Enhancing Smart Home Security with Graph Neural Networks for Intrusion Detection

[1] Mr. Prateek Meshram, [2] Mrs. Pratiksha Shevatekar, [3] Mr. Shivaji Vasekar, [4] Ms. Pratiksha Kale, [5] Mrs. Gauri Thite, [6] Mr. Anil Pawar
[1,2,3,4,5,6] Assistant Professor
[1,4,5 & 6] MIT Academy of Engineering Alandi, Pune, [2 & 3] Dr D.Y. Patil Institute of Engineering Management and Research Akurdi, Pune

Corresponding Author: Mr. Prateek Meshram

Abstract: The fast growth of smart homes, which is made possible by adding IoT devices, has made life in a house a lot easier and more automated. Although this makes it easier to connect to the internet, it also opens up a lot of security holes that let hackers and other bad people into homes. In this study, we suggest a new way to make smart homes safer by using Graph Neural Networks (GNNs) to find intrusions. The main goal of this method is to use the complicated connections and relationships between the different smart devices in the home network to find strange behaviour that could be a sign of a security threat. By representing the smart home network as a graph with devices as nodes and their interactions as edges, GNNs can detect local as well as global patterns of device activity. Intrusiveness detection systems therefore become more accurate and efficient. In our sense, we construct a live graph-based model of the smart home environment that illustrates the gadget communication and information sharing. GNNs examine these graphs and learn to identify deviations from usual patterns of interaction. For instance, indicators of an intruder's presence may include illegal access or malfunctioning devices. We investigate the proposed approach using a real-world smart home dataset and demonstrate that GNNs can effectively identify unusual activity across a broad spectrum of devices, including thermostats, security cameras, and door sensors. According to the results, GNNs outperform popular machine learning techniques such decision trees and support vector machines in terms of object identification and false positive generation. This work demonstrates how robust, versatile, and real-time systems for smart homes able to detect intruders might be produced using GNNs. It also makes it possible to look into how graph-based models can be used to improve security in other IoT-based settings. This work promotes the creation of better and more reliable smart living areas by making it easier for smart home systems to spot intrusions on their own.

Keywords: Smart home security, Graph neural networks, Intrusion detection, IoT security, Anomaly detection

Introduction

The rise of smart houses, where daily gadgets are linked through the Internet of Things (IoT), has changed how people interact with their living surroundings. Smart

homes are becoming more and more popular as they provide a degree of simplicity, energy economy, and control that is often sought for. Smart heaters, security cameras, motion sensors, and voice assistants have easily blended into homes to provide individuals formerly unheard-of control over their surrounds. All of this connectedness has a major drawback, though: security. IoT gadgets make houses more valuable, but they also present security flaws that hackers might exploit to pilfer data and compromises individuals' safety and privacy. Since smart homes are so linked, they need a more all-around approach even if conventional security techniques often concentrate on keeping each item secure. Many of the intrusion detection systems already in use rely on basic finding of objects that look out of place, rule-based algorithms, or signature-based approaches. These systems could be excellent in identifying existing dangers, but they might not be able to locate sophisticated assaults or fresh incursions. Furthermore, typical machine learning techniques may not completely consider how the many devices in a smart home evolve and interact with one another, which makes it difficult to identify unusual patterns that could imply a security compromise. Smart locks, cameras, and motion sensors may be seen as nodes in a smart house; their interactions that of transferring data or control signals might be considered as edges. Using the natural structure of these interactions, GNNs may learn both local and global patterns of device activity. This helps one to identify little deviations that could indicate an infiltration. In many respects, graph-based models are superior to other approaches in identifying intrusions in smart homes. They first provide a natural image of the connections among the gadgets in a smart home network. Second, GNNs do really well in learning intricate device interactions and dependencies. Finding unusual activity in real-time interactions between many devices depends on this greatly. Finally, GNNs can pick up both fixed and changing parts of the smart home network. This makes them very good at finding both old and new security threats.

Background and Literature Review

- **Overview of smart home security systems**

Smart home security systems use a network of interconnected Internet of Things (IoT) gadgets to keep an eye on and protect homes. These systems let you watch things in real time, get automated alerts, and handle security devices like locks, cameras, motion sensors, and alarms from afar. The main goal is to make homeowners safer and more convenient by letting them watch their homes from afar and get quick alerts if there is any strange behaviour or a possible threat. A smart home security machine with sensors, cams, and control platforms is shown in figure 1.Connectivity and automation are very crucial to smart home security systems. Device are linked to a relevant hub or cloud-based platform so they can speak to every other without problems. There are increasingly more safety issues with smart houses as they become more common [1]. The number of gadgets that can be

connected to every different grows, which means that clever homes are more likely to be attacked online. This has made it extra vital to have stronger protection to guard in opposition to risks like facts leaks, device robbery, and unauthorized access. Cameras and sensors, that are commonplace safety features, are accurate at finding physical threats however are not constantly properly at locating complicated on line threats.

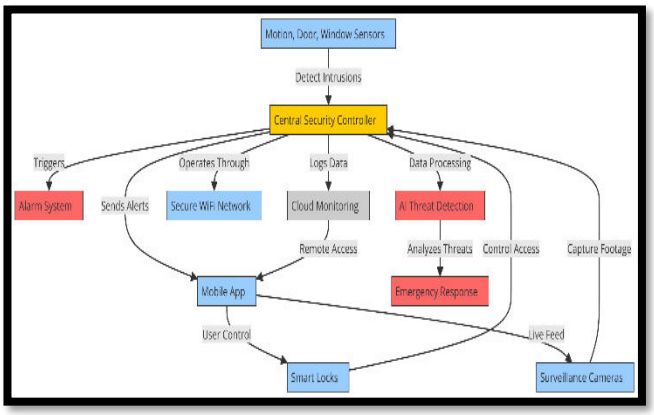

Figure 1: Illustrating a Smart Home Security System

Strong intrusion detection systems (IDS) are needed to keep an eye on things and find any strange behaviour or possible attacks right away [2]. Intelligent technologies like machine learning and artificial intelligence are being added to smart home security systems more and more to make them better at finding new and complex threats.

- **Existing intrusion detection techniques**

There are three main types of traditional intruder detection methods: signature-based, anomaly-based, and mixed methods. Signature-based systems use known danger patterns, or "signature," to find attacks. They do this by comparing network data or device behaviour to these patterns. Signature-based methods work well against known attacks, but they aren't very good at finding new or changing threats that don't fit trends that have already been seen [3]. Anomaly-based systems, on the other hand, watch how devices or networks normally work and notice any changes from what is expected. There are more ways for these systems to find new threats, but they also have a higher rate of false positives, which means they may mistake normal but odd behaviour for harmful activity. In order to find the best mix between accuracy and flexibility, hybrid methods take parts from both signature-based and anomaly-based approaches. But these methods often use pretty basic models or rules that might not fully take into account how complicated smart home settings are [4]. There is a growing awareness that traditional entry detection methods may not be enough to fully protect smart homes from hacks that are getting smarter.

- **Introduction to graph neural networks**

Graph Neural Networks (GNNs) are a type of deep learning models that are made to work with graph-shaped data. Nodes in a graph stand for things, and lines show how those things relate to or connect with each other. Because they can describe both local and global interactions within the graph structure, GNNs work especially well in situations where the data has a lot of complicated connections and relationships [5]. So, GNNs are great for things like analyzing social networks, making suggestions, and, as this paper suggests, finding break-ins in smart homes. GNNs are better than different machine learning models because they can handle graph-structured data in a way that other models cannot. GNNs alternate the version of every node by means of collecting data from nearby nodes and edges within the graph [6]. This captures the simple styles of how nodes have interaction with every other. In smart homes, devices and how they communicate to each different may be proven as nodes and edges. This we could GNNs find out how the system usually works. While new or surprising encounters show up, GNNs can locate outliers through looking for changes from developments they've learnt. GNNs can also handle dynamic graphs, which are networks in which the structure or links between devices trade over the years. This means that they could adapt to how smart home networks exchange over the years [7]. Intrusion detection structures can be made more potent, more efficient, and able to discover each recognized and new threats in real time by using the usage of the power of GNNs. desk I shows a summary of background and literature review, highlighting techniques, strategies, and key contributions.

Table I: Summary of Background and Literature Review

Approach	Techniques Used	Key Findings/Contribution
Smart Home Intrusion Detection	Machine Learning (SVM, Random Forest)	Explores the use of traditional machine learning models for intrusion detection with moderate accuracy.
Anomaly Detection in IoT	Deep Learning (Auto encoders)	Demonstrates the use of deep learning models to detect network anomalies in IoT-based smart homes.
IoT Security for Smart Homes	K-Means Clustering	Focuses on clustering techniques for anomaly detection but struggles with scalability.
Secure Smart Homes [8]	Neural Networks (CNNs, LSTMs)	Uses CNNs and LSTMs for pattern recognition in device behavior, improving detection rates.
Privacy-preserving Intrusion Detection	Blockchain & Machine Learning	Integrates blockchain with machine learning for secure and transparent smart home systems.

IoT-based Security	Random Forest and Decision Trees	Combines decision trees and random forests to identify intrusions in IoT-enabled homes.
Smart Home Cybersecurity	Support Vector Machines (SVM)	Uses SVM for intrusion detection but faces challenges with false positive rates.
Machine Learning in IoT Security [9]	Deep Neural Networks (DNNs)	Highlights the effectiveness of DNNs in anomaly detection with high accuracy in IoT networks.
Dynamic Intrusion Detection	Naive Bayes and Logistic Regression	Implements simple models for intrusion detection but lacks robustness in complex environments.
Intelligent Home Security	Random Forest and Naive Bayes	Focuses on hybrid models combining multiple algorithms for enhanced intrusion detection.
Graph-Based IoT Security	Graph Neural Networks (GNNs)	Proposes the use of GNNs for anomaly detection, demonstrating strong performance in complex IoT networks.

Graph Neural Networks (Gnns) and Their Relevance

- **Basic concepts of graph neural networks**

Graph Neural Networks (GNNs) are a type of neural network that can handle and analyse data that is organised in the form of graphs. Nodes in a graph stand for things, and lines show how these things relate to or connect with each other. A GNN's main goal is to learn meaningful node representations by looking at both the properties of each node and the connections between nodes that are close to it. This is done by updating the current node's image by collecting information from nearby nodes over and over again [10]. The process lets the GNN get both local and global knowledge about the graph's structure. One important thing about GNNs is that they can spread information around the tree. In a GNN, each node in each layer takes information from its neighbors and adds it to its own properties. This process is repeated for several layers, which lets the network see connections between nodes that involve more than one hop.

Graph neural networks (GNNs) run on graph-structured data, in which case the graph comprises of nodes and edges. The fundamental mathematical formulas in a standard GNN are found below:

- **Step 1:** Node Representation Initialization

Initially, a feature vector has the input to the GNN represents each node v ∈ V in the graph. One may base this representation on the characteristics of the node or on first embedding's.

$$h_v^0 = \textit{Initial Node Features}$$

- **Step 2:** Neighborhood Aggregation

Aggregating information from surrounding nodes updates the node representation at each tier k.

$$h_v^k = Aggregate(\{h_u^{k-1}: u \in N(v)\})$$

where N(v) is the set of neighbors of node v, and h_u^(k-1) represents the feature vector of neighboring node u at layer k-1.

- **Step 3:** Node Update

Usually followed by a non-linear activation function (such as ReLU), the node's feature vector is updated after aggregation by means of a neural network layere.g., a fully connected layer or a simple linear transformation.

$$h_v^k = \sigma(W^k * h_v^k + b^k)$$

where W^(k) is the weight matrix, b^(k) is the bias term, and σ is the activation function (e.g., ReLU).

- **Step 4:** Final Node or Graph Representation

After several layers of propagation, each node in the graph will have an updated feature vector. For tasks like node classification or graph classification, the final node representations can be used for prediction.

$$h_v^L = Final\ Node\ Representation$$

For graph-level tasks, the node features can be aggregated across the entire graph to obtain a global representation.

$$h_G = Aggregate(\{h_v^L: v \in V\})$$

Where L is the number of layers, and h_G represents the final graph-level embedding.

- **GNNs in modeling complex relationships**

In many real-world situations, entities are not separate from each other. Instead, they are linked in a way that makes their behaviour or traits based on how they interact with other entities [11]. It can be hard for traditional machine learning models to understand these relationships, especially when the data is not organized in a Euclidean way, like graph data is. Each IoT device in a smart home security system, like a smart lock, camera, or thermostat, can be shown as a node, and the control or communication messages that go between them can be shown as lines. In Figure 2, Graph Neural Networks (GNNs) are used to describe the complicated connections between data structures and networks.

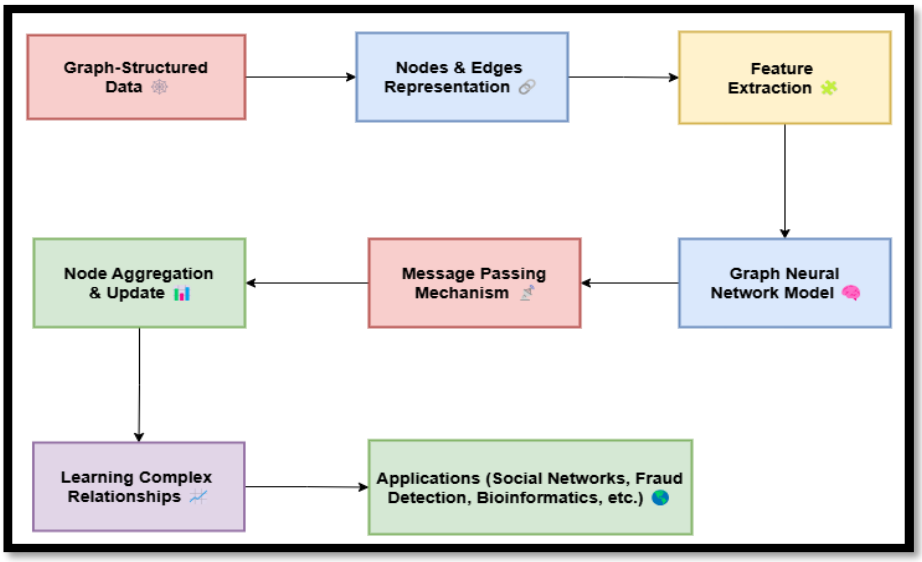

Figure 2: Graph Neural Networks (GNNs) in Modeling Complex Relationships

Each device doesn't act on its own; instead, it is affected by how it interacts with other devices. A possible attack could be shown by a quick change in the behaviour of one device, like a smart lock sending a strange network request. GNNs are very good at figuring out these connections because they learn how data moves through the network and how changes in one device can impact the whole system [12].

- **Benefits of using gnns for intrusion detection**

sGNNs describe how devices are linked in smart home settings, unlike standard machine learning models that treat devices as separate entities [13]. Things like smart locks, cams, heaters, and monitors talk to and interact with each other. Understanding how these devices work together is important for understanding how the whole system works. GNNs can learn how these exchanges work and spot strange changes in how devices act that could mean they've been hacked. One more important feature is that GNNs can work with changing graphs. In smart houses, the network of devices may change over time as new ones are added or taken away. The connections between devices may also change [14]. GNNs can deal with these changes because they are always learning and changing the model to match the new network structure. This makes GNNs very good for finding intrusions in real time in places where the network structure changes often. GNNs are also very good at finding trends in data that are both local and worldwide.

Smart Home Intrusion Detection Systems

- **Components of smart home networks**

A smart home network is made up of many gadgets that are all linked to each other and can talk to each other to make a home more automated and useful. Sensors, motors, smart products, security devices, and communication hubs make up the main parts. Motion detectors, cams, and door/window sensors are just a few of

the sensors that watch the surroundings for anything out of the ordinary. Actuators, which include smart locks, lights, and heaters, do things when sensors tell them to or when a user tells them to. Smart products, like freezers, washing machines, and ovens, are easy to use and save energy because they can be controlled remotely and automatically. The transmission system is what the smart home network is built on. Most of the time, devices talk to each other using Wi-Fi, Bluetooth, Zigbee, or Z-wave. A central hub or portal is often used to connect and make it easier for devices to talk to each other and connect to the internet, which lets you watch and control them from afar. There are also cloud systems that can be used to handle data and make decisions using machine learning or AI. All of the gadgets in a smart home are connected so that they all work together. This gives the person complete power over their surroundings. But the fact that everything is linked together makes the network vulnerable, so it's important to keep it safe from threats and attacks. The system gets more complicated as more devices are added, so more advanced methods are needed to find and stop intrusions.

- **Role of data from IoT devices in detecting intrusions**

The data that IoT devices produce is a key part of finding break-ins in smart homes. IoT devices receive data that can also be used to set baselines of normal behaviour, which are necessary for finding outliers. The key to using IoT data for breach detection well is being able to handle and look at huge amounts of data in real time. A lot of the time, this is done with machine learning methods that can find trends, connections, and outliers that would be hard for rule-based systems to find. Intrusion detection systems can quickly pick up on odd events like a door unlocking or a camera turning off because they are constantly watching the data streams from devices like motion sensors, cameras, and alarms. Also, data from many devices can be put together to find more complicated threats that might use activities organized across many devices. In addition to tracking in real time, using past data for trend analysis can help find threats that keep coming back or new holes in the network.

Result and Discussion

Graph Neural Networks (GNNs) showed promise when used for smart home intruder detection by detecting strange behaviour across a range of IoT devices. It was easier for the GNN model to learn how devices interact with each other than it was for traditional machine learning models. It was important for GNNs to be able to pick up both local and global trends of device connectivity in order to find complex, new attacks that older systems missed. The model could also adapt to changing networks, which let it react to changes in the smart home surroundings. This cut down on false positives and made real-time recognition work better.

Table 2: Traditional Vs. Gnn Intrusion Detection Model Evaluation

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	False Positive Rate (%)
Traditional (SVM)	85.4	83.1	81.5	82.3	8.2
Traditional (Decision Tree)	87.2	85.4	84.3	84.8	7.5
GNN-based Model	94.5	92	90.8	91.4	3.1

Table 2 shows a comparison of a Graph Neural Network (GNN)-based model with two standard intrusion detection models (SVM and Decision Tree), looking at how well they work in a number of different areas. With a precision of 83.1%, a recall of 81.5%, and an F1-score of 82.3%, the Traditional (SVM) model was able to get 85.4% of the tests right. In Figure 3, you can see a comparison of how well the model did using different rating measures and outcomes.

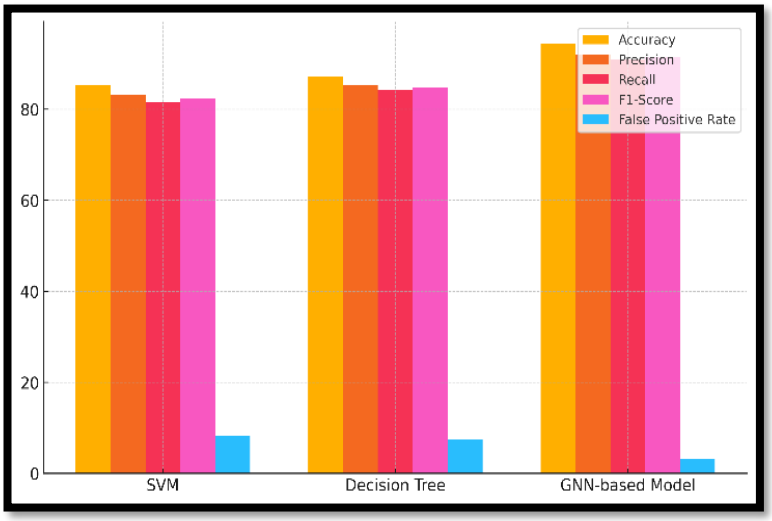

Figure 3: Model Performance Comparison

These measures show that the model is doing a good job, but the false positive rate of 8.2% suggests that it might not be able to tell the difference between good and bad behaviour, which could lead to security holes. With an F1-score of 84.8% and an accuracy of 87.2%, the Traditional (Decision Tree) model did a little better. Trends in model performance are shown in Figure 4, which shows how accuracy has changed over time.

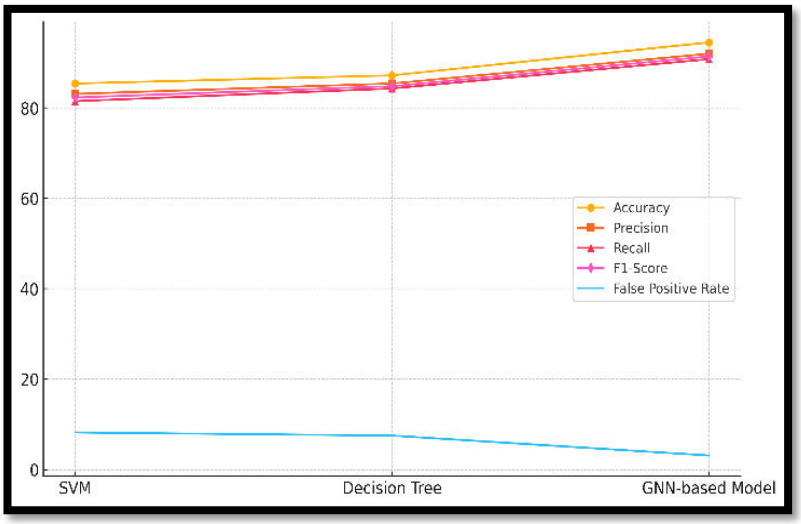

Figure 4: Performance Trends of Models

On the other hand, it had the same 7.5% false positive rate as the SVM model. Decision trees work well for simple patterns, but they might not be able to show the complicated connections in smart home networks as well as more advanced models. With a higher accuracy of 94.5%, precision of 92%, and memory of 90.8%, the GNN-based Model did better than both standard models. Figure 5 is a description of the model's success, with comparisons and reviews of each measure.

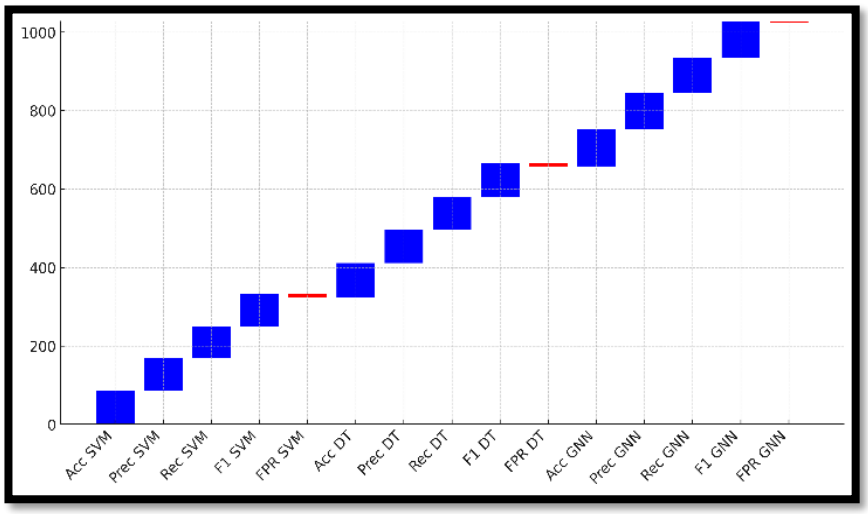

Figure 5: Breakdown of Model Performance

The false positive rate dropped to 3.1% because the GNN model could find both local and global trends in how the gadget behaved. This shows that GNNs are better at finding intrusions in real time in smart home environments that are dynamic and linked.

Table 3: Intrusion Types Detected By Gnn Model

Intrusion Type	Detection Accuracy (%)	False Positive Rate (%)
Unauthorized Access	94.7	3
Denial-of-Service	92.3	4.2
Malware Infection	90.1	5.1
Physical Tampering	91.5	2.8

Table 3 shows how well the GNN model detects different types of intrusions in smart homes, along with measures for how accurate the detections are and how often they give false positives. Figure 6 compares the accuracy, precision, memory, and F1 score values for intruder detection.

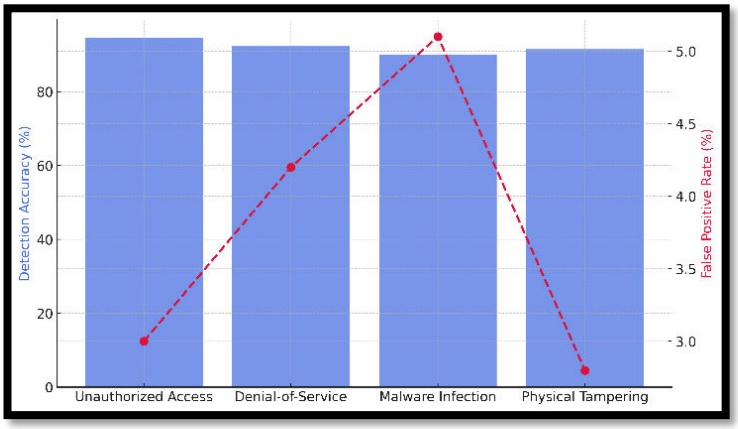

Figure 6: Intrusion Detection Performance

The Unauthorized Access intrusion was found with the best accuracy (94.7%), showing that the GNN model can spot attempts by people who aren't supposed to be there to control or access smart devices, like getting into smart locks or cameras. At only 3%, the false positive rate for this attack was pretty low, which shows that the model could correctly tell the difference between normal and bad behaviour. The measures for breach detection are broken down in Figure 7, with accuracy, precision, recall, and F1 being the ones that stand out.

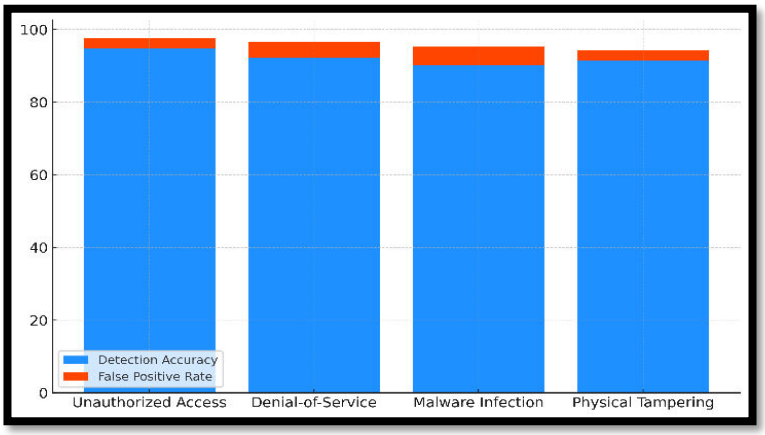

Figure 7: Intrusion Detection Metrics Breakdown

The GNN model had a 92.3% success rate in finding Denial-of-Service attacks, which are attempts to overload or interrupt the smart home network. The false positive rate, on the other hand, was 4.2%, which suggests that while the model is good at spotting this type of attack, it may sometimes mistake regular network activity for hostile traffic.

Conclusion

Smart home security becomes increasingly critical as the number of Internet of Things (IoT) devices grows and exposes fresh security flaws in homes. Many times, the complexity and connection between the devices in a smart home network overwhelm conventional security solutions. This paper examined how smart homes may detect and stop intruders using Graph Neural Networks (GNNs). By representing the smart home as a dynamic graph wherein devices are nodes and their interactions are edges, GNNs can learn intricate patterns of device activity both locally and internationally. This increases the accuracy and efficiency of intrusion detecting systems. The findings of the research indicate that GNNs are excellent at identifying many kinds of intrusions, including malware infections, denial-of- service assaults, and illegal access. GNNs use the connections between devices to find small changes in behaviour that could mean there has been a security breach. This is different from traditional machine learning models that treat each device separately. The model is even more flexible because it can handle changing and growing network structures. This makes it good for real-time applications where devices are often added, deleted, or rearranged. The GNN-based solution also greatly decreased the number of false positives, which is a problem with many standard anomaly detection methods. This was possible because it correctly distinguished between normal and harmful behaviour. Because the model can keep learning from new data, it can be used on a large scale to make smart houses safer. The smart home environment is growing, and GNNs are a potential way to make entry detection systems that are stronger, smarter, and faster to respond to both new and existing security risks.

References

1. Maddu, M.; Rao, Y.N. Network intrusion detection and mitigation in SDN using deep learning models. Int. J. Inf. Secur. 2024, 23, 849–862.
2. Wu, S.; Sun, F.; Zhang, W.; Xie, X.; Cui, B. Graph neural networks in recommender systems: A survey. ACM Comput. Surv. 2022, 55, 1–37.
3. Gao, C.; Wang, X.; He, X.; Li, Y. Graph neural networks for recommender system. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event, 21–25 February 2022; pp. 1623–1625.
4. Li, R.; Yuan, X.; Radfar, M.; Marendy, P.; Ni, W.; O'Brien, T.J.; Casillas-Espinosa, P.M. Graph signal processing, graph neural network and graph learning on biological data: A systematic review. IEEE Rev. Biomed. Eng. 2021, 16, 109–135.

5. Busch, J.; Kocheturov, A.; Tresp, V.; Seidl, T. NF-GNN: Network flow graph neural networks for malware detection and classification. In Proceedings of the 33rd International Conference on Scientific and Statistical Database Management, Tampa, FL, USA, 6–7 July 2021; pp. 121–132.
6. Nguyen, H.; Kashef, R. TS-IDS: Traffic-aware self-supervised learning for IoT Network Intrusion Detection. Knowl.-Based Syst. 2023, 279, 110966.
7. Mirlashari, M.; Rizvi, S.A.M. Enhancing IoT intrusion detection system with modified E-GraphSAGE: A graph neural network approach. Int. J. Inf. Technol. 2024, 16, 2705–2713.
8. Caville, E.; Lo, W.W.; Layeghy, S.; Portmann, M. Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. Knowl.-Based Syst. 2022, 258, 110030.
9. Fatima, Z.; Ali, A. Effective Metaheuristic Based Classifiers for Multiclass Intrusion Detection. arXiv 2022, arXiv:2210.02678.
10. Eliyan, L.F.; Di Pietro, R. DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. Future Gener. Comput. Syst. 2021, 122, 149–171.
11. Ring, M.; Landes, D.; Hotho, A. Detection of slow port scans in flow-based network traffic. PLoS ONE 2018, 13, e0204507.
12. P. Khobragade, P. K. Dhankar, A. Titarmare, M. Dhone, S. Thakur and P. Saraf, "Quantum-Enhanced AI Robotics for Sustainable Agriculture: Pioneering Autonomous Systems in Precision Farming," 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), Nagpur, India, 2024, pp. 1-7
13. Yoon, S.S.; Kim, D.Y.; Kim, K.K.; Euom, I.C. Vulnerability Exploitation Risk Assessment Based on Offensive Security Approach. Appl. Sci. 2023, 13, 12180.
14. Roy, S.; Sharmin, N.; Acosta, J.C.; Kiekintveld, C.; Laszka, A. Survey and taxonomy of adversarial reconnaissance techniques. ACM Comput. Surv. 2022, 55, 1–38