IoT-MGSec: Mitigating Man-in-the-Middle Attacks in IoT Networks using Graph-based Learning

Ebelechukwu Nwafor, Carter Schmidt
Department of Computing Sciences
Villanova University
Villanova, PA, USA
enwafor@villanova.edu

Habeeb Olufowobi
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, USA
habeeb.olufowobi@uta.edu

***Abstract*—The Internet of Things presents a transformative era of device connectivity while creating a new paradigm shift in the process. This, however, has been met with some major pitfalls, such as an increase in device insecurity, characterized by Main-in-The Middle (MiTM) attacks. In this paper, we propose IoT-MGSec, a novel solution to mitigating MiTM attacks using graph-based learning. Our approach employs graph modeling and embedding techniques to learn node and edge features such that we can generate a robust classifier to detect MiTM attacks with high accuracy. We validate the effectiveness of our approach by comparing its performance to baseline models, and the results indicate that our approach outperforms the baseline models. The findings suggest that this approach offers a more robust solution to detecting and mitigating Man-in-the-middle attacks, and it holds potential for integration into real-time intrusion detection systems, further enhancing its capacity to secure devices within the IoT landscape.**

***Index Terms*—Internet of Things, Intrusion detection, Graph representation learning, Man-in-the-Middle attack.**

I. Introduction

The Internet of Things (IoT) has revolutionized how devices interact, enhancing user experiences across various domains, from smart homes to industrial automation. IoT has greatly improved convenience and usability by enabling real-time communication among heterogeneous devices. Its implementation has yielded substantial benefits across multiple domains, enhancing efficiency and overall quality of life [1], [2]. However, despite its benefits, security has remained a critical concern to its widespread adoption. Many existing security measures have been implemented as an afterthought, which has led to potentially undesirable consequences. In addition, the heterogeneous connectivity results in new attack vectors, which necessitates proactive planning and innovative intrusion detection and prevention techniques. Among the myriad threats faced by IoT ecosystems, Man-in-The-Middle (MiTM) [3] attacks have emerged as particularly insidious. MiTM attacks represent one of the most covert threats in an IoT environment, as such attacks can discreetly intercept, alter, inject malicious payloads, and eavesdrop on communications between parties without their knowledge. This potentially can result in severe long-term consequences if not adequately addressed.

There has been some research in the area of mitigating MiTM attacks. Alicherry et al. [4] proposed DoubleCheck, a low-overhead approach to mitigate MiTM attacks that verifies the network connection path. Benton et al. [5] proposed an approach that verifies the executing device's certificate thumbprint against a stored certificate for detecting MiTM attacks in the Secure Socket Layer (SSL). Vallivaara et al. [6] introduced an approach for detecting MiTM attacks using timestamps in the TCP header of network packets. While some of these existing approaches offer promising solutions to mitigate MiTM attacks, their effectiveness against sophisticated attacks and adaptability to IoT device diversity remain uncertain. To address these challenges, we propose IoT-MGSec, a novel approach for mitigating MiTM attacks in IoT environments using graph-based learning.

Graphs offer a promising approach to modeling network data, as most real-world data can be naturally represented as graphs that capture complex relationships between components across networks. Graph-based learning facilitates the extraction of meaningful representations of network components and in understanding intricate graph features. The extracted representation can then be effectively used in downstream application tasks such as predicting interactions within the graph components, such as classifying nodes contained in the graph or predicting edge relationships (also known as link prediction). This approach has been shown to provide effective results in various domains, including health informatics, recommender systems, among many others [7], [8]. Graph-based learning can be leveraged to provide a comprehensive map of complex network interactions that exist between IoT devices. This allows for enhanced detection of network intrusions and anomalous network instances, significantly improving the detection accuracy of detecting MiTM attacks in an ever-evolving landscape of IoT networks.

In this paper, we leverage semantic relationships from edges and nodes contained in a graph to develop a robust classifier that can detect MiTM attacks with high precision and accuracy. In this approach, interactions across the network are represented as a homogeneous graph where connections are depicted as edge relationships and nodes represent network components. This network graph is then converted into an embedding space representation using a homogeneous embedding approach such as node2vec. This embedding space preserves the graph semantics such that nodes that appear close to each

other are also represented similarly to the embedding space. The node embedding is used as an input feature to develop a classifier that distinguishes edge connections across the network as malicious or benign. We validate the feasibility of our approach using a dataset that consists of known IoT-based MiTM attacks. Additionally, we compare IoT-MGSec against a baseline model, demonstrating its superiority in accurately classifying malicious instances of MiTM attacks. By doing so, we aim to provide a real-time MiTM detection and mitigation system that enhances the security posture of IoT networks. Our approach addresses the challenges posed by the heterogeneity of IoT devices and their connections, offering an innovative solution for detecting and thwarting evolving MiTM attacks in the dynamic landscape of IoT environments.

The rest of the paper is organized as follows: Section II provides the background and preliminaries of graph embeddings, Section III outlines the threat model and system architecture, Section IV provides the methodology of IoT-MGSec, Section V presents the experimental results and analysis, Section VI presents a comprehensive review of related work, and Section VII concludes the paper with future research directions.

II. Graph Embeddings

Graphs are a fundamental data structure for representing complex relationships and interactions. Graph embeddings have proven to be a versatile tool in various domains, facilitating the transformation of graph-structured data into low-dimensional vector representations. The process involves encoding graph properties, where nodes represent entities (vertices), and edges signify relationships between them, into dense and continuous numerical formats [9]. This format preserves the graph's inherent structure and semantic information, enabling efficient computation of graph algorithms and downstream tasks such as visualization, node classification, link prediction, and graph-level comparison [10], [11]. In the IoT environment, graph embeddings offer a promising approach to address some security challenges, including MiTM, by representing the IoT network as a property graph, where IoT devices are nodes, and communication channels between devices are edges. The embeddings capture the network's structural relationships and interactions, allowing for valuable insights to be gleaned for detecting and preventing anomalous events that may indicate attack attempts. One significant advantage of graph embeddings is their ability to capture the graph's topology and vertex-to-vertex relationships. Devices exhibiting similar communication patterns or common neighbors in the IoT network are likely to have identical representations in the embedding space. This property facilitates the detection of suspicious patterns that deviate from the learned normal behavior, signaling potential security threats.

Various graphs can model different aspects of the IoT network, including communication graphs representing the communication patterns between IoT devices and topology graphs capturing devices' physical connections and arrangements. Also, behavior graphs model the similarities in behavior between devices, while temporal graphs incorporate time-series data to analyze changes over different time steps. Inputs to these graphs differ depending on the scenarios and may include device metadata, communication logs, sensor readings, and network traffic data. The potential output of graph embedding is a low-dimensional vector representing a part of the graph (or a whole graph) that can inform various security-related tasks [12]. Graph embeddings leverage multiple techniques to learn meaningful representations, including Node2Vec, graph convolutional network (GCN), and matrix factorization-based methods.

A. Preliminaries

Given a graph with nodes representing entities and edges denoting relationships between them, the goal is to map each node to a continuous vector in a lower-dimensional space. Mathematically, a graph can be defined as $G = (V, E)$, where V is the set of nodes and E is the set of edges. The objective is to learn a function $\phi : V \rightarrow \mathbb{R}^d$, where $\phi(v)$ represents the embedding of node v, and d is the desired dimensionality of the embeddings. The function phi captures the graph's topology and preserves node similarities and relationships. Each edge e_{ij} in the edge set E describes the connection between two different nodes v_i and v_j, represented as $e_{ij} = (v_i, v_j)$, where $v_i, v_j \in V$, and nodes v_i and v_j are adjacent nodes.

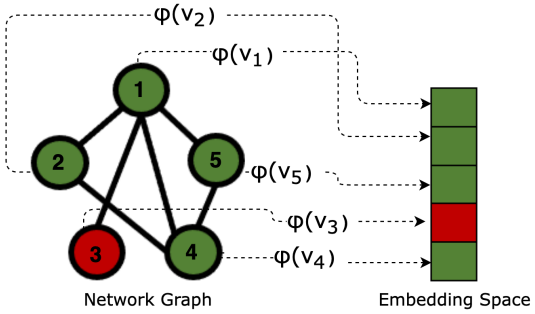

Fig. 1. An Example of a Graph Embedding Approach.

Graphs can be classified into various categories based on edge properties, including directed or undirected graphs, homogeneous or heterogeneous graphs, and weighted or binary graphs. Graph embedding techniques leverage different strategies to generate embeddings that effectively capture the graph's diverse structural patterns. The choice of the embedding method depends on the characteristics of the graph and the specific objectives of the analysis. For our graph embedding approach, we utilize a homogeneous graph embedding technique, such as Node2Vec.

B. Node2vec

Node2Vec is a graph embedding technique that applies the principles of word2vec to the graph domain to generate low-dimensional vector representations (embeddings) for nodes in large-scale graphs. The approach captures local and global

structural patterns, making it valuable for various graph representation learning tasks. Node2Vec generates multiple random walks on a graph that are sequences of nodes obtained by traversing the graph from a starting node in a stochastic manner [13]. Node2Vec utilizes biased random walks that balance breadth-first (BFS) and depth-first (DFS) graph searches [9]. This balance is achieved through the return parameter (p) and the in-out parameter (q), which control the likelihood of revisiting nodes during a random walk and the probability of moving towards nodes that are further away or within the local neighborhood, respectively. By appropriately tuning p and q parameters, Node2Vec captures higher-order proximity between nodes, preserving community structure and structural equivalence. Higher-order proximity is the ability to gather structural information beyond nearby nodes, while community structure refers to densely connected nodes within the same community. The embeddings generated by Node2Vec accurately encode meaningful similarities between nodes, making them well-suited for tasks like node classification and link prediction. The flexibility, scalability, and representation learning capabilities of Node2Vec make it a valuable tool for analyzing IoT networks and addressing security challenges.

III. IoT-MGSec Framework

In this section, we provide a high-level overview of the IoT-MGSec framework. We describe its threat landscape and its conceptual system architecture for detecting attacks.

A. Threat Model

Our network topology consists of multiple devices that are connected through a few or more critical network gateways. These gateways are essential to routing data traffic across the network and ensuring data delivery to appropriate devices. We define MiTM threats as attacks involving data injection or interception of data packets over a shared network. These attacks involve network packets that have been intercepted or rerouted from their destination to a malicious network. We assume that attacks performed on devices are conducted to a remote or local network while using attack measures such as address resolution protocol (ARP) spoofing, SSL stripping, and eavesdropping. This process implies that the attacker can steal, alter, or simply monitor data, breaching the confidentiality and integrity of a device or the entire network.

B. System Architecture

IoT-MGSec consists of four main components:

- The **Data processing** phase involves curating and processing network datasets that contain malicious and normal network traffic information. Processing involves removing null values, deleting redundant data, and converting data to an appropriate format.
- In the **Graph generation** phase, the processed dataset is converted into a homogeneous graph of device interactions where nodes represent the device types and edges represent the device connections. For a graph $G = (V, E)$, an edge is generated for each node V such that if devices v_1 and v_2 are connected, an edge $e_{12} = (v_1, v_2)$ is formed.
- **Graph embeddings** phase involves converting homogeneous network graphs into an appropriate embedding space using node2vec graph embedding techniques.
- The **Classification** phase involves using the generated graph embeddings and machine learning classifiers to develop models that classify nodes or edges contained in the graph as either malicious or benign. This might be done at the node level (node classification) or at the network level (edge classification).

Figure 2 shows a visual representation of the system approach. Data is generated from device interactions across the network. This data is then processed to ensure it is in an appropriate format. Once the data has been processed, it is represented as a graph that encapsulates all network interactions. The graph is used to generate embeddings that capture the semantic relationship between nodes and edges. These embeddings are fed into a classifier, resulting in a model that distinguishes between malicious and benign data. Next, we provide details of the implementation of our approach.

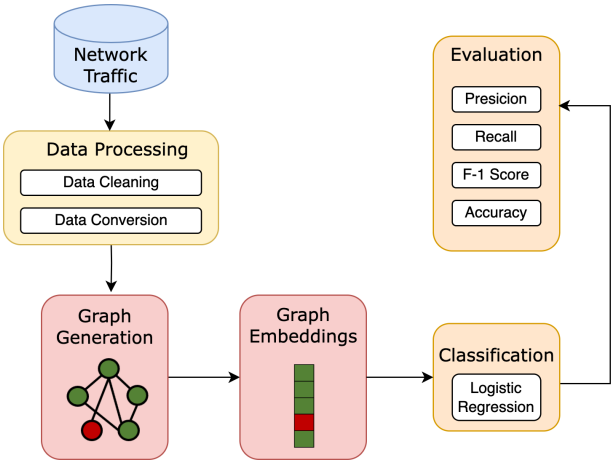

Fig. 2. IoT-MGSec System Architecture.

IV. Implementation

In this section, we provide details of the IoT-MGSec framework implementation. Subsequent subsections will discuss the components in detail.

A. Model Implementation

1) Data Acquisition: Choosing a robust dataset is crucial to validating our approach's effectiveness. For our implementation, we use the IoT Network Intrusion Dataset [14], which consists of network attacks on IoT devices. This dataset consists of 42 packet capture (pcap) network files, with each file containing thousands of normal and malicious network interactions. Each pcap file contains a specific attack type, such as man in the middle, denial of service, and Mirai botnets. Out of these attack types, we focused on the MiTM attacks, which consist of 6 of the 42 files in the dataset. After converting

the pcap files into csv as well as identifying malicious and normal packets, the dataset was imported into the pandas dataframe for more seamless data manipulation. For the data processing step, we began by labeling malicious and normal packets. Next, we encoded all of the categorical string values, such as protocol information, MAC, source, and destination IP address, into a numerical representation, as this is required by the classifiers when developing our classification models. After data processing, the feature set consists of Packet Number (when data was sent), Protocol Number, Frame Length (packet size), Malicious flag, Encoded source, and destination.

2) Graph Construction: We use NetworkX [15], an open-source graph library for studying and manipulating complex network graphs, to convert our processed data into a homogeneous graph. Each node consists of source and destination information, and each edge is labeled as either malicious or benign. Additionally, We divide our network dataset into six distinct subgraphs, allowing for more granular analysis of the network's dynamics and interactions across entities contained in the network. By segmenting the dataset into smaller subgraphs, we can isolate and pinpoint specific patterns that may be lost in a larger graph. Furthermore, working with multiple subgraphs enhances the computational efficiency of our approach, making data processing faster. This process not only facilitates a fine-grained analysis of the network interactions but also promotes a robust and efficient system.

3) Graph Embedding: To generate more meaningful representations of the network graph structure, we use node2vec to generate embeddings that capture the semantics of the graph. This simple and effective approach has been shown to work well for homogeneous graphs [16], [17]. We implement node2vec using stellargraph [18], a library for developing graph machine-learning algorithms. Table IV-A3 provides details of the parameters used in our node2vec implementation.

Parameter	Value
p, q	1, 1
Random walk length	80
Window size	10
Number of walks	10
Vector size	128

TABLE I
PARAMETERS USED FOR THE NODE2VEC IMPLEMENTATION.

B. Predicting MiTM attacks using node connections

The main goal of generating embeddings is to develop a robust classifier for detecting MiTM attacks. Algorithm 1 provides details of the prediction approach. The node2vec approach produces embeddings that offer representations for individual nodes across the graph. To obtain edge embeddings, we explored several techniques, using different operators to merge pairs of node embeddings into a single embedding. For a graph $G = (V, E)$, suppose $e_{ij} = (v_i, v_j)$ represents an edge. We define a function $\phi : V \rightarrow \mathbb{R}^d$, which represents the mapping of nodes to an embedding space. The edge embedding, M, can be defined using the source node embedding $\phi(v_i)$ and the destination node embedding $\phi(v_j)$. For example, one method can involve computing the product of v_i and v_j, i.e., $(M_{ij} = (\phi(v_i) \times \phi(v_j))$. We evaluated four distance operators:

- $M_{ij} = (\phi(v_i) \times \phi(v_j))$
- $M_{ij} = |\phi(v_i) - \phi(v_j)|$
- $M_{ij} = (\phi(v_i) - \phi(v_j))^2$
- $M_{ij} = (\frac{(\phi(v_i)+\phi(v_j))}{2})$

We evaluated each operator on every graph, selecting the best-performing result for each graph for its corresponding classifier. Overall, the best performing operators were $|\phi(v_i) - \phi(v_j)|$ and $(\phi(v_i) - \phi(v_j))^2$. Figure 3 provides a visual description of the edge embeddings for each graph using principal component analysis (PCA) dimensionality reduction.

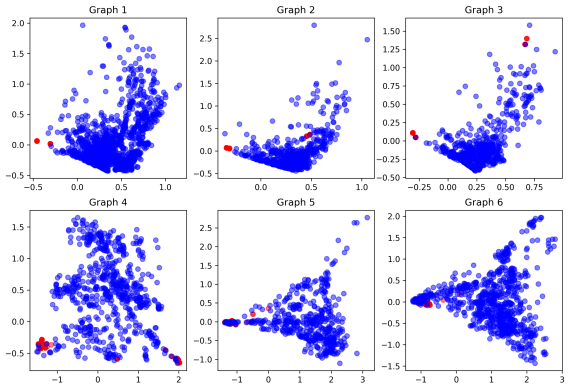

Fig. 3. A visualization of edge embeddings for each graph using Principal Component Analysis (PCA). The red and blue dots represent malicious and normal edge embedding, respectively.

Algorithm 1 MiTM Attack Prediction Using Edge Embeddings

Require: Graph $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$ such that $e_{ij} = (v_i, v_j)$.

1: Initialize node embeddings using the node2vec algorithm: $N \leftarrow$ Node2Vec(G)
2: Initialize an empty list for edge embeddings: $M \leftarrow []$
3: **for** each edge $e_{ij} = (v_i, v_j) \in E$ **do**
4: Compute edge embedding $m_{ij} = |N_i - N_j|$
5: Append m_{ij} to M
6: **end for**
7: Partition X and Y into training and testing sets
8: Train a classifier using Logistic Regression on the training set
9: Evaluate classifier performance on the test set

Algorithm 1 utilizes node2vec to obtain node embeddings, capturing the structural patterns and relationships of individual nodes. The algorithm captures semantic relationships between edge connections by computing edge embeddings using the absolute difference between connected node embeddings. Given a graph $G = (V, E)$ as input, the algorithm outputs a classifier capable of predicting potential MiTM attacks based on edge

Graph Number	Baseline				IoT-MGSec			
	F1-score	Precision	Recall	Accuracy	F1-score	Precision	Recall	Accuracy
1	**0.85**	**0.89**	**0.84**	**0.85**	0.95	0.96	0.95	0.95
2	0.62	0.63	0.62	0.65	0.94	0.95	0.94	0.94
3	0.76	0.77	0.76	0.77	**0.99**	**0.99**	**0.99**	**0.99**
4	0.51	0.85	0.56	0.71	0.79	0.86	0.80	0.80
5	0.60	0.83	0.62	0.62	0.90	0.90	0.90	0.90
6	0.57	0.83	0.57	0.78	0.55	0.67	0.60	0.60

TABLE II
PRECISION, RECALL, F1-SCORES AND ACCURACY FOR EACH GRAPH

embeddings. In the algorithm, node2vec initializes the node embeddings in line 1, and an empty list for edge embeddings is initialized in line 2. The edge embeddings are computed as the absolute difference between the embeddings of the two nodes connected by the edge and are appended to the list of edge embeddings in lines 4 and 5. Line 7 involves splitting the edge embeddings into training and test sets using the EdgeSplitter class in StellarGraph. In addition, the data is split into a feature set (X) and output labels (y) for both the test and training sets. The output labels are derived from the labels of the edges as defined in the graph. The processed edge embeddings serve as input features to the binary classifier in line 8. For this classifier, we choose to use Logistic Regression due to its simplicity and efficiency in binary classification tasks.

V. EVALUATION AND ANALYSIS

A. Evaluation metrics

We evaluate the feasibility of IoT-MGSec using standard machine learning evaluation metrics such as precision, recall, and F-1 score. Precision defines the number of true malicious traffic out of all the malicious traffic detected by the classifiers. This is defined as: $\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False Positive}}$. Recall is defined as the number of malicious instances detected out of the entire malicious traffic present, which is characterized using: $\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}}$. F-1 score is defined as the harmonic mean of the precision and recall characterized by: $\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$.

B. Result

In this section, we provide an evaluation of the performance of IoT-MGSec. We compare the performance of our approach to a baseline model. Table II provides a description of the results for the baseline and IoT-MGSec.

1) Baseline: In our baseline evaluation, we employed a logistic regression model with feature engineering without using edge embeddings as specified in IoT-MGSec. This approach was applied using all six graphs. The results were promising, with Graph 1 having the highest precision, recall, and F-1 scores of 0.85, 0.89, and 0.84, respectively. Graph 4 recorded the lowest F-1 score at 0.51.

2) IoT-MGSec Approach: We evaluate our approach using the evaluation metrics. The results reveal notable improvements compared to our baseline model. Some graphs achieved a precision, recall, and F-1 score exceeding 0.90. Graph 3 outperformed other graphs with precision, recall, and an F-1 score of 0.99. The results are promising and underscore the notion that incorporating graph-based learning into intrusion detection systems provides enhanced detection of MiTM attacks with higher accuracy.

VI. RELATED WORK

This section discusses related work in graph-based intrusion detection and their applications in IoT security. We categorize the approaches based on their primary focus.

A. Graph-based IoT Botnet Detection

Several methods have been proposed in the literature that address the use of graph-based learning techniques for intrusion detection. Nguyen et al. [19] proposed a lightweight approach for detecting IoT botnets by extracting high-level features from functional-call graphs known as PSI-Graph. The results show that the approach can detect these attacks with high accuracy. Pahl et al. [20] proposed a graph-based access control micro-service approach for IoT security. This approach runs as a microservice on each IoT device by intercepting and firewalling inter-service communication represented as a graph. This graph is used to classify inter-service communication traffic as normal or anomalous based on a defined communication model for each microservice.

B. Graph-based Anomaly Detection

Alasmary et al. [21] proposed an approach to detect malware in an IoT network using control flow graphs. This approach utilizes various characterizing features of the control flow graph to build a deep learning-based classifier that detects malware intrusions with high accuracy. Sanz et al. [22] proposed GRAFFITO-IDS, a graph-based approach to detecting network intrusions. This approach infers information from a graph using a time-windowed snapshot of the input network traffic. They evaluated their approach using three classification models, and the results indicate that their approach improves threat detection accuracy from the baseline. Parveen et al. [23] proposed a graph-based anomaly detection approach for detecting malicious insider threat intrusions using ensemble-based stream mining. The authors compared the approach for supervised and unsupervised learning and discovered that the supervised learning approach performs best.

C. Graph Embedding for IoT Security

While various graph-based intrusion detection approaches have been proposed, limited methods exist to address the challenge of learning meaningful embeddings for nodes in

large-scale graphs in the context of IoT. Paudel et al. [24] proposed a graph-based outlier detection in the IoT (GODIT), an approach that detects denial of service attacks using real-time graph network traffic. Their approach utilizes a shingling-based graph sketching model for graph embedding generation. Nwafor et al. [25] introduced a provenance graph-based approach for detecting anomalies in network sensor data utilizing cosine similarity to compare graph structures. Abusnaina et al. [26] proposed an approach for adversarial learning attacks on graph-based IoT malware detection systems using a combination of control flow graphs and deep learning. Their approach employs two different methods: off-the-shelf adversarial learning algorithms and graph embedding and augmentation. Manzoor et al. [27] proposed Streamspot, an efficient approach for detecting anomalous instances in streaming heterogeneous graphs using clustering. Graphs are represented as sketch vectors and are compared based on the relative frequency of their local substructures using a similarity function.

VII. Conclusion and Future Work

In this paper, we propose IoT-MGSec, a graph-based learning approach for detecting MiTM attacks. Our IoT-MGSec transforms network data into graphs, encapsulating intricate interactions within the network. These network graphs are subsequently translated into graph embeddings that capture the semantics of graph interactions across the network. We evaluate the feasibility of our approach by comparing the detection accuracy of our approach to a baseline model. The results are promising, with IoT-MGSec detecting MiTM attacks with high accuracy. In future work, we plan to employ the use of advanced graph learning models such as Graph Convolution Networks and GraphSAGE. These models eliminate the need for feature engineering, streamlining the representation learning process.

References

[1] P. Brous, M. Janssen, and P. Herder, "The dual effects of the internet of things (iot): A systematic review of the benefits and risks of iot adoption by organizations," *International Journal of Information Management*, vol. 51, p. 101952, 2020.

[2] M. Kassab, J. DeFranco, and P. Laplante, "A systematic literature review on internet of things in education: Benefits and challenges," *Journal of computer Assisted learning*, vol. 36, no. 2, pp. 115–127, 2020.

[3] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the https protocol," *IEEE Security & Privacy*, vol. 7, no. 1, pp. 78–81, 2009.

[4] M. Alicherry and A. D. Keromytis, "Doublecheck: Multi-path verification against man-in-the-middle attacks," in *2009 IEEE Symposium on Computers and Communications*, 2009, pp. 557–563.

[5] K. Benton, J. Jo, and Y. Kim, "Signaturecheck: A protocol to detect man-in-the-middle attack in ssl," in *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research*, ser. CSIIRW '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi-org.ezp1.villanova.edu/10.1145/2179298.2179365

[6] V. A. Vallivaara, M. Sailio, and K. Halunen, "Detecting man-in-the-middle attacks on non-mobile systems," in *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 131–134. [Online]. Available: https://doi-org.ezp1.villanova.edu/10.1145/2557547.2557579

[7] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[8] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: Graph-based attention model for healthcare representation learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 787–795. [Online]. Available: https://doi.org/10.1145/3097983.3098126

[9] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

[10] M. Xu, "Understanding graph embedding methods and their applications," *SIAM Review*, vol. 63, no. 4, pp. 825–853, 2021.

[11] E. Nwafor and H. Olufowobi, "Towards an interactive visualization framework for iot device data flow," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 4175–4178.

[12] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE transactions on knowledge and data engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[13] A. Epasto and B. Perozzi, "Is a single embedding enough? learning node representations that capture multiple social contexts," in *The world wide web conference*, 2019, pp. 394–404.

[14] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, "Iot network intrusion dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/q70p-q449

[15] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[16] F. Hu, J. Liu, L. Li, and J. Liang, "Community detection in complex networks using node2vec with spectral clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 545, p. 123633, 2020.

[17] H. Zhou, G. Sun, S. Fu, L. Wang, J. Hu, and Y. Gao, "Internet financial fraud detection based on a distributed big data approach with node2vec," *IEEE Access*, vol. 9, pp. 43 378–43 386, 2021.

[18] C. Data61, "Stellargraph machine learning library," https://github.com/stellargraph/stellargraph, 2018.

[19] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "A novel graph-based approach for iot botnet detection," *International Journal of Information Security*, vol. 19, no. 5, pp. 567–577, 2020.

[20] M.-O. Pahl, F.-X. Aubet, and S. Liebald, "Graph-based iot microservice security," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–3.

[21] H. Alasmary, A. Khormali, A. Anwar, J. Park, J. Choi, A. Abusnaina, A. Awad, D. Nyang, and A. Mohaisen, "Analyzing and detecting emerging internet of things malware: A graph-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8977–8988, 2019.

[22] I. J. Sanz, G. A. Fontes Rebello, and O. C. Muniz Bandeira Duarte, "Graffito-ids: A graph-based algorithm for feature enrichment on online intrusion detection systems," in *2022 6th Cyber Security in Networking Conference (CSNet)*, 2022, pp. 1–7.

[23] P. Parveen, N. McDaniel, Z. R. Weger, J. Evans, B. M. Thuraisingham, K. W. Hamlen, and L. Khan, "Evolving insider threat detection stream mining perspective," *Int. J. Artif. Intell. Tools*, vol. 22, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:15499906

[24] R. Paudel, T. Muncy, and W. Eberle, "Detecting dos attack in smart home iot devices using a graph-based approach," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 5249–5258.

[25] E. Nwafor, A. Campbell, and G. Bloom, "Anomaly-based intrusion detection of iot device sensor data using provenance graphs," in *1st International Workshop on Security and Privacy for the Internet-of-Things*, vol. 59, 2018.

[26] A. Abusnaina, A. Khormali, H. Alasmary, J. Park, A. Anwar, and A. Mohaisen, "Adversarial learning attacks on graph-based iot malware detection systems," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 1296–1305.

[27] E. A. Manzoor, S. Momeni, V. N. Venkatakrishnan, and L. Akoglu, "Fast memory-efficient anomaly detection in streaming heterogeneous graphs," 2016.