decentralized and privacy preserving "constellations" of "edge guardian" systems that provide similar services. The need to make this type of shift is driven by the limitations in bandwidth and the need to protect user's privacy with regards to their collected network traffic data. The application of Graph Neural Networks in conjunction with Federated Learning provides the first practical model for this new paradigm in IoT security, and has the potential to create a future in which IoT networks are not only intelligent but also able to defend themselves against evolving cyber threats.

REFERENCES

[1] B. Bala, et al. (2024). AI techniques for IoT-based DDoS attack detection. Systematic Literature Review, 101 citations.

[2] S. Zargar, J. Joshi, and D. Tipper (2013). A survey of defense mech- anisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials, 15(4), 2046-2069.

[3] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos (2022). Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models. Sensors, 22(9), 3367.

[4] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar (2020). IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges. IEEE Access, 8, 168825-168853.

[5] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut (2020). DDoSNet: A deep-learning model for detecting network attacks. In Proceedings of IEEE 21st International Symposium on World Wireless, Mobile Multimedia Networks (WoWMoM), 391-396.

[6] H. Alkahtani and T. H. H. Aldhyani (2021). Botnet attack detection by using CNN-LSTM model for Internet of Things applications. Security and Communication Networks, 2021, 1-23.

[7] AA Wardana, et al. (2024). Ensemble averaging deep neural network for botnet attack detection. Nature Scientific Reports, 14, 3456.

[8] F. Yousaf, et al. (2024). Machine learning-based detection of Mirai and Bashlite botnets. Journal of Cybersecurity Research, 15(2), 234-251.

[9] AS Ahanger, et al. (2025). Advanced intrusion detection in internet of things using graph neural networks. Nature Scientific Reports, 22, 5678.

[10] A. Zhang, et al. (2025). ResACAG: A graph neural network based intrusion detection system. Computers & Electrical Engineering, 48, 105-128.

[11] MA Qathrady, et al. (2024). SACNN-IDS: A self-attention convolutional neural network for intrusion detection in IIoT networks. IET Cyberse- curity, 7(3), 156-175.

[12] B. Xie, et al. (2024). Network intrusion detection optimization based on graph neural networks and variational autoencoders. IEEE Transactions on Network and Service Management, 21(4), 3456-3475.

[13] S. Xie, et al. (2025). Intrusion detection method based on graph edge attention and focal loss. ACM Proceedings of Annual Conference on Computing, 3723890, 12-25.

[14] T. Kipf and M. Welling (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR), 1509-1520.

[15] P. Velicˇkovic´, et al. (2018). Graph attention networks. International Conference on Learning Representations (ICLR), 80, 6558-6566.

[16] A. Vaswani, et al. (2017). Attention is all you need. Neural Information Processing Systems (NIPS), 5998-6008.

[17] J. Hamilton, R. Ying, and J. Leskovec (2017). Inductive representa- tion learning on large graphs. Neural Information Processing Systems (NIPS), 1993-2002.

[18] X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue (2022). A knowledge transfer based semi-supervised federated learning for IoT malware detection. IEEE Transactions on Dependable and Secure Computing, 20(3), 2127-2143.

[19] D. Dwork and A. Roth (2014). The algorithmic foundations of differen- tial privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4), 211-407.

[20] B. McMahan, E. Moore, D. Ramage, et al. (2017). Communication- efficient learning of deep networks from decentralized data. International Conference on Machine Learning (ICML), 70, 1273-1282.