International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(19s) 2026 pp. 704-716
https://doi.org/10.70917/ijcisim-2026-5077
Article

A Hybrid Machine Learning-Based Smart Home IoT Security System for Threat Detection, Classification, and Alert Mechanisms

P. Praveen[1], K Sridhar[2], B. Srinivasa Rao[3], Vemula Sridhar[4], D. Bikshalu[5], Shaik Saddam Hussain[6]

[1]Assistant Professor, CSE, VNR Vignana Jyothi Institute of Engineering and Technology, Vignana Jyothi Nagar, Pragathi Nagar, Nizampet, Hyderabad, Telangana - 500118, E-mail: praveen_p@vnrvjiet.in
[2]Associate Professor, Dept of CSE, School of CSE, Malla Reddy (MR) Deemed To Be University, Medchal, Telangana, INDIA. E-mail: sridhark529reddy@gmail.com
3 Assistant Professor, Dept: CSE -Data Science, Guru Nanak Institutions Technical Campus, Ibrahimpatnam Ranga Reddy District, Hyderabad, Telangana,501506. Email: bsrao.cnu26@gmail.com
[4]Associate Professor, Department of CSE, MVSR Engineering College, Nadergul, Hyderabad – 501510, E-Mail: sridhar_cse@mvsrec.edu.in
[5]Assistant Professor, CSE, KG Reddy College of Engineering and Technology, Moinabad, Ranga Reddy District, Hyderabad, Telangana 501504, Email: bikshalu.d@gmail.com
[6]Assistant Professor, CSE, VNR Vignana Jyothi Institute of Engineering and Technology, Vignana Jyothi Nagar, Pragathi Nagar, Nizampet, Hyderabad, Telangana - 500118, E-mail: saddamhussainshaik573@gmail.com.

Abstract: The growing usage of Internet of Things gadgets has facilitated smart environments including automated homes, smart lighting systems, and smart industrial applications. In spite of these advantages, the IoT devices are frequently not well secured because of their poor computing power and compact architecture. This makes them very susceptible to cyberattacks that may cripple the functionality of their devices, interfere with network processes or reveal confidential information. The identification of malicious activity on the IoT networks has thus become a key challenge to cybersecurity. This paper has discussed a machine learning-driven intrusion detection system that aims to detect attacks on IoT devices. An actual IoT system based on ESP32 microcontrollers and a smart bulb was used to create network traffic both in the regular operation environment and in the simulated attack environment. To enhance the efficiency of the model, the data obtained on the traffic was subjected to various preprocessing steps such as data cleaning, feature encoding, normalization and feature selection. Local Outlier Factor (LOF) based anomaly detection method was employed to determine abnormal network behavior with a random forest classifier used to determine the category of attack. The system is constantly watching the IoT traffic and sends automatic email notifications in case of suspicious activity. The outcomes of the experiment suggest that machine learning methods can be successfully used to differentiate between legitimate and malicious network behavior, which can be used as a viable solution to enhance the security and surveillance of IoT-based systems. The proposed system achieved an accuracy of 88.22%, precision of 90.41%, recall of 85.51%, and an F1-score of 87.89%, demonstrating effective detection of malicious IoT network activity.
Keywords: NA

1. INTRODUCTION

The fast advancement of the Internet of Things (IoT) has dramatically altered the communication and interaction patterns of the devices in the contemporary setting. IoT technology can be used to make devices including smart bulbs, sensors, cameras, and home automation systems connect to the internet and exchange data in an automatic manner. These interrelated devices enhance convenience, automation and efficiency in many fields such as smart homes, healthcare and industrial systems **[5], [6]**. Nevertheless, there are also serious cybersecurity issues that are brought about by the rising number of interconnected devices. A diverse range of IoT devices is built with a limited

amount of computing power and insignificant in-built security, which creates exposure to various cyber-attacks **[13], [15]**.

The conventional security tools like firewalls and signature-based intrusion detection systems may not be enough to secure IoT networks. These techniques are primarily based on preprogrammed attack patterns and have a hard time detecting emerging or changing attacks. Cyber-attacks are increasingly becoming complex and this has led to the increased demand of smart detection systems that can detect abnormal network activities in real time. The techniques of machine learning are a good solution because they examine the patterns of network traffic and automatically identify normal and malicious actions **[14]**.

In this publication, an intrusion detection method is created using machine learning to identify attacks on IoT devices. An actual IoT setup that included ESP32-based devices linked to a smart bulb was utilized to record network traffic under regular condition and in the simulated attack conditions. The obtained data were fed through processing to train detection models in which Local Outlier Factor (LOF) was implemented in detecting anomalies and a Random Forest classifier was employed in classifying the detected attacks. The system is also capable of tracking the network activity and automatically generating email notifications in case of suspicious activity, which is an effective system of enhancing the security of IoT settings.[3]

The remaining sections of this paper are organized as follows: Section I discusses the existing work related to IoT intrusion detection systems, Section II presents the proposed methodology, Section III describes dataset utilization and preprocessing techniques, Section IV explains the evaluation metrics, Section V details the system architecture and working modules, Section VI provides model comparison and selection, and Section VII presents the experimental results.

2. EXISTING WORK

The initial studies conducted on intrusion detection in the IoT networks were primarily based on rule-based and signature-based intrusion detection systems. These methods identify malicious traffic by matching network traffic to known attack signatures **[1], [2]**.

Although they are efficient in detection of known threats, they are not very effective in detection of new or changing attacks as most of the time attackers change their behavior in order to avoid detection by the already existing signatures. With the increasing popularity of IoT and the increased complexity of network traffic, scholars realized that more dynamic security systems were necessary that could perform analysis of patterns in network traffic instead of just depending on established rules **[3]**.

To address these shortcomings, numerous researches started to use machine learning to detect intrusion. Support Vector Machines (SVM), Decision Trees, Random Forest, and k-Nearest Neighbors are the machine learning models that have been extensively used in classifying the network traffic as normal or malicious. These algorithms are trained on historical network data and they are able to detect suspicious behavior even in cases where attack signatures are not determined explicitly **[16], [17], [18]**. It has been demonstrated that these models can attain high detection accuracy and false alarms are minimized when conditioned on benchmark datasets, including NSL-KDD, CICIDS2017, and UNSW-NB15.

More recent works have examined the advanced and hybrid methods, which combine machine learning with deep learning strategies to enhance the detection performance in IoT settings. Neural networks, convolutional neural networks (CNN) and ensemble learning models are all techniques that can be deployed to extract complicated patterns in network traffic data and detect anomalies more efficiently **[21], [23], [24]**. There are also some studies that came up with hybrid models, which combine anomaly detection with classification algorithms to detect known and unknown attacks better **[38], [39]**.

The above developments indicate that machine learning-based intrusion detection systems are increasingly gaining relevance in enhancing security in contemporary IoT networks.

3. PROPOSED WORK

The proposed system will focus on designing a smart system of intrusion detection that can detect the presence of malicious activities that target Internet of Things (IoT) devices. To create a typical network of smart devices, a real IoT test setup was created using ESP32 microcontrollers attached to a smart bulb. Network traffic was recorded in normal operation and simulated attack conditions so as to create a dataset that would reflect the various types of IoT network behavior. The data obtained then was preprocessed by applying various preprocessing techniques such as data cleaning, categorical attributes encoding, data normalization of the numeric attributes and feature selection to enhance the performance of the detection models. [Fig 1]

Several anomaly detection algorithms were tested and compared in order to identify the best method of detection. Various machine learning models were trained and tested using the processed data so as to examine their capability to identify abnormal network behavior. The accuracy, precision, recall, and F1-score were used as conventional measures of the performance of these models. [Fig 1]

According to this comparative analysis, the best anomaly detection model was chosen in order to identify the suspicious network activity and then a Random Forest classifier was used to classify the identified malicious traffic. The system will be monitoring network traffic and send automated email alerts in case of any suspicious activity.[Fig 1]

Fig 1. Work Flow

- **Isolation Forest**

Reason for Selection:

Isolation Forest was chosen due to its effectiveness in identifying anomalies in a large dataset. It is specifically applicable in intrusion detection activities where the abnormal events are few relative to the normal data. The algorithm is computationally efficient and it is able to detect anomalies without significant data preprocessing.

Mechanism:

Isolation Forest Isolation Forest operates by building several random decision trees by recursively splitting the data. Anomalous data points are most likely to be isolated with less splits in the tree structure since they are anomalous and not similar to the normal data. The algorithm computes a score of anomaly, which is the average path length to isolate a data point. The shortness of the path length implies increased chances of being an anomaly. [Fig 2]

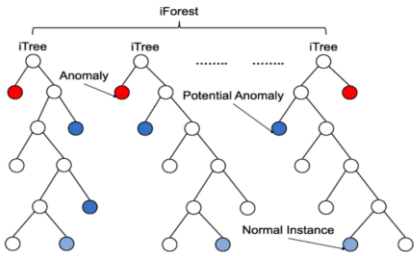

Fig 2. Isolation Forest

- **One-Class Support Vector Machine**

Reason for Selection:

One-Class SVM was added since the model is useful in the situation when the model is majorly trained with normal data. This renders it appropriate to intrusion detection systems where the abnormal traffic is comparatively uncommon or hard to get in the course of training.

Mechanism:

One-Class SVM is used to learn a perimeter that encloses the normal data points in a high dimensional feature space. In the process of detection, any piece of information that falls beyond this threshold is regarded as an anomaly. The model basically determines abnormalities of learned normal behavior patterns. [Fig 3]

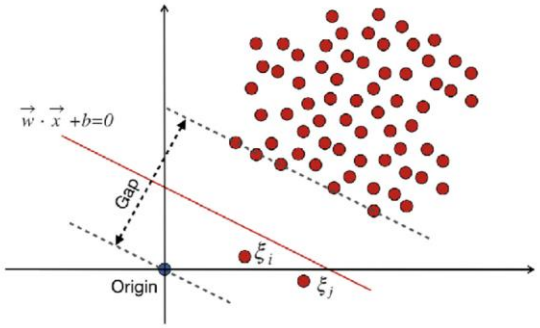

Fig 3. One Class SVM

- **Local Outlier Factor (LOF)**

Reason for Selection:

The choice of Local Outlier Factor is explained by the fact that this method identifies anomalies when there are differences in local density. The method is also quite effective in detecting subtle anomalies, which might not be easily detected by the global statistical approach.

Mechanism:

LOF measures the concentration of each point on its neighbors. In case the density of a specific observation is much less in comparison to the neighbors, then it is regarded as an outlier. Such local density comparison enables the algorithm to identify the abnormal traffic behavior in groups of the normal data. [Fig 4]

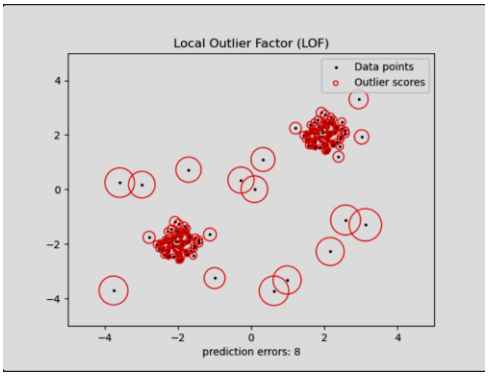

Fig 4. Local Outlier Factor

- **Elliptic Envelope**

Reason for Selection:

Elliptic Envelope was also added as a statistical method of anomaly detection, which assumes the data is normally distributed. It offers a probability method to determine severe deviations of the normal data distribution.

Mechanism:

The algorithm approximates a dataset of elliptical boundary with the help of robust covariance estimation. The points that are not within this elliptical boundary are considered as anomalies and it means that there might be malicious network activity. [Fig 5]

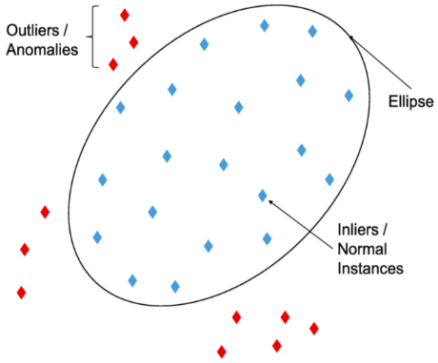

Fig 5. Elliptic Envelope

- **Autoencoder**

Reason for Selection:

Autoencoder was chosen in order to check the performance of the deep learning-based methods of anomaly detection. It can also learn complicated trends in information and detect deviation in the form of reconstruction errors.

Mechanism:

An Autoencoder is a type of neural network that reduces input data to a lower-dimensional form and then reconstructs the input data to its original form. When it is trained using normal data, it learns to reproduce similar patterns correctly. This is however, not the case when abnormal data has been given and the reconstruction error is much higher. This is a reconstruction error that is applied to detect network traffic anomalies. [Fig 6]

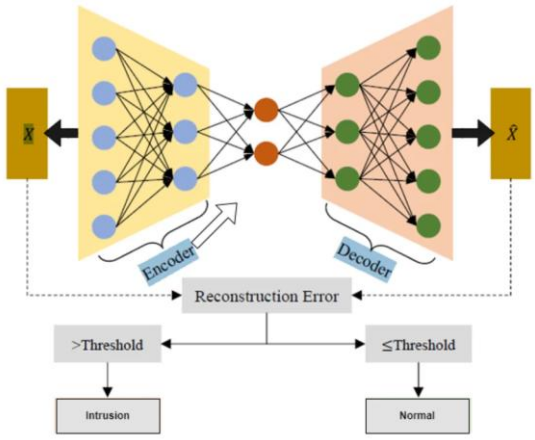

Fig 6. Auto Encoder

4. DATASET UTILIZATION AND PREPROCESSING

The data set in this research was done with a real IoT setting which is comprised of ESP32 devices that were attached to a smart bulb. Network traffic of normal functioning of devices and simulated attacks were recorded to depict various behaviors of the IoT network. The data gathered had several network-related attributes that characterized the patterns of communication such as packet statistics, protocol data and aspects of connections. To transform the dataset into an acceptable model of machine learning, a number of preprocessing procedures were used to clean the data and enhance its quality.

Data Cleaning

The preprocessing phase consisted of the initial phase of cleaning the dataset by eliminating useless or empty columns that were not relevant to the detection of attacks. Columns with missing or meaningless values were removed to minimize noise in the dataset. This allowed simplifying the dataset and also made sure that only useful attributes were not left out in the further analysis.

Categorical Feature Encoding.

The dataset had categorical attributes which meant that these attributes had to be transformed into numeric values in order to be processed using machine learning algorithms. Label encoding was used to convert categoric values into numerical values. This process was converted in order to enable the models to learn and interpret patterns through these attributes in the course of training.

Correlation Analysis

The correlation analysis has been conducted on the dataset to determine the highly correlated features. Two or more features may carry a very similar information thus can create a redundancy leading to adverse effects on the model performance. A correlation matrix was created, and those features whose correlation was greater than a pre-set value were dropped. The process trimmed redundancy of features and enhanced computational power.

Missing Value Handling

In real-life data, missing data may influence the model performance as well as training stability. To solve this problem, the missing values were imputed, through median imputation method, whereby the median of each separate feature was taken to fill in the missing ones. This method maintains the statistical qualities of the data and the dataset is not lost.

Feature Scaling

Numerical features were then normalized using StandardScaler after missing values were taken care of. The feature scaling is used to convert the data to a similar range and distribution of all the features. The reason why this step is significant is that most machine learning algorithms operate well when input features are normalized.

Feature Selection

To enhance the work of the models further and decrease the number of dimensions the SelectKBest method

was implemented with mutual information as a scoring criterion. This method compares the correlation of each

feature to the target label and only the most informative features are selected. The dataset will also be more

efficient to be trained by keeping the highest-ranked features and still retains the most relevant information to

obtain accurate attack detection.

5. EVALUATION METRICS

To assess the effectiveness Accuracy, precision, recall, and the F1 Score are some of the key performance

metrics used to evaluate how well machine learning models detect botnet attacks. These metrics offer a

Extensive understanding of the model's ability to differentiate between malicious and legitimate network traffic.

★ Accuracy- Overall Classification Capability:

The percentage of total predictions that the model correctly predicts is known as accuracy. It is computed as follows:

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

- **TP (True Positive)**: Malicious traffic accurately recognized as malicious
- **TN (True Negative)**: Benign traffic accurately recognized as benign
- **FP (False Positive)**: Benign traffic incorrectly flagged as malicious
- **FN (False Negative)**: Malicious traffic wrongly labeled as benign

★ Precision — Reducing False Alarms:

The accuracy of the model's positive predictions is the main focus of precision

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

High precision indicates that the model is probably right when it marks a flow as malicious. This reduces false

alarms in cybersecurity and makes sure security analysts don't waste time looking into non-threat

★ Recall — Detecting Every Threat:

Recall emphasizes the model's ability to recognize actual threats and is also known as sensitivity or true positive rate:

$$Recall = \frac{TP}{TP + FN}$$

In intrusion detection, where even one missed attack can have dire repercussions, a high recall guarantees that the majority of botnet attacks are identified, lowering the possibility of allowing malicious flows to go unnoticed.

★ F1 Score — Harmonizing Precision and Recall

Precision and Recall's harmonic mean is the F1 Score

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

6. SYSTEM ARCHITECTURE AND METHODOLOGY

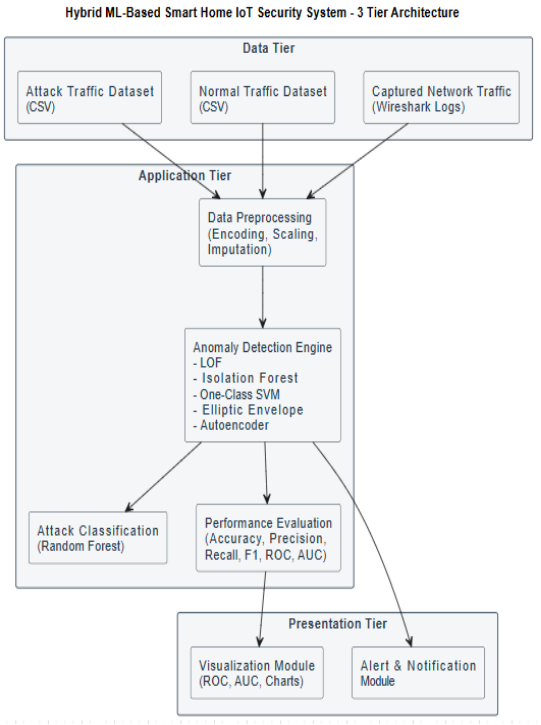

Fig 7. System Architecture

The intrusion detection system proposed is based on various functional modules that collaborate towards gathering network data, processing the data, abnormal behavior detection and alerts. Every module carries out a particular operation within the larger detection pipeline towards having proper and effective monitoring of the traffic of the IoT network.

Data Collection Module

The data collection module will have the role of capturing network traffic produced by the IoT environment. In this project, the simulated real-world IoT communication was carried out by using an IoT system comprising of ESP32 chips that interacted with a smart bulb. Datasets were captured and stored with network packets that were generated during normal operation of the device and in the case of a simulated attack. The datasets have different network characteristics, communication patterns, packet statistics, connection features, etc. applied to the subsequent analysis. [Fig 7]

Data Preprocessing Module

Preprocessing module pre-processes the data that has been collected to be processed by the machine learning. Raw network traffic data is frequently missing values, redundant features, and categorical features which require conversion prior to model training. This module undertakes a number of pre-processing tasks such as data cleaning, encoding of categorical data, missing values, scaling of features, and selecting features. These measures are useful to enhance the quality of data, minimize noise, and make sure that the dataset can be utilized to train machine learning models. [Fig 7]

Feature Selection Module

The feature selection module determines the most relevant attributes which have great contribution to attack detection. Even in most datasets, not all features will be of any value and could create a redundancy. The feature selection methods are applied in order to save only the most informative features and eliminate less useful ones. It simplifies the computations and increases the accuracy of the machine learning models by targeting the most significant attributes of data.

Anomaly Detection Module

The anomaly detection module is tasked to detect the abnormal modes in the network traffic. Isolation Forest, One-Class SVM, Local outlier factor, Elliptic envelope and Autoencoder are some of the anomaly detection models that were tested to ascertain their efficacy in this system. These models examine the traffic patterns of the network and identify the abnormalities. Having compared, the most effective model is identified and the suspicious network activity is identified. [Fig 7]

Attack Classification Module

After abnormal traffic has been identified, the attack classification module identifies the type of attack. This purpose is explained by the fact that a Random Forest classifier possesses a robust predictive power and is capable of handling complex data sets. The classifier takes the observed suspicious traffic and classifies it as a particular attack type depending on patterns learnt during training. [Fig 7]

Alert and Monitoring Module

The last component of the system will be the one that monitors the threats identified and informs the users of possible security threats. The system will send an alert message with the relevant information of the identified threat whenever suspicious activity or an attack is detected.

These warnings are automatically emailed to the user in form of email. This module makes sure that the administrators are aware of the possible attacks on the spot to take action and ensure safety of the IoT environment. [Fig 7]

7. MODEL COMPARISION AND SELECTION

Model Comparison

To determine the most appropriate anomaly detection method to be used in detecting malicious IoT network traffic, several machine learning models were analyzed and compared. It was compared to the Isolation Forest, One-Class Support Vector Machine (One-Class SVM), Local Outlier Factor (LOF), Elliptic Envelope, and Autoencoder

models. These algorithms identify abnormalities in various ways, based on the learning strategies and statistical methods.

Isolation Forest always identifies oddities by subdividing the dataset randomly. The reason behind this is that anomalous points are very rare and are usually outliers of normal data points, and thus they are usually isolated more rapidly than normal data points. One-Class SVM operates by training the border which surrounds normal data in a high-dimensional feature space and classifying those data which lie above this border and are considered anomalies. The Elliptic Envelope algorithm is based on the assumption that the normal data is distributed as a Gaussian distribution and it identifies anomalies by finding points which are out of the determined distribution boundary.

Local Outlier Factor (LOF) is used to detect the anomalies based on the local concentration of the data points with the surrounding points. When the density of a point is much less than that of the surrounding, the point is an outlier. Autoencoder is another model that is a product of deep learning and is used to identify anomaly by learning to replicate normal patterns of data. The error rate becomes very large when the abnormal data is introduced as input implying the possibility of abnormalities in the dataset.

The models were tested on standard evaluation metrics which included accuracy, precision, recall and F1-score. A comparison between these metrics in the various models allowed selecting the most effective algorithm to detect abnormal network behavior in the IoT setting.

Best Model Selection

According to the comparative analysis of the discussed models, the Local Outlier Factor (LOF) algorithm proved the strongest results in terms of recognizing the abnormal network traffic patterns. The global density variations are well captured by LOF, and thus the slight anomalies are identified, and they might not be identified by global statistical models. The feature is especially applicable to the IoT network environment where deviant traffic patterns can be similar to normal communication patterns. [Fig 8]

The LOF was chosen as the main model of anomaly detection within the proposed system because it has better detection ability and can be regularly used regardless of the metrics applied to its evaluation. Upon detecting an anomalous traffic by the LOF model, a Random Forest classifier is used to analyze the detected anomalies to further label them as attacks of a certain type. This two-step detection method enhances the security and reliability of the intrusion detection system, besides offering an opportunity to monitor the IoT network protection well. [Fig 8]

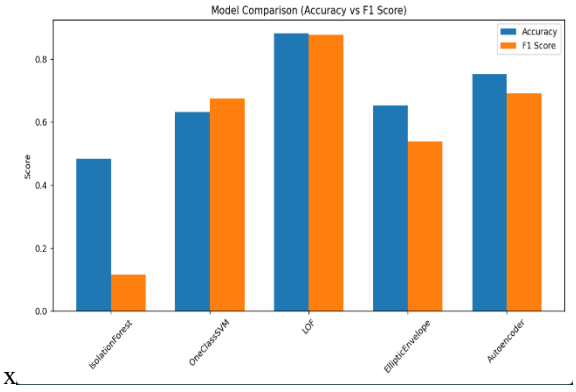

Fig 8. Model Comparison

8. Results

Once the suggested intrusion detection framework had been adopted, the results of various models of anomaly detections were assessed on the basis of the processed IoT network traffic dataset. The models that were compared in this paper were Isolation Forest, One-Class Support Vector Machine (One-Class SVM), Local Outlier Factor (LOF), Elliptic Envelope, and Autoencoder. All the models were trained using the normal network traffic data and were tested

on a dataset that included normal and malicious traffic samples. The effectiveness of each model in the process of identifying abnormal network activity was measured using standard evaluation metrics like accuracy, precision, recall and F1-score.

The experiment findings established that all the tested models could identify the anomalies in the IoT network traffic to a certain level. Nonetheless, their performance differed with regard to the detection mechanism that each algorithm employed. Isolation Forest and One-Class SVM also exhibited sound results in the recognition of abnormal traffic behavior with Elliptic Envelope indicating moderate results because of its assumption of data of a normal distribution.

Autoencoder model could detect anomalies based on reconstruction error, however, it relied on training conditions and the network structure.

The Local Outlier Factor (LOF) algorithm was the most suitable algorithm in the overall evaluation because it was the most successful in identifying anomalies in the IoT network traffic dataset. LOF was effective to record local density variations of the data and could detect subtle differences of normal patterns of communication. Because of this strength, LOF generated better accuracy and high detection reliability than the others that were tested.[Fig 9]

Once the anomalous traffic was identified through the LOF model, the malicious traffic was identified and then the Random Forest classifier was applied to categorize the malicious traffic detected as a particular attack of a specific category. [Fig 10]

The predictive performance of the classifier was high because it has an ensemble learning mechanism and can process complicated datasets. The system was able to detect malicious activities in the IoT environment by using the combined method of anomaly detection and attack classification.

As well as detecting and classifying of attacks, the system was also able to create automatic email notifications whenever a suspicious activity was identified. This real-time alert system allows the administrators to react to the possible threat rather fast and enhances the overall surveillance of the IoT network. The findings prove that the suggested machine learning-powered framework is an effective and scalable solution to detecting cyber-attacks against IoT devices. [Fig 11]

```
=== LOF Detection Performance ===
Accuracy: 0.8822142821696194
Precision: 0.904139776483551
Recall: 0.8551012306470822
F1 Score: 0.8789370329754406
```

Fig 9. LOF Detection Performance

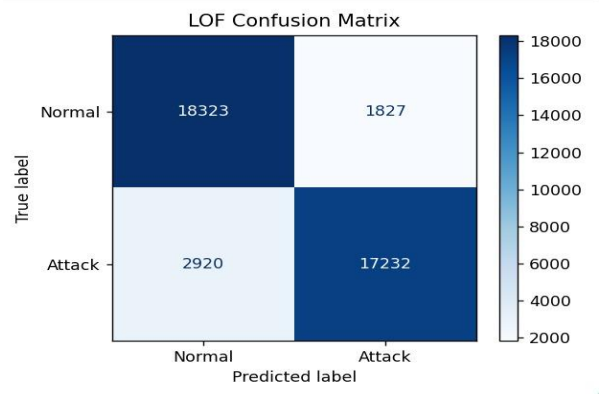

Fig 10. LOF Confusion Matrix

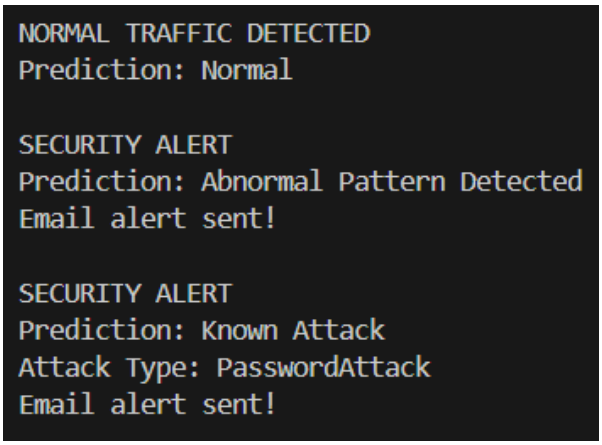

Fig 11. Detection Results

Methods	**Accuracy (%)**			
IoT Security Testbed System [1]	79.50	81.20	75.40	78.20
Smart Home Traffic Analysis [4]	82.10	83.70	80.30	81.90
Machine Learning IDS [16]	84.60	86.10	82.50	84.20
Deep Learning-based IDS [21]	86.80	88.40	84.90	86.60
Proposed IoT IDS (LOF + RF)	88.22	90.41	85.51	87.89

Table 1. Performance Comparison

REFERENCES

1. O. A. Waraga, M. Bettayeb, Q. Nasir, and M. Abu Talib, “Design and implementation of automated IoT security testbed,” Computers & Security, vol. 88, 2020.
2. S. Siboni et al., “Security Testbed for the Internet of Things,” arXiv:1610.05971, 2016.
3. Z. Fang et al., “IOTA: A Framework for Analyzing System-Level Security of IoTs,” IoTDI, 2022.
4. O. Alrawi et al., “YourThings: A Comprehensive Annotated Dataset of Smart-Home Device Traffic,” ACSAC, 2022.
5. Y. Li, A. M. Mandalari, and I. Straw, “Who Let the Smart Toaster Hack the House?,” arXiv:2306.09017, 2023.
6. M. H. Mazhar and Z. Shafiq, “Characterizing Smart Home IoT Traffic in the Wild,” arXiv:2001.08288, 2020.
7. R. Trimananda et al., “PingPong: Packet-Level Signatures for Smart Home Device Events,” NDSS, 2020.
8. W. Zhang et al., “HoMonit: Monitoring Smart Home Apps from Encrypted Traffic,” ACM CCS, 2018.
9. NDSS Supplemental: Packet-Level Signatures for Smart Home Devices, 2020.
10. IoT Inspector: Crowdsourcing Labeled Network Traffic from Smart Home Devices, arXiv, 2019.
11. O. A. Waraga et al., “Automated IoT Security Testbed,” Elsevier, 2020.
12. A Testbed for Security and Privacy Analysis of IoT Devices,” Conference Paper.
13. OWASP IoT Project & NIST Guidelines for IoT Security, 2021.
14. I. Ciccio et al., “CICIoT2023: A Real-Time Dataset and Benchmark for IoT Attack Detection,” Sensors, 2023.
15. A. Sivanathan et al., “Experimental Evaluation of Cybersecurity Threats to Consumer IoT Devices.”
16. M. Hodo et al., “Threat analysis of IoT networks using machine learning intrusion detection system,” ICITST, 2016.