International Journal of Research in Engineering and Innovation (IJREI)

Journal home page: https://ijrei.com/

RESEARCH PAPER

Man-in-the-middle and spamming in IoT Networks

Arpit Mittal, Manas Aswal, Arman Raza, Priyanshi Bhatt, Rahul Shivhare

Department of Information Technology, Meerut Institute of Engineering and Technology, Meerut, India

Article Information

Received: 07 April 2026
Revised: 27 April 2026
Accepted: 01 May 2026
Available online: 03 May 2026

Keywords:

Deepfake Detection
CNN
Video Frame Analysis
Feature Extraction
Computer Vision
Digital Forensics

Abstract

The Internet of Things (IoT) has become an essential element of modern technology, facilitating the connection and automatic data exchange among devices such as wearable technology, smart appliances, and sensors. These technologies assist in optimizing everyday tasks and enhancing productivity in various fields. Nevertheless, IoT networks often face security challenges because of the large volume of connected devices, most of which possess insufficient computing capabilities. Man-in-the-Middle (MITM) attacks and spamming threats represent two significant risks. A Man-in-the-Middle (MITM) attack occurs when an attacker secretly intercepts or modifies the communication between two devices without the knowledge of the victims. The act of sending excessive or harmful data is referred to as spamming, which can lead to network slowdowns or disrupt normal operations.

1. Introduction

It refers to a network of interconnected physical devices that are capable of collecting, processing, and exchanging data through the internet without requiring direct human interaction. These devices are embedded with sensors, software components, communication modules, and computational capabilities that enable them to monitor environmental conditions, process information, and communicate with other systems [1, 2].

IoT technology has enabled the development of smart environments in which everyday objects can communicate intelligently and autonomously. Examples of IoT devices include smart home appliances such as thermostats, lighting systems, and security cameras; wearable devices such as smartwatches and fitness trackers; industrial monitoring equipment used in manufacturing plants; healthcare monitoring systems; and connected vehicles used in modern transportation networks [3]. IoT devices communicate with one another using a range of specialized protocols designed to operate efficiently in low-power and resource-constrained environments. Among the most commonly used protocols is HTTP (Hypertext Transfer Protocol), which enables web-based communication between devices and servers. MQTT (Message Queuing Telemetry Transport) is another widely adopted protocol, known for its lightweight design and suitability for low-bandwidth communication in IoT systems. Additionally, CoAP (Constrained Application Protocol) is specifically developed for constrained devices, offering efficient web transfer capabilities with minimal overhead. Underpinning these protocols is the TCP/IP (Transmission Control Protocol/Internet Protocol) suite, which forms the foundation of internet communication [4]. The rapid expansion of IoT technology has brought transformative changes across various industries. Recent technology reports indicate that billions of IoT devices are currently deployed worldwide, and this number is expected to increase significantly in the coming years.

1.1 Security Challenges in IoT Networks

Even though IoT technology offers many benefits, it also brings several security challenges. Most IoT devices are built with limited hardware capabilities, such as low memory, less

Corresponding author: Arpit Mittal
Email Address: arpitmittal2206@gmail.com
https://doi.org/10.36037/IJREI.2026.10202

processing power, and restricted battery life. Due to these constraints, implementing robust security measures such as advanced encryption, adequate authentication, or intrusion detection systems becomes challenging. Another concern is that IoT devices frequently operate in diverse environments where devices from various manufacturers utilize different protocols and standards [5]. This absence of uniformity can lead to compatibility issues and heighten the risk of security vulnerabilities. In numerous instances, IoT devices are set up with default usernames and passwords, which users often neglect to change. This situation facilitates easier access for attackers to the devices with minimal effort. Furthermore, the majority of IoT communication occurs via wireless networks, which are more susceptible to threats such as data interception or eavesdropping. Due to all these factors, IoT networks have become an easy target for attackers. They can take advantage of these weaknesses to steal data, gain unauthorized access, spread malware, or disrupt normal network operations.

1.2 Man-in-the-Middle (MITM) Attacks in IoT Networks

One of the most significant security threats in IoT networks is the Man-in-the-Middle (MITM) attack, in which an attacker secretly intercepts communication between two devices without their knowledge. In this scenario, the attacker positions themselves between the sender and receiver, enabling them not only to monitor the exchanged data but also to modify it or inject malicious commands. Such attacks are particularly dangerous because they are difficult to detect, as both communicating devices assume, they are directly connected while, in reality, the attacker is controlling the interaction. Several techniques are commonly associated with MITM attacks in IoT environments. ARP spoofing involves sending forged Address Resolution Protocol messages to associate the attacker's MAC address with the IP address of another device on the network, thereby redirecting traffic. DNS spoofing, on the other hand, manipulates domain name system responses to redirect users or devices to malicious servers instead of legitimate ones, potentially leading to data theft or malware injection [6]. Packet sniffing is another technique used to capture and analyze data packets traveling across the network, allowing attackers to extract sensitive information from IoT communications. Additionally, session hijacking occurs when an attacker takes over an active communication session by obtaining session IDs or authentication tokens, enabling them to impersonate legitimate users and gain unauthorized access. Collectively, these threats highlight the critical need for robust security mechanisms in IoT systems to ensure data integrity, confidentiality, and secure communication.

1.3 Spamming Attacks in IoT Networks

Another prevalent risk in IoT networks is spamming attacks. These attacks consist of transmitting a significant volume of unsolicited or malicious messages to devices within the network. In conventional computing systems, spam is typically associated with unsolicited emails. Nevertheless, in Internet of Things (IoT) environments, spamming can manifest in various ways. This may involve transmitting a large volume of data packets, inserting malicious commands, or producing automated messages from devices that have already been compromised.

1.4 Need for Intelligent Attack Detection Systems

Conventional security approaches, including firewalls, antivirus programs, and rule-based intrusion detection systems, often prove inadequate for contemporary IoT networks. These systems typically rely on established attack patterns, which can result in their inability to identify novel or unfamiliar threats. To address this limitation, machine learning serves as a more effective solution. Machine learning models possess the ability to analyze extensive volumes of network data and detect anomalous patterns that could signify suspicious behavior. By utilizing historical network data to train these models, it becomes feasible to develop systems capable of identifying attacks in real-time, rather than depending solely on established rules. Employing machine learning for intrusion detection provides numerous advantages. It is capable of identifying novel or unfamiliar attacks, enhancing overall precision, minimizing the likelihood of false positives, and managing substantial amounts of data with greater efficiency.

The primary objective of this research is to develop a machine learning-driven system capable of identifying Man-in-the-Middle (MITM) and spamming attacks within IoT networks. The system operates by examining the network traffic produced by IoT devices and employing machine learning methods to categorize the communication as either normal or suspicious [7]. By integrating intelligent detection techniques with ongoing network surveillance, this strategy seeks to enhance the overall security and dependability of IoT systems.

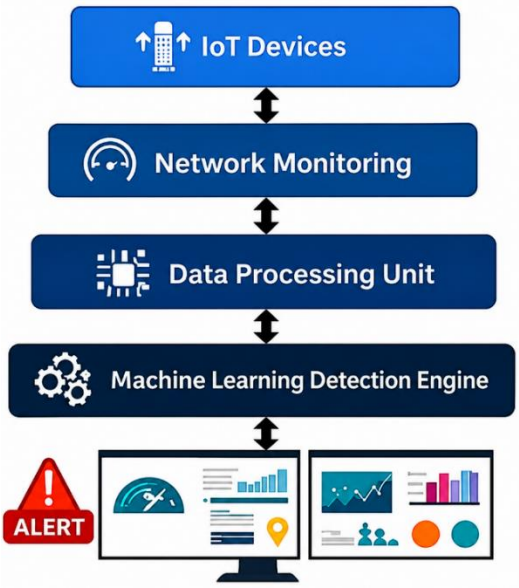

Figure 1: Proposed architecture system

The suggested system is structured as an integration of real-time network surveillance and machine learning-driven

detection is shown in Fig. 1. Its primary objective is to consistently monitor the traffic produced by IoT devices and to detect any irregular or questionable behavior. The system monitors network communications and analyzes packet-level data to identify patterns that could suggest an attack. The comprehensive operation of the system can be comprehended through the subsequent steps. Initially, network traffic is gathered from IoT devices. Subsequently, significant features are extracted from the acquired data. Following this, the data undergoes cleaning and normalization to ensure its effective utilization. Machine learning models are then developed to categorize the traffic as either normal or malicious. Ultimately, if any suspicious behavior is detected, the system issues alerts.

2. Research Gap

The rapid advancement of IoT technology has attracted considerable attention from researchers, particularly in the domain of cybersecurity. Numerous studies have proposed diverse techniques for detecting cyber threats such as Man-in-the-Middle (MITM) attacks, Distributed Denial of Service (DDoS) attacks, malware propagation, and general network intrusions. Despite these advancements, several critical challenges remain unresolved. One major limitation in existing research is that many systems are designed to detect only a single type of attack at a time. For example, some models focus exclusively on MITM attacks, while others are tailored specifically for DDoS detection. However, in real-world IoT environments, multiple types of attacks often occur simultaneously, including spoofing, packet interception, botnet activity, and spam traffic. As a result, single-attack detection systems may fail to provide comprehensive security coverage. Additionally, some studies are restricted to specific communication protocols. Certain models are developed exclusively for protocols such as MQTT or Wi-Fi, whereas modern IoT networks typically operate across multiple protocols simultaneously, including HTTP, MQTT, CoAP, and BLE. Consequently, protocol-specific systems may overlook threats transmitted through other communication channels, highlighting the need for more versatile and integrated security solutions.

3. Dataset and Data Collection

To evaluate the effectiveness of the proposed system, network traffic was collected from a simulated Internet of Things (IoT) environment that comprised devices such as sensors, gateways, and client systems. Throughout this procedure, two distinct categories of network conditions were established:
To evaluate the effectiveness of the proposed system, network traffic was collected from a simulated Internet of Things (IoT) environment consisting of sensors, gateways, and client systems. During this process, two distinct network conditions were established: normal traffic and attack traffic. Normal traffic represented routine operations such as sensor data transmission, device authentication, and standard communication between devices using protocols like HTTP and MQTT. In contrast, attack traffic was generated to assess system performance under various threat scenarios. Multiple types of cyberattacks were deliberately introduced, including ARP spoofing, DNS spoofing, packet interception, and spam packet injection. Network data was captured using tools such as Wireshark and Tcpdump, which recorded packets flowing through the system. The captured data was then organized into a structured format, and several relevant features were extracted for further analysis, including source IP address, destination IP address, packet size, protocol type, connection duration, and packet flags.

4. Research Methodology

The primary objective of this research is to create a machine learning-driven system capable of identifying Man-in-the-Middle (MITM) and spamming attacks within IoT networks. To accomplish this, a systematic approach was adopted, beginning with the collection of network data and culminating in the assessment of the models' performance.
The procedure encompasses multiple phases, including the collection of network traffic, the cleansing and preparation of data, the selection of relevant features, the training of machine learning models, and ultimately, the evaluation of their performance.
In summary, the methodology can be perceived as a pipeline in which data is initially gathered from IoT devices, subsequently transformed into a structured format, and ultimately analyzed through machine learning techniques to identify potential attacks.

4.1 Data Processing

The data collected from IoT environments is often unstructured and not immediately suitable for analysis, as it may contain duplicate records, missing values, and irrelevant information that can adversely affect the performance of machine learning models. Therefore, data preprocessing is a vital step before the implementation of any algorithm, with the primary aim of cleaning, organizing, and transforming the data into a usable format for model training. In this study, several preprocessing techniques were employed to prepare the dataset for further analysis. Duplicate records, which commonly arise in network packet captures due to repeated transmissions or logging errors, were identified and removed to prevent bias and improve the model's ability to generalize new data. Categorical features present in network traffic data, such as protocol types (TCP, UDP, HTTP, MQTT), were converted into numerical form using encoding techniques including label encoding and one-hot encoding, as machine learning algorithms require numerical inputs. Furthermore, since different features may have varying numerical scales—for instance, packet size and packet frequency—feature normalization was performed using Min-Max scaling to ensure balanced contribution from all variables.

4.2 Feature Selection

Feature selection is an important part of the machine learning process because not every feature plays a useful role in detecting malicious network activity. Some features may not add any meaningful information and can even make the model

slower or less accurate. Therefore, it is necessary to choose only those features that actually help in identifying attacks. In this research, various network-related characteristics were analyzed to determine which are most effective in identifying MITM and spamming attacks. Only the most pertinent features were chosen for additional processing. For instance, the frequency of packet transmission was taken into account since spam attacks typically produce a substantial volume of packets within a brief period. Additionally, the type of protocol employed in communication was significant, as simplicity in understanding and interpretation. It is capable of managing both numerical and categorical data types with relative ease. Nevertheless, as the dataset grows in complexity, the model may attempt to fit the data excessively, which can impair its performance on new or previously unseen data.

4.3 Machine Learning Models

In this study, multiple machine learning algorithms were implemented and evaluated to detect malicious network traffic, with a particular focus on identifying Man-in-the-Middle (MITM) and spamming attacks. The objective was to compare the performance of different models and determine their effectiveness in classifying normal and attack traffic. Among these, the Decision Tree algorithm was employed as a supervised learning technique widely used for classification tasks. It functions by constructing a tree-like structure, where each internal node represents a decision based on a specific feature, and each branch corresponds to an outcome of that decision, ultimately leading to a classification result at the leaf nodes. This hierarchical decision-making process enables the model to effectively distinguish between normal and malicious behavior.

One of the key advantages of the Decision Tree model is its simplicity and ease of interpretation, making it suitable for analyzing network traffic data. It can efficiently handle both numerical and categorical features without requiring complex transformations. However, as the complexity of the dataset increases, the model may become prone to overfitting, which can reduce its ability to generalize well on unseen data.

During preprocessing, feature normalization was also applied to ensure consistency in data scale, particularly when handling features with varying ranges. This was achieved using Min-Max normalization, expressed as:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Where x is the original value, x_{min}= min value of the data set

This normalization technique helps in scaling features to a uniform range, thereby improving the performance and stability of machine learning models.

Random Forest is an ensemble machine learning technique that builds upon the concept of decision trees by utilizing multiple trees instead of relying on a single model. Each tree in the forest is trained on a different random subset of the dataset, allowing the model to capture diverse patterns and relationships within the data. During prediction, all individual trees generate their outputs, and the final result is determined through majority voting. This approach significantly improves the model's stability, accuracy, and robustness. One of the key advantages of Random Forest is its ability to reduce overfitting compared to a single decision tree, making it highly effective for handling large and complex datasets commonly found in real-world IoT environments.

Support Vector Machine (SVM) is another powerful classification technique used in this study to distinguish between normal and malicious network traffic. It operates by identifying an optimal boundary, known as a hyperplane, that separates data points belonging to different classes. The primary objective of SVM is to maximize the margin between classes, ensuring clear separation and improved classification performance. In the context of this work, SVM helps in detecting abnormal or suspicious network behavior by effectively distinguishing it from regular traffic patterns.

Logistic Regression is also employed as a classification method due to its simplicity and effectiveness in probability-based predictions. It estimates the likelihood that a given input belongs to a particular class using a mathematical function that maps input features to output probabilities. This makes it particularly useful for binary classification problems, such as identifying whether network activity is normal or malicious. Its interpretability and efficiency make it a valuable baseline model for comparison with more complex algorithms.

$$P(y=1) = \frac{1}{1+e^{-z}}$$

Where $z = w_1x_1, w_2x_2, w_3x_3 \ldots\ldots\ldots\ldots\ldots\ldots w_nx_n$

4.4 Model Training

Following the preparation of the dataset and the selection of key features, the subsequent step involved training the machine learning models. To achieve this, supervised learning techniques were employed, enabling the models to learn from labeled data. The dataset was subsequently divided into two segments. Approximately 70% of the data was utilized for training the models, whereas the leftover 30% was reserved for evaluating their performance. Throughout the training process, the models acquired the ability to distinguish between typical network activity and harmful behavior by analyzing the patterns found in the data. To enhance the reliability of the models, cross-validation was implemented as well. In this approach, the dataset is segmented into several smaller sections, and the model undergoes training multiple times with various combinations of these sections. This process contributes to the improvement of overall performance and minimizes the risk of overfitting.

Fig. 2 illustrates the comparative performance of four machine learning models—Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression—based on key evaluation metrics: accuracy, precision, recall, and F1-score. All models demonstrate relatively similar performance, with accuracy values consistently around 85%, indicating that each model is capable of correctly classifying a high

proportion of network traffic instances. The recall values are slightly higher (around 86%) across most models, suggesting that they are effective in identifying actual attack instances. However, precision values are comparatively lower (approximately 82–83%), indicating the presence of some false positives in the predictions.

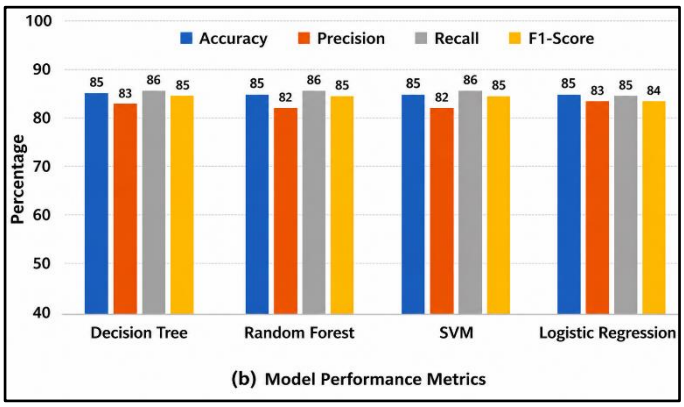

Figure 2: Model Performance Metrics of Machine Learning Algorithms

The F1-score, which represents the balance between precision and recall, remains stable around 84–85% for all models. Overall, the results show that while all four models perform reliably, Decision Tree and Random Forest exhibit slightly better balance in performance metrics, whereas SVM and Logistic Regression also provide competitive and consistent results, making them suitable for IoT-based intrusion detection systems.

4.5 Confusion matrix

A confusion matrix is a fundamental evaluation tool used to measure the performance of a classification model by comparing its predicted outcomes with the actual values in the dataset. It is typically organized in a tabular form consisting of four key components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). True Positives represent instances where the model correctly identifies a positive class (e.g., correctly detecting an attack), while True Negatives indicate correctly identified negative instances (e.g., normal traffic correctly classified). False Positives occur when the model incorrectly predicts a positive class for a negative instance (false alarm), and False Negatives occur when the model fails to detect a positive instance (missed detection). By analyzing these four components, the confusion matrix provides a comprehensive understanding of the model's strengths and weaknesses, enabling the calculation of important performance metrics such as accuracy, precision, recall, and F1-score. This makes it an essential tool for evaluating and improving classification models, particularly in applications like IoT security where both detection accuracy and error minimization are critical.

Table 1 presents the confusion matrix for binary classification of network traffic into attack and normal categories. It compares the model's predictions with actual outcomes. When an actual attack is correctly identified as an attack, it is termed a True Positive (TP), whereas if it is incorrectly classified as normal, it becomes a False Negative (FN), indicating a missed detection. Similarly, when normal traffic is wrongly classified as an attack, it is called a False Positive (FP), representing a false alarm, and when it is correctly identified as normal, it is termed a True Negative (TN). This table provides a clear framework for evaluating the model's performance in terms of detection accuracy and error rates.

Table 1: Confusion Matrix for Binary Classification (Attack vs Normal Traffic)

	Predicted Attack	Predicted Normal
Actual Attack	True Positive (TP)	False Negative (FN)
Actual Normal	False Positive (FP)	True Negative (TN)

Fig. 3 presents a confusion matrix that illustrates the performance of the classification model in distinguishing between normal and malicious network traffic. The matrix is divided into four quadrants representing True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP) outcomes. The model correctly identified 3200 instances of normal traffic as normal (True Negatives), indicating strong performance in recognizing benign activity. It also successfully detected 2530 instances of malicious traffic as attacks (True Positives), demonstrating effective threat detection capability. However, there are 150 instances where normal traffic was incorrectly classified as malicious (False Positives), which may lead to unnecessary alerts. Additionally, 120 malicious instances were misclassified as normal (False Negatives), representing missed detections that could pose security risks. Overall, the confusion matrix indicates that the model performs well with high correct classifications, although there is still scope for improvement in reducing false alarms and missed attacks.

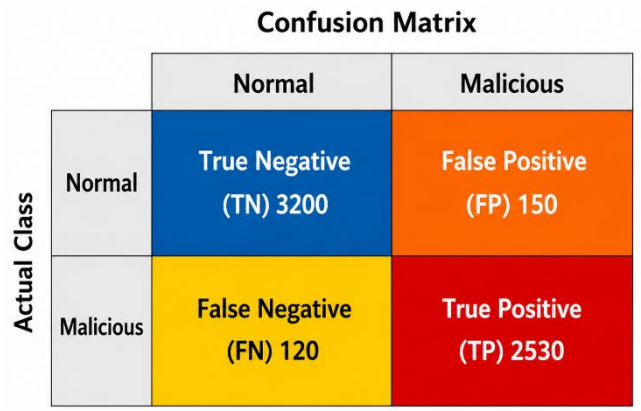

Figure 3: Confusion Matrix for Classification of Normal and Malicious Traffic

In order to evaluate the performance of the classification models, several standard metrics were used, each providing a different perspective on how well the model distinguishes between attack and normal traffic. Accuracy represents the overall correctness of the model and is calculated as the ratio of correctly predicted instances (both attacks and normal traffic) to the total number of instances. It is given by:

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN}$$

where TP (True Positives) are correctly detected attacks, TN (True Negatives) are correctly identified normal instances, FP (False Positives) are normal instances incorrectly classified as attacks, and FN (False Negatives) are attacks that the model failed to detect.

Precision measures the accuracy of positive predictions, indicating how many of the predicted attacks are actually attacks:

$$\text{Precision} = \frac{TP}{FP+TP}$$

A high precision value means fewer false alarms. Recall, also known as sensitivity, measures the model's ability to correctly identify actual attacks:

A high recall indicates that most attack instances are successfully detected. To balance both precision and recall, the F1 Score is used, which is the harmonic mean of the two:

$$F1 = 2\times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$$

This metric is particularly useful when there is an uneven class distribution. Additionally, the False Positive Rate (FPR) measures the proportion of normal instances incorrectly classified as attacks:

e positive ratio (FPR) = FPFP+TNThese evaluation metrics were used to compare the performance of different machine learning models. Ultimately, the model demonstrating the highest accuracy along with balanced precision, recall, and F1-score was selected as the most reliable for detecting attacks in the IoT network.

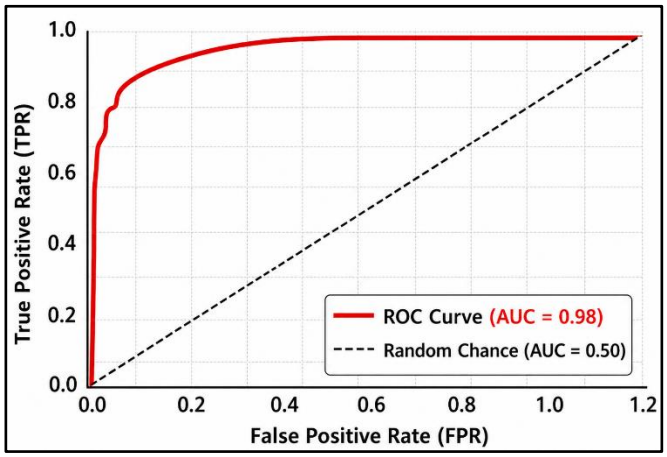

Figure : Receiver Operating Characteristic (ROC) Curve for Model Performance Evaluation

Fig. 4 represents the Receiver Operating Characteristic (ROC) curve, which is used to evaluate the performance of the classification model in distinguishing between normal and malicious network traffic. The ROC curve plots the True Positive Rate (TPR), also known as recall, against the False Positive Rate (FPR) at various threshold settings. The red curve represents the model's performance, while the diagonal dashed line indicates random chance (AUC = 0.50). The model achieves an Area Under the Curve (AUC) of 0.98, which is very close to 1, indicating excellent classification capability. This means the model can effectively differentiate between attack and normal traffic with high accuracy. The curve's position near the top-left corner further signifies a high true positive rate with a low false positive rate, reflecting strong detection performance and minimal false alarms.

4.6 Methodology Workflow

The working process of the proposed system can be explained in a series of steps.

- Initially, network traffic is gathered from the IoT environment utilizing packet capturing tools. This encompasses both standard communication data and traffic produced during attacks.
- Upon gathering the data, the unprocessed packets are arranged into a suitable dataset to facilitate subsequent analysis.
- Subsequently, the dataset undergoes cleaning and preparation. Any extraneous or absent information is addressed, and the data is modified to ensure that all values fall within a comparable range.
- Once the data has been prepared, the most relevant features are chosen. These features assist in distinguishing between normal and malicious traffic.
- Subsequently, various machine learning models are trained utilizing the prepared dataset. These models identify patterns within the data to differentiate between safe and harmful network behaviorThe trained models are then tested and compared using standard evaluation methods to check their performance.
- Ultimately, the model that provides the highest accuracy and reliability in results is selected for identifying attacks within the IoT network.

5. Results and Discussion

Table 2 represents a comparative analysis of different machine learning algorithms based on key performance metrics: accuracy, precision, and recall. Among all the models, Random Forest demonstrates the highest performance, achieving an accuracy of 96.4%, precision of 95.7%, and recall of 96.1%, indicating its strong capability in correctly identifying both attack and normal traffic with minimal errors. Support Vector Machine (SVM) also performs well, with balanced metrics across all categories, making it a reliable choice for classification tasks.

Table 2: Performance Comparison of Machine Learning Algorithms

Algorithm	Accuracy (%)	Precision (%)	Recall (%)
Decision Tree	92.3	91.8	90.9
Random Forest	96.4	95.7	96.1
Support Vector Machine	93.8	92.5	93.2
Logistic Regression	89.7	88.9	88.5

The Decision Tree model shows good performance but slightly lower recall, suggesting it may miss some attack instances. On the other hand, Logistic Regression records the lowest values

among the models, indicating comparatively lower effectiveness in detecting malicious activity. During the testing phase, it was noted that the models were able to distinctly identify atypical packet behavior linked to MITM attacks, including unforeseen delays and inconsistent communication patterns. The Random Forest algorithm demonstrated superior performance primarily due to its utilization of multiple decision trees rather than depending on a solitary tree. This approach facilitates more balanced decision-making and diminishes the likelihood of errors. It was observed that spam-related attacks generated a significant volume of packets in a brief period. This abrupt surge in traffic facilitated the model's ability to differentiate between legitimate and harmful activities.

The findings derived from this research distinctly demonstrate that machine learning methodologies can improve the security of IoT networks. The models successfully identified atypical patterns in network traffic, thereby facilitating the effective detection of potential attacks. Among all the algorithms, the Random Forest model exhibited superior performance in comparison to others. This is primarily due to its integration of multiple decision trees, which aids in minimizing errors and enhancing overall prediction accuracy. Nonetheless, it is crucial to emphasize that the experiments took place in a controlled setting. In actual IoT networks, traffic behaviors may exhibit greater complexity and unpredictability. Consequently, additional testing in real- world conditions is essential to confirm the efficacy of the proposed system.

6. Conclusion

This research introduced a machine learning-driven method for identifying Man-in-the-Middle (MITM) and spamming attacks within IoT networks. A controlled IoT environment was created to simulate both typical and malicious network activities. Different types of attacks, including spoofing and spam traffic, were generated to assemble a dataset for analysis. The gathered network data underwent processing and was utilized to train various machine learning models. Of all the algorithms implemented, the Random Forest classifier demonstrated the highest performance regarding accuracy and dependability. It successfully identified unusual traffic patterns while keeping the rate of false detections low. The findings of this study suggest that machine learning methods can significantly contribute to enhancing the security of IoT systems. Through the analysis of network behavior and the identification of atypical patterns, the proposed system aids in the early detection of potential threats. In summary, this strategy offers a practical and effective means of improving the safety of IoT networks.

References

[1] Saed, M., & Aljuhani, A. (2022). Detection of man-in-the-middle attack using machine learning. In *2022 2nd International Conference on Computing and Information Technology (ICCIT)* (pp. 388–393). IEEE. https://doi.org/10.1109/ICCIT52419.2022.9711555.

[2] Sasi, T., Lashkari, A. H., Lu, R., Xiong, P., & Iqbal, S. (2024). A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges. *Journal of Information and Intelligence, 2*(6), 455–513. https://doi.org/10.1016/j.jiixd.2023.12.001

[3] C, A. S., Vijayalakshmi, A., Broody, J., Sathishkumar, J. S., Abishek, B., & K, S. P. (2023). Detection of man-in-the-middle attack in 5G IoT using machine learning. In *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)* (pp. 1–5). IEEE. https://doi.org/10.1109/ICRASET59632.2023.10420166.

[4] Sultan, A. B. M., Mehmood, S., & Zahid, H. (2022). Man-in-the-middle attack detection for MQTT-based IoT devices using different machine learning algorithms. In *2022 2nd International Conference on Artificial Intelligence (ICAI)* (pp. 118–121). IEEE. https://doi.org/10.1109/ICAI55435.2022.9773590.

[5] Vennam, P., Mouleeswaran, S. K., Shamila, S., & Kasarla, S. R. (2022). A comprehensive analysis of fog layer and man-in-the-middle attacks in IoT networks. In *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)* (pp. 1–5). IEEE. https://doi.org/10.1109/MysuruCon55714.2022.9972612.

[6] N, S., & S, K. (2023). Fuzzy logic-based man-in-the-middle attack detection and improving routing efficiency in the IoT network. In *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICAISC58445.2023.10200105.

[7] Sharma, A., Babbar, H., & Vats, A. K. (2024). A supervised machine learning framework for early detection of man-in-the-middle attacks. In *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICIPTM59628.2024.10563778.

Cite this article as: Arpit Mittal, Manas Aswal, Arman Raza, Priyanshi Bhatt, Rahul Shivhare, Man-in-the-middle and spamming in IoT Networks, International Journal of Research in Engineering and Innovation Vol-10, Issue-2 (2026), 44-50. *https://doi.org/10.36037/IJREI.2026.10202*