Journal of Information Systems Engineering and Management
2025, 10(20s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing IoT Network Attack Detection with Ensemble Machine Learning and Efficient Feature Extraction

*Dr. Mawahib Sharafeldin Adam Boush[1], Dr. Nithinsha Shajahan[2], Ms. Preethi Rajan[3], Najla Babiker[4], Niyasudeen Fazuludeen[5], Jezna Ansari, Jayasuriya Panchalingam[6]

[1]Assistant Professor, Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan Saudi Arabia.
Email ID: mboush@jazanu.edu.sa
[2]Assistant Professor, Department of Computer Science, College of Engineering & Computer Science, Jazan University, Jazan, SaudiArabia
nanvarsha@jazanu.edu.sa
[3]Lecturer, Management Department at Applied College, Jazan University, Jazan, SaudiArabia
prajan@jazanu.edu.sa
ORCID ID : 0000-0003-0613-8126
[4]Department of Computer Science, College of Engineering & Computer Science, Jazan University, Jazan, SaudiArabia
nbabiker@jazanu.edu.sa
[5]Virtual Desktop Infrastructure administrator, Deanship of Human Resources and Technology, Jazan University, Jazan, Saudi arabia.
niyasudeen@jazanu.edu.sa
[6]Department of Computer Science and Engineering, College of Engineering, Trivandrum, Kerala, India
jezna@cet.ac.in
[7]Department of Computer Science, College of Engineering & Computer Science, Jazan University, Jazan, SaudiArabia
jpanchalingam@jazanu.edu.sa
Corresponding author : Dr. Mawahib Sharafeldin Adam Boush, mboush@jazanu.edu.sa

ARTICLE INFO

Received: 15 Dec 2024

Revised: 29 Jan 2025

Accepted: 12 Feb 2025

ABSTRACT

Attacks on Internet of Things (IoT) networks have been on the increase, making their security a top priority. Research in this area is on how to strengthen IoT network security via better attack detection using machine learning (ML) methods. To improve detection performance, this study proposes a novel method that combines effective feature extraction with ensemble ML methods such as XGBoost, LightGBM, and CatBoost. These algorithms are chosen due to their exceptional accuracy in classification tasks and their ability to handle complex, large datasets. The goal of the feature extraction procedure is to improve the efficacy and efficiency of the learning models by capturing essential properties from data collected by IoT networks. This will specifically design the model for IoT attack detection and train and validate it using a publicly accessible dataset from Kaggle. This study will use important metrics such as accuracy, area under the curve (AUC), precision, recall as well as F1-score to assess the effectiveness of the proposed approach in detecting assaults. The proposed method in this article is termed as IoT-SecureNet, which aims to enhance the security of IoT networks through the application of sophisticated ML techniques and the optimization of feature selection.

Keywords: Ensemble Algorithms, Attack Detection, Feature Extraction, Machine Learning (ML), IoT.

1. INTRODUCTION

Electrical equipment may communicate with a server via the IoT without experiencing any human interference. Computers are susceptible to many risks because users may manipulate them remotely from anywhere [1]. IoT is a technology that has gained a lot of traction in both the academic and industrial sectors. IoT aims to establish a single, intelligent network by fusing the physical and cyber worlds. Numerous application domains have utilized this technology in the development of healthcare applications, sensor networks, smart homes, smart cities, corporate networks, smart grid technologies as well as online applications. Protecting devices and networks, preventing assaults on IoT networks, and managing resource-constrained networks are only a few of the numerous security concerns raised by these widely growing applications in several domains [2].

As a precautionary step for massive IoT networks that rely on NetFlow, a Network Intrusion Detection System (NIDS) is provided in this article. This NIDS enhances ML with a tweaked arithmetic optimization algorithm (AOA) to identify the best characteristics. Two of the many ML models trained using the selected seven characteristics are Random Forest and Extra Trees [3]. Using the Bot-IoT dataset, this study compares ML algorithms for binary as well as multi-class classification. The experimental comparisons, used the metrics like recall, F1 score, accuracy, precision as well as log loss to evaluate the above-mentioned ML algorithms. When it comes to distributed denial-of-service (DDoS) attacks against Hypertext Transfer Protocol (HTTP), RF has a 99% success rate [4].

The report delves into the expanding cybersecurity landscape as it pertains to the widely-used IoT technologies, which are more vulnerable to cyberattacks. The widespread use of IoT systems increases the volume of data flow and the complexity of interactions between devices, opening up a number of new possibilities for cybercriminals [5]. There are many advantages to using tiny Internet-connected devices, and they may help individuals be more efficient, but they also present a security risk. Because there are so many IoT devices, malicious systems are always seeking new ways to attack weak systems. This is especially the case with DDoS attacks, which use the interconnection of many devices—including IoT devices—to impersonate bots and flood services with false requests, effectively shutting them down [6].

An existing article presents a ML-based method for detecting DDoS attacks in an IOT controller, which is a component of the Software Defined Networking-Wireless-Sensor-Networks (SDN-WISE) solution. This study has created a testbed environment to mimic the creation of traffic from a DDoS attack as well as included a detection module based on ML into the controller. The SDN-WISE controller incorporates a logging system that records network logs and transforms them into a dataset [7]. Another analysis shows that while ML-based IDS are better at spotting new types of attacks, they also come with a lot of drawbacks, including higher computing requirements, vulnerability to adversarial attacks, problems with scalability, and compromises between accuracy and false positives [8]. The following section details the present work's contribution.

- For IoT network attack detection, the paper proposes employing sophisticated ensemble ML techniques such as XGBoost, LightGBM, and CatBoost.
- This work uses a new feature extraction method to discover essential properties from IoT network data, aiming to increase the accuracy of the models.
- A strong and clearly defined assessment framework is ensured by the use of a publicly accessible Kaggle dataset that is particularly built for IoT attack detection.
- This work tests the proposed IoT-SecureNet method using numerous performance metrics, including F1-score, AUC, accuracy, precision as well as recall to ensure its dependability in detecting IoT attacks.

The structure of the work is as follows: Section 2 reviews some existing works involving IoT networks and ML. Section 3 discusses the details about the proposed technique. Section 4 includes details on the research findings and comments, along with some limitations of the current study. Section 5 concludes the work followed with the references.

2. LITERATURE REVIEW

Islam et al. [9] discussed about healthcare, agriculture, education, transportation, traffic monitoring, utility services, and the supply chain that are just a few of the many areas that have benefited from IoT-based system improvements in recent years. One of the trickiest problems with the IoT is the security concerns brought up by node heterogeneity. Security methods such as authentication, encryption as well as access control are inadequate for IoT devices. As the title suggests, this paper talks about various types of IoT threats and IDS that are based on deep learning (DL). These include deep neural networks (DNN), random forests (RF), support vector machines (SVM), long short-term memory (LSTM), stacked LSTM, bidirectional LSTM (Bi-LSTM), deep belief networks (DBN) as well as decision trees (DT).

Altulaihan et al. [10] defined that the IOT relies on devices that can automatically set themselves up to connect to networks without requiring a lot of user intervention during setup. IoT devices that can self-configure use a combination of protocols, technologies, and automated processes to establish connections to networks, find services, and adjust their settings automatically, all without human interaction or setup. Attackers may breach users' security and privacy in order to steal their personal information, do financial harm, or spy on them. By blocking access to services for legitimate users, DoS attacks are among the most destructive types of attacks against IoT systems. Such a cyberattack has the potential to severely compromise IoT services and smart environment apps on an IoT network.

Almotairi et al. [11] discussed about IoT networks that are vulnerable to intrusions and other security issues because devices use different protocols and have different levels of computing capability. This provides a heterogeneous ML-based stack classifier model for IoT data to tackle the difficulty of intrusion detection (ID) in the IoT. The model explores and improves critical classification metrics for ID of IoT data using feature selection and ensemble modeling. The two main components of this method involve the construction of an ensemble model, which incorporates several classic ML models, and the application of the K-Best technique to select features, specifically the top fifteen most significant ones. When these parts work together, they improve classification performance by using data from chosen features and combining the power of several models.

Karthikeyan et al. [12] discussed about the IoT and Wireless Sensor Networks (WSNs)that are working together to collect, share as well as process data. The primary objective of this collaboration is to enhance data analysis and automation in order to facilitate improved decision-making. The security and reliability of the connected WSN and IoT components rely on implementing protective measures and securing the IoT with the aid of WSN. This study takes a giant leap forward in IoT and WSN security by combining the power of ML with the Firefly Algorithm (FA) in a complementary way. There are two main contributions to this study. Firstly, the suggested FA-ML method may significantly improve the accuracy of ID in the WSN-IoT environment. Second, the combination of ML and the FA has opened up a new dimension for security-focused optimization methods.

Rafique et al. [13] described about IoT that has generated data and connections on a scale never seen before due to its meteoric rise. Anomaly detection is a security tool that helps find and fix anomalies quickly when system behaviour differs from the usual. When AI and the IoT collaborate to improve anomaly detection, it improves the security, efficiency, as well as dependability of IoT systems. AI-powered anomaly detection systems can identify various risks in IoT settings. These threats include brute force attacks, buffer overflows, injections, replay attacks, DDoS attacks, SQL injections, and backdoor exploits. IoT devices must have intelligent IDSs to identify network intrusions or anomalies. Although IoT is being used more and more in many sectors, it also has a broad attack surface, so there are more opportunities for hackers to get in.

Li et al. [14] noted that the distance-based aspect of the cloud computing paradigm could pose challenges in locations where the quality of the Internet connection is insufficient for critical tasks. Edge computing, a distributed computing architecture, aims to achieve near-end data processing and analysis by relocating applications, data, and services from the network's core node to its logical edge node. This reduces the need for processing and analysis on the cloud. By reducing latency, boosting efficiency, and enhancing security, the convergence of the IoT with edge computing might speed up the development of intelligent systems. This article provides an overview of many topics related to the IoT, including its evolution, edge computing's role in IoT monitoring and management, and ML's data analysis and issue detection capabilities.

Alwahedi et al. [15] discussed about cybersecurity concerns that have emerged due to the IoT ecosystem's exponential growth. Several factors contribute to these difficulties, including the diversity of IoT devices, extensive deployment, and intrinsic processing limitations. The dynamic IoT ecosystem is evolving, making it crucial to include new technologies to solve these issues. In order to overcome IoT security challenges, ML, a quickly developing technology, has shown great potential. It has a big impact and pushed cyber threat detection research forward. It also highlights the difficulties and important questions that remain unanswered in this dynamic industry. Table 1 shows the existing works pros and limitations.

Table 1: Existing works Review

Papers and Authors	Method	Advantages	Limitations
Islam et al. [9]	Different ML algorithms	Helps resolve security concerns.	This do not process this data in real time.

Altulaihan et al. [10]	Different ML algorithms	For better protection against DoS attacks, strengthen the security of IoT networks.	Feature selection algorithms are not the focus of this study.
Karthikeyan et al. [12]	FA-ML technique	Dependability and security.	This study considers only some feature selection strategies.

3. PROPOSED METHODOLOGY

Cyberattacks targeting IoT networks are driving the need for more sophisticated security measures. This work proposes a better attack detection system that makes use of efficient feature extraction methods in conjunction with ensemble ML algorithms like CatBoost, LightGBM, and XGBoost termed as IoT-SecureNet. The algorithms' efficacy in classification tasks and their capacity to manage massive datasets were the deciding factors in their selection. Enhancing the system's performance and computational efficiency, efficient feature extraction ensures that the model training relies only on the most crucial features.

This study train the proposed model, IoT-SecureNet, on a publicly accessible Kaggle IoT attack detection dataset. This work assesses the model using various performance measures, including F1-score, AUC, accuracy, precision as well as recall to ensure its robust performance. There are noticeable advances in the method's ability to identify attacks like port scanning, DoS, and other types of malicious behavior that often target IoT networks. This anticipate the proposed method, which leverages the strengths of ensemble learning and feature selection, to provide high detection accuracy, scalability, and real-time applicability for protecting IoT settings. Figure 1 shows the IoT-SecureNet Flow [16].

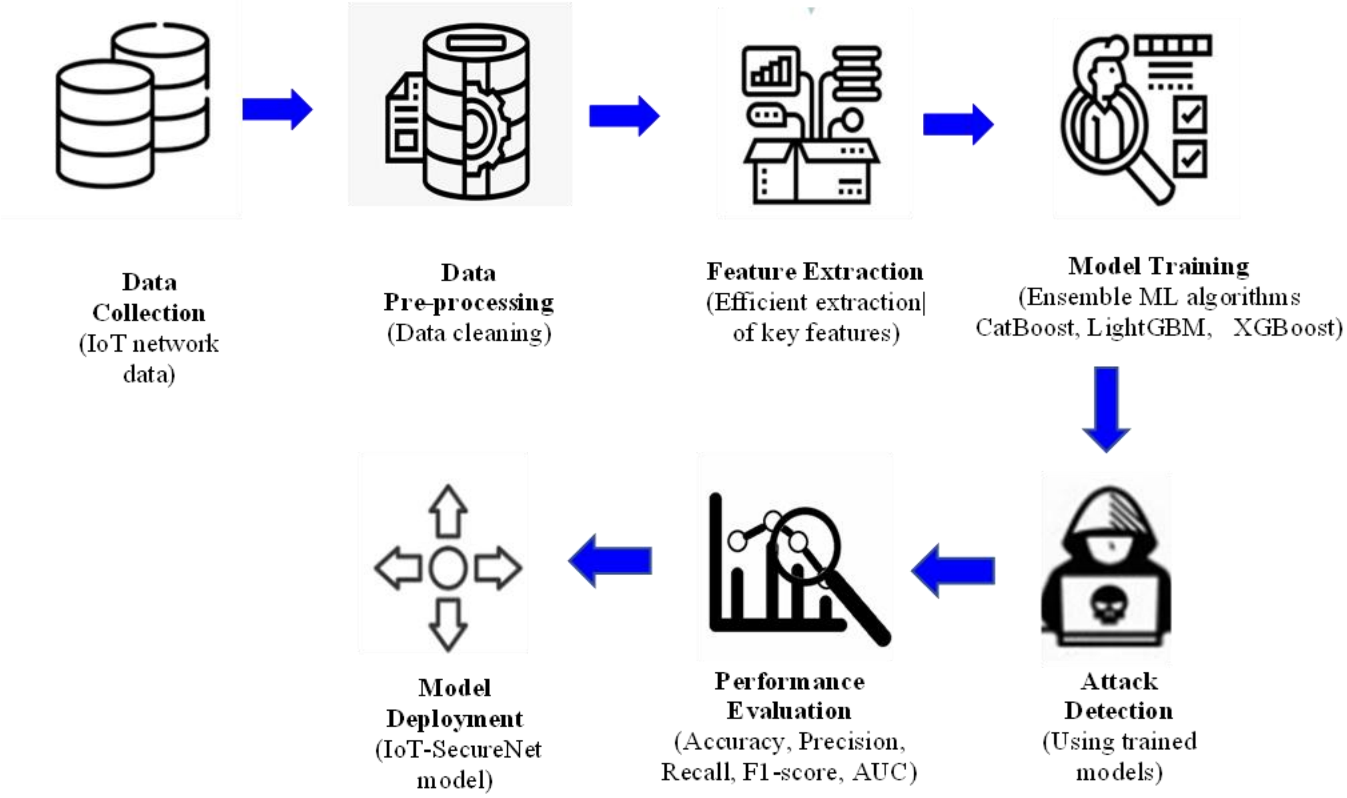
Data Collection (IoT network data)
Data Pre-processing (Data cleaning)
Feature Extraction (Efficient extraction| of key features)
Model Training (Ensemble ML algorithms CatBoost, LightGBM, XGBoost)
Attack Detection (Using trained models)
Performance Evaluation (Accuracy, Precision, Recall, F1-score, AUC)
Model Deployment (IoT-SecureNet model)

Figure 1. IoT-SecureNet Flow

3.1 Dataset Description

This study used the 2017 IoT-23 Dataset from the Canadian Institute for Cybersecurity, available on Kaggle. This work recommends this dataset for attack detection due to its careful preparation for the express purpose of assessing security solutions in IoT networks. The dataset encompasses over 2 million records of network activity from IoT devices, identifying attacks such as port scanning, DoS, and others. An essential component of any classification model for detecting network anomalies is the dataset's inclusion of information such as IP addresses, source, packet sizes as well as destination ports, protocols employed, along with length of connections.

With a wealth of normal and attack data, the IoT-23 dataset provides a great representation of real-world IoT networks. It is possible to train the model on a range of situations since it comprises 23 distinct sorts of attacks. The dataset is valuable for anomaly detection and classification tasks since each instance has several characteristics that reflect network traffic behavior. This dataset allows the model to generalize to other sorts of attacks and network topologies, making it more resilient when used on real-world IoT networks. This study have partitioned this dataset into training and testing sets, labeling the attack classes, to conduct effective supervised learning. This can evaluate the proposed attack detection models using a diverse set of attack situations.

3.2 Preprocessing

In order to get the IoT dataset ready for ML models, preprocessing is an essential step. The preprocessing steps in this study include encoding, feature selection, data cleanup, and normalization.

3.2.1 Data Cleaning:

Missing Data Handling: The dataset manages missing or null values by either removing them or imputing them based on the percentage of missing values. This imputes missing data by substituting the mean of the relevant feature for the missing values, ensuring the datasets remain intact and free of biases.

Outlier Detection: The Z-score approach finds outliers in the data and either deletes or caps them to prevent them from affecting the accuracy of the model. Equation (1) contains the formula that produces the Z-score.

$$Z = \frac{X-\mu}{\sigma} \qquad (1)$$

Where X represents value, μ is the mean of the feature, as well as σ represents standard deviation.

3.2.2 Normalization/Standardization

Min-Max Scaling approach is used to standardize the characteristics within a given range, usually [0, 1]. The Min-Max scaling formula in equation (2) is:

$$X_{norm} = \frac{X-X_{min}}{X_{max}-X_{min}} \qquad (2)$$

Where X represents original value of the feature, X_{min} and X_{min} is the minimum and maximum values of the feature, respectively and X_{min} are the normalized value. This ensures that characteristics with large numerical ranges do not dominate the model's learning process.

3.2.3 Feature Encoding

Categorical features, like protocol type, undergo one-hot encoding. For ML algorithms that use numerical input, this means turning category values into binary vectors.

3.2.4 Feature Selection

Feature selection is done by aid of Recursive Feature Elimination (RFE). Depending on the model's performance, RFE repeatedly deletes the least important features to select the most important ones [17]. The preprocessing stage guarantees that the dataset is clean, consistent, and suitable for training the ML models.

3.3 Feature Extraction

To find the most relevant aspects for IoT attack detection, this work's innovative feature extraction method makes use of statistical and time-domain analysis. Due to the high dimensionality and noisy nature of IoT traffic, it is essential to extract discriminative features that enhance detection accuracy.

3.3.1 Statistical Feature Extraction

Basic statistical features like mean (μ),standard deviation(σ), and variance(σ^2) are calculated for each feature across the dataset. These statistical measures help capture the central tendency and spread of the data, which can be indicative of attack patterns in equation (3):

$$\mu = \frac{1}{n}\sum_{i=1}^{n} X_i, \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2 \qquad (3)$$

Where X_i refers value of the feature and n refers number of data points.

3.3.2 Time-Domain Features

Packet Inter-arrival Time disputes involving denial of service attacks might lead to packet timing issues in equation (4). The inter-arrival time T_{inter} between consecutive packets is calculated to detect anomalies.

$$T_{inter} = T_{arrival}(i + 1) - T_{arrival(i)} \qquad (4)$$

Where $T_{arrival(i)}$ is the timestamp of the i-th packet.

3.3.3 Entropy-Based Features

Shannon Entropy measures the unpredictability in a dataset. For instance, low entropy in IoT traffic could indicate DoS, while high entropy suggests random traffic in equation (5):

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i) \qquad (5)$$

Where $p(x_i)$ are the probability of occurance of feature x_i.

3.3.4 Domain-Specific Features

Some protocols, such as UDP or TCP, are especially vulnerable to particular kinds of attacks. Checking each protocol's packet usage percentage can help distinguish them. This study trains an ensemble of ML models using these retrieved attributes. With feature extraction, this can save important information for attack classification while reducing the dataset's dimensionality.

3.4 Model Training

3.4.1 CatBoost

Given the prevalence of categorical characteristics in data collected from IoT networks, the efficient gradient-boosting technique CatBoost (Categorical Boosting) is well-suited to this kind of data. CatBoost constructs DT sequentially using gradient boosting. Iteratively, it uses the residual mistakes from one decision tree to fit another tree. CatBoost excels in categorisation features compared to its competitors. By taking the target distribution into account for each category, it uses a method known as ordered target encoding to transform categorical variables into numerical features. CatBoost minimizes log-loss functions to optimize the model. These functions are computed as in equation (6):

$$L(\hat{y}, y) = -\sum_{i=1}^{n}\left(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)\right) \qquad (6)$$

Where y_i represents the true label as well as $\hat{y}_i$ is the predicted probability.

There is no need for one-hot encoding or extra preprocessing when using CatBoost to handle categorical information; it does it natively and efficiently. For categorical data, it employs ordered target encoding, a method that lessens the likelihood of overfitting. Using the ordered boosting technique, which takes categorical feature values into account in a bias-free manner during training, CatBoost is able to decrease overfitting, which is its main benefit. For complicated real-world datasets, including both continuous and categorical variables, such as IoT network traffic, this makes CatBoost more resilient. Ideal for real-world datasets with inconsistent or missing values, such as IoT traffic, it automatically resolves missing data and avoids bias in categorical feature encoding.

3.4.2 LightGBM

For a gradient boosting framework that's quick and efficient, especially when dealing with big datasets, LightGBM is a fantastic option. Better accuracy and deeper trees are the results of LightGBM's leaf-wise construction method, as opposed to level-wise. In contrast to conventional level-wise approaches, LightGBM employs a leaf-wise technique that, when split, chooses the leaf with the loss function reduction that is largest. More precise tuning may prevent overfitting, but it achieves better accuracy. To reduce memory use and speed up training, LightGBM employs histogram-based techniques to discretize continuous features into buckets.

In LightGBM, a logistic loss function serves as the goal function for binary classification as in equation (6). LightGBM helps with generalization by using early stops to terminate training when validation set performance begins to decline. LightGBM's capacity to directly handle categorical features is a significant benefit in IoT datasets, and its speed surpasses that of many conventional gradient boosting techniques, particularly when dealing with big datasets.

3.4.3 XGBoost

XGBoost (Extreme Gradient Boosting) is a popular ML technique that excels in speed and accuracy when applied to structured/tabular data [18]. With XGBoost, one may train a cascade of DT, with each tree learning to address the mistakes (residues) made by the ones before it. XGBoost adds a penalty term to the loss function by using L1 (Lasso) and L2 (Ridge) regularizations to reduce overfitting in equation (7).

$$L_{regularized} = L + \lambda \sum_j \theta_j^2 + \alpha \sum_j \left|\theta_j\right| \quad (7)$$

Where L denotes original loss, λ as well as α are the regularization parameters, and θ_j represents the model coefficients. For big datasets in particular, XGBoost's innovative Weighted Quantile Sketch approach effectively finds the optimal split for DT. XGBoost, like CatBoost and LightGBM, uses a log-loss function for binary classification in equation (6).

For optimal performance, XGBoost uses a combination of boosting, regularization, and parallelization. Real-time applications such as IoT network security can leverage its efficiency. XGBoost is a powerful tool for handling complicated and huge datasets since it adheres to the concepts of gradient boosting and uses second-order gradient information to provide more precise updates. XGBoost's mix of speed, high performance, and flexibility has made it a go-to solution for structured/tabular data. Its affinity for parallel processing makes it an ideal component of IoT security systems that operate in real time.

3.5 Ensemble

In order to rise the accuracy along with resilience of IoT network attack detection, this work integrates CatBoost, LightGBM, and XGBoost, three potent gradient boosting methods, into an ensemble model. The ensemble technique leverages the strengths of each model while also mitigating their limitations. Two main phases make up the core technique for merging these models:

On the same IoT attack detection dataset, using the preprocessed and feature-engineered data, CatBoost, LightGBM, and XGBoost are trained independently. Each model learns to categorise network traffic into normal or attack groups depending on its unique strengths, such as managing categorical data (CatBoost), rapid learning on big datasets (LightGBM), and regularization to avoid overfitting (XGBoost).

Voting is done that combine all the models' predictions using a majority voting technique, and each model casts a vote to select the predicted class (attack or no attack). A majority vote finalizes the class prediction. For binary classification jobs, such as attack detection, this approach is both simple and effective.

Weighted Averaging is another option is to average the models' outputs and assign a weight to each model according to how well it performed during training. It uses the weighted average to make the final prediction. Models with a greater weight will significantly influence the final prediction, indicating their superior performance on the validation set. The weighted averaging formula looks like this in equation (8):

$$\hat{y} = \sum_{i=1}^{n} w_i . \hat{y_i} \tag{8}$$

Where w_i is the weight of the $i - th$ model and $\hat{y_i}$ is the predicted probability of the $i - th$ model. A further method is stacking, which involves taking the predictions from three different models (CatBoost, LightGBM, and XGBoost) and feeding them into a meta-learner. This meta-learner is usually a simpler model, such as a DT or logistic regression, and it combines the predictions to get the final prediction.

This work uses these procedures to train the ensemble model. This work separates the dataset into testing as well as training sets to evaluate the model on previously unseen data. Three separate models, CatBoost, LightGBM, and XGBoost, receive training from the training dataset. The models learn patterns in the data related to attack detection [19]. The models make predictions on the test set after training. Voting or weighted averaging, which aggregates the predictions, determines the final classification choice.

4. Results

Using the CICIDS 2017 Attack-23 dataset, this use CatBoost, LightGBM, and XGBoost to identify attacks in IoT networks. This dataset represents a broad variety of security threats connected to the IoT and includes labeled traffic data for both normal and attack scenarios. This study trained and assessed the models after undergoing preprocessing, which included data cleaning, feature selection, and normalisation. By decreasing overfitting and enhancing generalisation, the ensemble technique outperformed individual models in terms of F1-score, accuracy, precision as well as recall.

This study used measures of F1-score, AUC, accuracy, precision as well as recall to compare the three models: CatBoost, LightGBM, and XGBoost. When it came to identifying uncommon forms of attacks, the ensemble technique significantly improved classification accuracy. All experiments conducted in Python utilized libraries such as Scikit-Learn, XGBoost, LightGBM, and CatBoost. This used an Intel i5 processor with 16 GB of RAM for the training. Strong results from the ensemble model validate its use for IoT security in real time.

4.1 Evaluation Metrics

This section evaluates the model's performance using a variety of important metrics. The accuracy of the model quantifies its ability to classify both benign and malicious occurrences. However, it might not be effective in datasets where one class is significantly more prevalent than the other in equation (9).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

Where TP is the True Positives and correctly predicted positive instances, TN denotes True Negatives that is the correctly predicted negative instances, FP represents the False Positives that is the incorrectly predicted as positive and FN denotes False Negatives i.e. incorrectly predicted as negative. Precision shows the accuracy of attack predictions. Since false positives might cause needless alerts or resource allocation, a better level of accuracy is particularly crucial in security applications where they occur in equation (10).

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

Recall evaluates the model's ability to identify all genuine attacks. A greater recall ensures the identification of the majority of attacks by reducing the production of false negatives. Security works generally prioritize recall over accuracy to ensure the identification of crucial attacks in equation (11).

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

The F1 score is useful in balancing memory and accuracy. In unbalanced classification tasks, such as IoT attack detection, it provides a dependable measure for model performance as it produces a single number that captures both the precision and accuracy of the model in equation (12).

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (12)$$

The AUC indicates how well the model can differentiate between non-attack and attack classes. A higher AUC improves performance. The ROC curve, which compares the recall rate (the true positive rate) with the false positive rate, is the basis of this method in equation (13).

$$AUC = \int_0^1 ROC\ curve \qquad (13)$$

When the AUC is near to 1, the model performs very well; when it's close to 0.5, it's as excellent as random guessing. The Matthews Correlation Coefficient (MCC) is a measure of the quality of binary classifications. Here, all four types of results such as the true positives, false positives, and true negatives are considered. MCC excels when working with skewed datasets in equation (14).

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (14)$$

By considering all four types of predictions, MCC gives a fair assessment of the classifier's efficiency. The output may be anything from -1 to 1, with 1 indicating entirely accurate predictions, 0 indicating no better than random classification, and -1 indicating totally wrong predictions.

4.2 Comparison Results

The results of CatBoost, LightGBM, XGBoost, and the proposed IoT-SecureNet on the CICIDS 2017 IoT-23 dataset are shown in the table below. To provide a full depiction of how well each model detects attacks in IoT networks, the evaluation measures include F1-score, AUC, MCC, recall, accuracy as well as precision. Table 2 shows the comparison results.

Table 2. Comparison Result

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC**	**MCC**
CatBoost	95.6	94.3	93.7	94.0	98.2	0.91
LightGBM	94.4	93.8	92.5	93.1	97.9	0.89
XGBoost	94.8	94.0	93.2	93.6	98.1	0.90
IoT-SecureNet	96.2	95.1	94.8	95.0	98.5	0.93

With an accuracy of 96.2%, the IoT-SecureNet model exceeds the individual models (CatBoost, LightGBM, and XGBoost), which stood at 95.6%, 94.4%, and 94.8%, respectively. The results show that the ensemble method greatly enhances overall classification performance, which in turn leads to better attack detection in IoT networks. With an accuracy of 95.1%, IoT-SecureNet outperforms all other models. The proposed model effectively identifies attacks with few false positives, resulting in a low false positive rate. While LightGBM reaches 93.8%, XGBoost 94.0%, and CatBoost 94.3%.

By reducing the amount of false negatives, IoT-SecureNet demonstrates an improved capacity to detect attacks with a recall of 94.8%. With recall ranging from 92.5% (LightGBM) to 93.7% (CatBoost), IoT-SecureNet significantly outperforms the competition. IoT-SecureNet demonstrates an optimal balance between recall and accuracy, achieving an F1 score of 95.0%. This score is greater than CatBoost's (94.0%), LightGBM's (93.1%), and XGBoost's (93.6%), showing that the ensemble correctly detects attacks with a balanced detection while minimizing false positives and false negatives.

While LightGBM has an AUC of 97.9%, XGBoost of 98.1%, and CatBoost of 98.2%, IoT-SecureNet's AUC is 98.5%, the highest of the four models. A greater AUC indicates that the model can more accurately differentiate between benign and malicious traffic, making it more resilient to different types of attacks. The numerical value of the MCC: Among all the models, IoT-SecureNet has the highest MCC value at 0.93. To keep things fair, MCC considers all possible outcomes, including true positives, true negatives, false positives, and false negatives. The fact that IoT-SecureNet has a higher MCC value shows that it is better at telling the difference between attack and normal classes without any bias. This means that the model is more stable and fair.

4.3 Discussion

This work proposes a new method, IoT-SecureNet, which combines effective feature extraction techniques and ensemble learning models to outperform single models like CatBoost, LightGBM, and XGBoost. Feature extraction determines the most important network properties, simplifying the learning process and increasing the model's accuracy in detecting anomalies.

In addition, the model's ensemble structure enables the strengths of each algorithm to synergize, resulting in improved overall performance. Although XGBoost and LightGBM are competitive models on their own, IoT-SecureNet's hybrid approach improves classification power and deals better with the ever-changing nature of IoT network traffic. A more secure IoT network is the result of an all-encompassing strategy, which increases detection rates and improves handling of different sorts of attacks.

5. Conclusion

The IoT-SecureNet ensemble model outperforms all other models in terms of F1-score, AUC, MCC, recall, accuracy as well as precision. The better results show that when CatBoost, LightGBM, and XGBoost are combined, the ensemble can use the best features of each model. This makes it better at finding attacks in IoT networks. The ensemble may be able to identify a wider variety of attacks with fewer false alarms because of its improved recall and accuracy. IoT-SecureNet is an excellent choice for protecting IoT networks. In future, further optimization could enable IoT-SecureNet to identify threats faster in low-latency, real-time IoT systems. DL models like Convolutional Neural Network (CNNs) or Recurrent Neural Network (RNNs) may improve the identification of increasingly complex and diverse IoT traffic data. Creating adaptive models that can respond to different kinds of attacks and network patterns in IoT systems might be a also a future area of research. Considering different device types and traffic scenarios are investigating approaches to scale the model for large-scale IoT networks.

REFERENCES

[1] Haji, Saad Hikmat, and Siddeeq Y. Ameen. "Attack and anomaly detection in iot networks using machine learning techniques: A review." Asian J. Res. Comput. Sci 9, no. 2 (2021): 30-46.

[2] Thakkar, Ankit, and Ritika Lohiya. "A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges." Archives of Computational Methods in Engineering 28, no. 4 (2021): 3211-3243.

[3] Fraihat, Salam, Sharif Makhadmeh, Mohammed Awad, Mohammed Azmi Al-Betar, and Anessa Al-Redhaei. "Intrusion detection system for large-scale IoT NetFlow networks using machine learning with modified Arithmetic Optimization Algorithm." Internet of Things 22 (2023): 100819.

[4] Churcher, Andrew, Rehmat Ullah, Jawad Ahmad, Sadaqat Ur Rehman, Fawad Masood, Mandar Gogate, Fehaid Alqahtani, Boubakr Nour, and William J. Buchanan. "An experimental analysis of attack classification using machine learning in IoT networks." Sensors 21, no. 2 (2021): 446.

[5] Inuwa, Muhammad Muhammad, and Resul Das. "A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks." Internet of Things 26 (2024): 101162.

[6] Alahmadi, Amal A., Malak Aljabri, Fahd Alhaidari, Danyah J. Alharthi, Ghadi E. Rayani, Leena A. Marghalani, Ohoud B. Alotaibi, and Shurooq A. Bajandouh. "DDoS attack detection in IoT-based networks using machine learning models: a survey and research directions." Electronics 12, no. 14 (2023): 3103.

[7] Bhayo, Jalal, Syed Attique Shah, Sufian Hameed, Awais Ahmed, Jamal Nasir, and Dirk Draheim. "Towards a machine learning-based framework for DDOS attack detection in software-defined IoT (SD-IoT) networks." Engineering Applications of Artificial Intelligence 123 (2023): 106432.

[8] Krishnamoorthy, Gowrisankar, and Sai Mani Krishna Sistla. "Exploring Machine Learning Intrusion Detection: Addressing Security and Privacy Challenges in IoT-A Comprehensive Review." Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online) 2, no. 2 (2023): 114-125.

[9] Islam, Nahida, Fahiba Farhin, Ishrat Sultana, M. Shamim Kaiser, Md Sazzadur Rahman, Mufti Mahmud, A. S. M. SanwarHosen, and Gi Hwan Cho. "Towards Machine Learning Based Intrusion Detection in IoT Networks." Computers, Materials & Continua 69, no. 2 (2021).

[10] Altulaihan, Esra, Mohammed Amin Almaiah, and Ahmed Aljughaiman. "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms." Sensors 24, no. 2 (2024): 713.

[11] Almotairi, Ayoob, Samer Atawneh, Osama A. Khashan, and Nour M. Khafajah. "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models." Systems Science & Control Engineering 12, no. 1 (2024): 2321381.

[12] Karthikeyan, M., D. Manimegalai, and Karthikeyan RajaGopal. "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection." Scientific Reports 14, no. 1 (2024): 231.

[13] Rafique, Saida Hafsa, Amira Abdallah, Nura Shifa Musa, and Thangavel Murugan. "Machine learning and deep learning techniques for internet of things network anomaly detection-current research trends." Sensors 24, no. 6 (2024): 1968.

[14] Li, Hanzhe, Xiangxiang Wang, Yuan Feng, Yaqian Qi, and Jingxiao Tian. "Driving Intelligent IoT Monitoring and Control through Cloud Computing and Machine Learning." arXiv preprint arXiv:2403.18100 (2024).

[15] Alwahedi, Fatima, Alyazia Aldhaheri, Mohamed Amine Ferrag, Ammar Battah, and Norbert Tihanyi. "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models." Internet of Things and Cyber-Physical Systems (2024).

[16] Talbi, Mohammed. "Safeguarding IoT Networks Using Machine Learning for Intrusion Detection & Prevention." PhD diss., The George Washington University, 2024.

[17] Alotaibi, Fahad Ali, and Shailendra Mishra. "Cyber Security Intrusion Detection and Bot Data Collection using Deep Learning in the IoT." International Journal of Advanced Computer Science & Applications 15, no. 3 (2024).

[18] Ratnavath, H., and V. Narasimha. "Network intrusion detection using ensemble weighted voting classifier based honeypot and ids framework." Network 52 (2023): 3.

[19] Abinayaa, Sennanur Srinivasan, Prakash Arumugam, Divya Bhavani Mohan, Anand Rajendran, Abderezak Lashab, Baoze Wei, and Josep M. Guerrero. "Securing the Edge: CatBoost Classifier Optimized by the Lyrebird Algorithm to Detect Denial of Service Attacks in Internet of Things-Based Wireless Sensor Networks." Future Internet 16, no. 10 (2024): 381.