DESIGN OF AN INTRUSION DETECTION MODEL FOR IOT-ENABLED SMART HOME

LIBA MARIYAM K (2200705013)
Department of Computer Science
Central University of Kerala
Periye, Kasaragod
liba.2200705013@cukerala.ac.in

Dr.Manohar Naik S
Department of Computer Science
Central University of Kerala
Periye, Kasaragod
manoharamen@cukerala.ac.in

***Abstract*—In the era of the Internet of Things (IoT), security has become a paramount concern due to the increasing number of devices communicating with each other using different protocols. However, these devices often lack the processing power to ensure safety, necessitating a significant improvement in our current methods of protecting IoT networks. Machine learning (ML) has emerged as a promising solution for securing IoT systems. This project presents the design of an intrusion detection model for IoT-enabled smart homes, leveraging machine learning to enhance security. The project employs a range of ML classification algorithms, including Logistic Regression (LR), Random Forest (RF), Light Gradient Boosting Machine (LGBM), and Extreme Gradient Boosting (XGB). These algorithms are used both individually and in ensemble to classify and predict various types of network attacks. The DS2OS dataset, which includes 'normal' and 'anomalous' network traffic, is used for this study. The performance of the classifiers was evaluated based on the accuracy, Receiver Operating Characteristic (ROC), runtime, and confusion matrix. The LGBM ensemble classifier was found to have better performance. The proposed intrusion detection model, "LGB-IDS," was validated using ensemble techniques, such as majority voting. The main objective of this project is to propose the design of an efficient intrusion detection model with high accuracy, better time efficiency, and a reduced false alarm rate. The proposed model achieves an accuracy of 99.5% and has a time efficiency that is much higher than those of other prevalent algorithm-based models.**

***Index Terms*—learning classification algorithms, ensemble classifiers, gradient boosting algorithms, light gradient boosting machines (LGBM) and intrusion detection systems (IDS).**

I. Introduction

In recent years, the proliferation of connected devices in the Internet of Things (IoT) [1] ecosystem has been accompanied by a concerning surge in cybercrimes, particularly targeting IoT-enabled smart home environments. The interconnected nature of these devices, operating on diverse protocols, poses unprecedented challenges to security and privacy. IoT, characterized by a network of devices communicating seamlessly, has become an integral part of modern living. However, the autonomous and expansive nature of IoT networks has drawn the attention of cybercriminals, leading to an increase in intrusions and malicious activities.

Smart homes1, constituting a significant component of the IoT landscape, are particularly susceptible to security breaches, thereby jeopardizing the safety and privacy of IoT consumers [2]. The intrinsic vulnerabilities in IoT networks, coupled with the escalating sophistication of cyber threats, underscore the critical need for robust security measures. Network security and user privacy emerge as paramount concerns, prompting the development of effective Intrusion Detection Systems (IDS) tailored for IoT environments.

An Intrusion Detection System plays a pivotal role in safeguarding IoT systems by monitoring network activities, detecting attack patterns, and scrutinizing user behavior to prevent security violations. The sheer size, autonomy, and enticing features of IoT networks present unique challenges, making it imperative for researchers to devise intelligent solutions that go beyond traditional security measures. The increase in cybercrime within the IoT ecosystem necessitates the development of more sophisticated and efficient IDS to prevent and detect a diverse array of threats.

The primary objective of this project is to address the challenges posed by cyber threats in IoT-enabled smart homes through the design and implementation of a comprehensive Intrusion Detection System [3]. The project is motivated by the need for enhanced security in the face of escalating crime rates, the difficulty of identifying unknown attacks, and the potential harm posed by these attacks. The proposed IDS aims to monitor network and system assets for unexpected activities, identify suspicious behavior, and raise alerts to thwart malicious actors before they compromise the integrity of a network or system.

Recognizing the limitations of conventional investigation processes in identifying unknown attacks, the project leverages machine learning, specifically the Light Gradient Boosting Method (LGBM) [4], to enhance the efficiency of intrusion detection. The focus is on developing a model that not only achieves high accuracy but also demonstrates improved time efficiency, reducing the false alarm rate. In addition to studying various machine learning classifiers and employing ensemble techniques, the project explores dimensionality reduction approaches to streamline the intrusion detection system.

To evaluate the effectiveness of the proposed model, the project utilizes the 'DS2OS' dataset [5], a collection of traces from various smart home devices capturing both normal and anomalous network traffic behaviors. The performance metrics include accuracy, prediction error, Precision, Recall, and F1

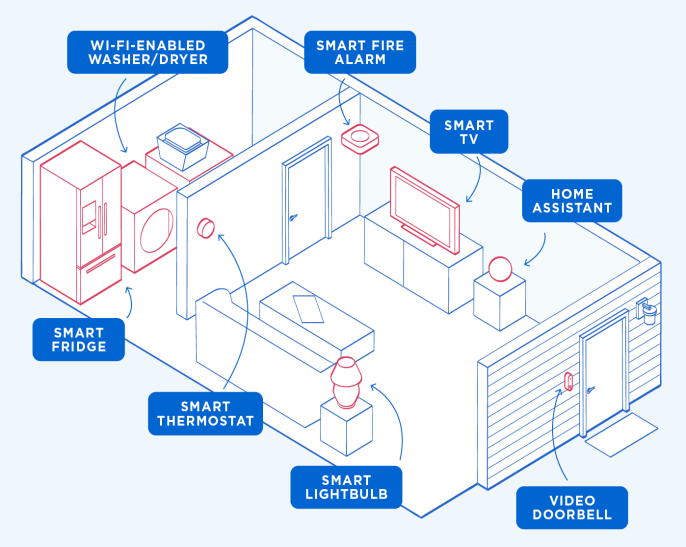


Fig. 1. IOT Enabled Smart Home

Score , among others. The ultimate goal is to contribute to the development of a robust and efficient intrusion detection model tailored for the unique challenges presented by IoT-enabled smart home environments.

II. Related Works

Intrusion detection using machine learning techniques: This paper presents an experimental comparison of various machine learning techniques for intrusion detection. It could be useful for understanding the strengths and weaknesses of different algorithms [6]

Cyber-attack detection system using random forest, KNN, XGBoost algorithms. The proposed work has been evaluated on BoT-IoT and DS2OS datasets. The proposed work claims to achieve 90to 100% detection rate. [7]

Analysis of Machine Learning Algorithms for Anomaly Detection on Edge Devices. In this study, the entire dataset was divided into training (80%) and test (20%) datasets, and then a new smaller dataset was created by selecting the samples randomly for balancing the dataset. Models based on different ML algorithms (LR, DT, SVM, RF) have been compared with ANN algorithm-based models. [8]

Anomaly based network intrusion detection for IoT attacks using deep learning technique.This model was also designed using deep learning. Authors used filter-based feature selection which has been implanted by dropping highly correlated features. [9]

Network Intrusion Detection for IoT Security Based on Learning Techniques.This paper implement their own NIDS and finally to propose new smart techniques in IoT context considering IoT limitations. [10]

Intrusion detection based on machine learning in the Internet of Things, attacks and counter measures. [11]

Building an efficient intrusion detection system based on feature selection and ensemble classifier.The CFS-BA heuristic algorithm for dimensionality reduction in order to select the most relevant and distinct subsets and show correlations between features. The proposed ensemble approach was based on c4.5 and RF by Penalizing Attributes algorithms with an average of probability rule. The probability distributions of base learners were incorporated using voting techniques for better performance of attack recognition. The results of the proposed system were evaluated using NSL-KDD, AWID, and CIC-IDS2017 datasets. [12]

BoostedEnML-Efficient Technique for Detecting Cyberattacks in IoT Systems Using Boosted Ensemble Machine Learning:This paper proposes an efficient method for detecting cyberattacks and network intrusions based on boosted ML classifiers,named BoostedEnML.they trained with different ML classifiers (DT, RF, ET, LGBM, AD, and XGB) and obtain an ensemble using the stacking method and another with a majority voting approach. Two different datasets containing high-profile attacks were used to train, evaluate, and test the IDS model. [13]

An improved anomaly detection model for IoT security using decision tree and gradient boosting,using machine learning (ML) and deep learning (DP) algorithms. [14]

Performance Improvement of Intrusion Detection System for Detecting Attacks on Internet of Things and Edge of Things.Here the light gradient boosting machine, decision tree, gradient boosting machine, k-nearest neighbor, and extreme gradient boosting algorithms were used for classification. [15]

III. Methodology

This section delineates the comprehensive process of designing, analyzing, and implementing the proposed model for an Intrusion Detection System (IDS) [16] in an IoT-enabled Smart Home. The preliminary stages encompass acquiring project dependencies and installing essential libraries to lay the foundation for the subsequent phases.

The construction process of the proposed intrusion detection model is shown in figure 2

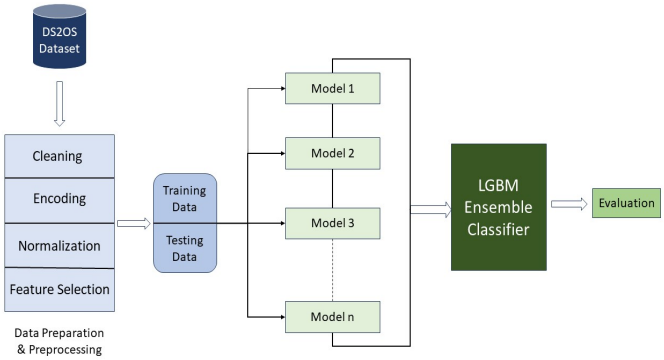


Fig. 2. LGBM-IDS Model

The **DS2OS Dataset** is a significant resource for researchers and developers focusing on intrusion detection systems (IDS)

and security solutions for smart home environments. It was generated in a smart home setting and contains traces from various smart home devices such as light controllers, batteries, washing machines, thermometers, smartphones, smart doors, and movement sensors. The dataset also encapsulates the communication between these IoT devices. The dataset comprises information about the source and destination IP addresses, ports used, protocol, packet size, timestamp, operation performed (e.g., read, write), and the value associated with the operation.This dataset contains 7 malicious classes which include 'DoSattack' (DoS), 'dataProbing' (Probe), 'malitiousControl' (MC), 'malitiousOperation' (MO), 'scan', 'spying' (Spy), 'wrongSetUp' (WS), and one 'normal' class.Fundamental details of DS2OS dataset is given in table I

TABLE I
DS2OS DATASET

Total Features	13
Total Instances	357953
Total Classes	7(attacks)+1(benign)

A. Data Prepration & Preprocessing

1) Data Cleaning: Data cleaning is an essential step in any machine learning project. It involves transforming raw data into a clean and structured format that can be easily analyzed and used for training machine learning models.

- Removing irrelevant, redundant, and less useful instances: This step is necessary because it helps to reduce the complexity of the dataset, which can make the learning process more efficient and the results more reliable.

- Filling missing values: Missing values can hinder the performance of machine learning models if not handled properly. Therefore, it is crucial to fill these missing values with appropriate values to maintain the integrity of the dataset.

- Removing duplicates: Duplicate entries can skew the results of the machine learning model, leading to inaccurate predictions. Therefore, it is important to remove duplicate entries from the dataset.

2) Data Transformation: Data transformation is another crucial step in the preprocessing of the DS2OS dataset [17]. It involves converting the data into a format that can be easily understood by machine learning algorithms. [18]

- Label encoding: This technique is used to convert categorical data into a format that could be provided to machine learning algorithms to improve prediction. It is important because it helps to convert categorical data into a numerical format that can be easily processed by machine learning algorithms.

- One-hot encoding: This technique is used to convert categorical data into a binary vector format.Apply one-hot encoding to handle categorical variables with multiple categories. This ensures that the model can effectively interpret and utilize categorical information.

-Normalization: Apply normalization to the encoded dataset to scale and standardize it.This is particularly important in machine learning and statistical modeling, where the scale of variables can greatly influence the performance of algorithms. In this paper we use Min-Max Normalization It involves transforming the data into a range of [0,1].

$$X_{\text{normalized}} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

3) Feature Engineering: Feature selection helps address this problem by reducing the dimensionality of the dataset, thereby improving computational efficiency and the performance of machine learning models. It also enhances the interpretability of the model by reducing the number of features, making it easier to understand the underlying relationships and insights

Univariate feature selection is a method used to select the most important features in a dataset. The idea behind this method is to evaluate each individual feature's relationship with the target variable and select the ones that have the strongest correlation. This process is repeated for each feature and the best ones are selected based on defined criteria, such as the highest correlation or statistical significance.

Here we are using the SelectKBest method from the "sklearn.feature selection_module" to perform univariate feature selection [19].

B. Classification

After preprocessing the data and selecting the best features, we can train the model. We have tried several classifiers, including Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), and LightGBM (LGBM). For each classifier, we need to fit the model to our training data.A brief discription of these classification alogorithms given below.

Logistic Regression (LR): Logistic Regression is a statistical model that uses a logistic function to model a binary dependent variable. It works by fitting the best line to the data, and then transforming the output using the logistic function to get a probability. The output of the logistic function is then thresholded to give a binary output. The model parameters are estimated using maximum likelihood estimation

Random Forest (RF): Random Forest is an ensemble learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Each tree is built from a subset of the training data and a random subset of the features. This randomness is what makes the algorithm robust and less prone to overfitting.

XGBoost (XGB): XGBoost stands for eXtreme Gradient Boosting. It is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XGBoost builds an ensemble of decision trees in a stage-wise fashion, and it generalizes to other optimization objectives in addition to maximum likelihood estimation. It also introduces several new techniques to handle overfitting, such as regularization, and it has built-in cross-validation to tune parameters.

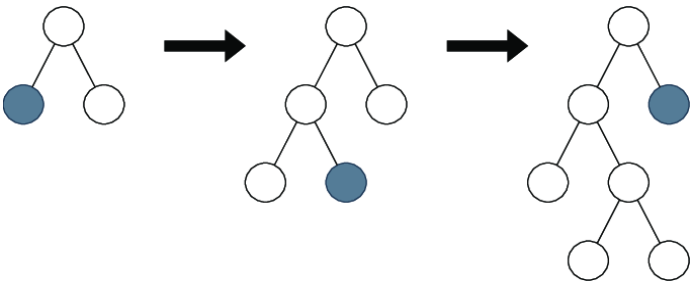


Fig. 3. Leaf wise tree growth

LightGBM (LGBM): LightGBM is a gradient boosting framework that uses tree-based learning algorithms.LightGBM grows tree by leaf-wise (best-first) as shown in figure 3 It will choose the leaf with max delta loss to grow. When growing same leaf, leaf-wise algorithm can reduce more loss than level-wise algorithm. It is designed to be distributed and efficient with the following advantages: faster training speed and higher efficiency, lower memory usage, better accuracy, support of parallel and GPU learning, capable of handling large-scale data. It is also designed to be easy to use, with a Python interface that is easy to use and efficient, and it has built-in cross-validation to tune parameters.

In the context of machine learning, a single classifier and an ensemble classifier are two different approaches to building a model for classification problems.A single classifier is a model that makes predictions based on a single algorithm or method. They work by learning a set of parameters that best fit the data and then using these parameters to make predictions. An ensemble classifier is a model that makes predictions based on the combined output of multiple models. These models are usually trained independently and then their predictions are combined in some way to make the final prediction. Ensemble classifiers are often more accurate than single classifiers because they can capture more complex patterns in the data. The proposed 'LGB-IDS' ensemble classifier is likely an ensemble of LGBM models, where each model makes a prediction and the final prediction is made based on the majority vote of the models.

IV. Evaluations & Discussion

In this project, we compared different machine learning classifiers.For each classifier, used certain metrics like accuracy score, speed, and error determination to evaluate these classifiers. We applied these classifiers to a mix of training and test data to get these metrics.. Then, we used these metrics to see how well the classifiers were making predictions [20].

A. CLASSIFICATION ACCURACY

In the present paper, accuracy presents the percentage of the classified normal and anomolous index. Accuracy can be calculated using the mathematical formula given in equation 2

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \times 100\% \quad (2)$$

The accuracy scores of LR, RF, XGB, and LGBM classifiers are shown in Table II

TABLE II
CLASSIFICATION ACCURACY

LR	RF	XGB	LGBM
99.47	99.48	99.44	99.5

B. ERROR RATE PERFOMANCE

Error rate performance measures how often a model predicts incorrectly. MAE measures the average absolute discrepancies between predicted and actual values, while MSE and its square root, RMSE, quantify the average squared errors, emphasizing the impact of larger deviations. In essence, these metrics collectively gauge the effectiveness of a model by assessing the closeness of its predictions to the actual outcomes, with the objective of minimizing these discrepancies for optimal performance. Comparison of Error rate is given in the table III

TABLE III
Error Rate

Classifiers	MAE	MSE	RMSE
LR	0.036	0.255	0.505
RF	0.0356	0.251	0.501
XGB	0.038	0.270	0.520
LGBM	0.0379	0.263	0.513

ROC Curve & AUC: The ROC curve is a graphical representation of the performance of a binary classification model. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The ROC curve is created by plotting the TPR against the FPR for a range of threshold values, from 0 to 1.

The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. The area under the ROC curve (AUC) provides an aggregate measure of performance across all possible classification thresholds. An AUC of 1 represents a perfect classifier, while an AUC of 0.5 represents a classifier that is no better than random guessing.Here the every classifiers ,LR,RF,XGB ,LGBM shows better ROC plots and the roc curve of proposed model is show in figure 4

C. CROSS VALIDATION

It's a resampling procedure that involves partitioning a dataset into complementary subsets, training a model on one subset (the training set), and validating the model on the other subset (the validation set or testing set).

The goal of cross-validation is to estimate how accurately a model will perform on unseen data. It helps to identify problems like overfitting or selection bias and provides insight on how the model will generalize to an independent dataset.

There are several types of cross-validation, but the most common is k-fold cross-validation. In this method, the data is divided into k groups or "folds".Here is a simple example of

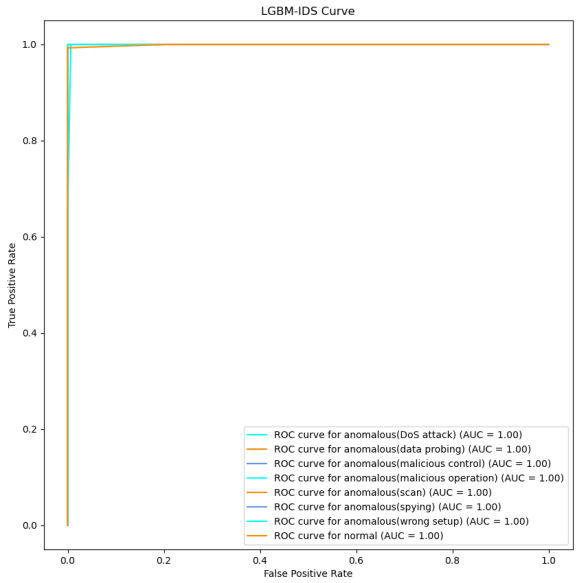


Fig. 4. ROC Curve of LightBGM

how k-fold cross-validation works:
-Split the data into k groups or folds.
-For each unique group:
Take the group as a holdout or test dataset.
Take the remaining groups as a training dataset.
Fit a model on the training set and evaluate it on the test set.
Retain the evaluation score and discard the model.
-The average of the evaluation scores from each iteration is the final estimate of the model's performance
Cross-validation is important because it allows for a more robust and reliable estimate of a model's performance. It helps to ensure that the model is not overfitting the training data and can generalize well to new, unseen data.The standard deviation(SD) is used to measure the variability of the cross-validation scores.A low standard deviation indicates that the values tend to be close to the mean, while a high standard deviation indicates that the values are spread out over a wider range(Refer Table IV)

TABLE IV
STANDARD DEVIATION

LR	RF	XGB	LGBM
0.0005	0.0005	0.0004	0.0003

D. TIME EFFICIENCY

Run time or speed of an algorithm refers to the amount of time it takes for the algorithm to execute or complete its task. Calculating runtime involves measuring the elapsed time from the start to the end of a program's execution.When evaluating runtime, it's essential to consider the input size or workload as runtime can vary based on the complexity of the task and the amount of data processed. Faster runtimes often indicate better performance and efficiency, especially in scenarios where quick execution is crucial, such as in real-time systems, large-scale data processing, or time-sensitive applications. Time in seconds for each classifier LR,RF,XGB and LGBM is given in table V

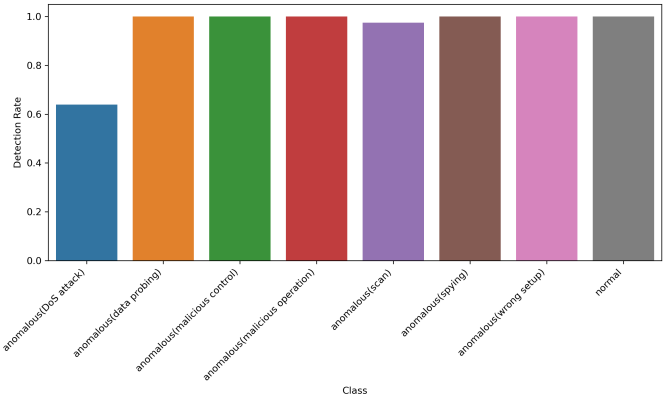
TABLE V
RUN TIME IN SECONDS

LR	RF	XGB	LGBM
10.4	13.89	17.66	1.5

E. PREDICTION AND DETECTION RATE

It the ability of a model to correctly identify positive instances or the true positive rate in a binary classification problem. It measures the proportion of actual positive instances that are correctly identified by the model. In other words, it calculates the ratio of correctly predicted positive samples to the total actual positive samples. LGBM classifier shows a better Prediction and dtection rate which shown in fugure 5

Fig. 5. Threat detection rate of LGBM



V. LGBM-IDS MODEL

Here the proposed model using ensembled Light Gradient Boosting Machine (LGBM) classification using majority voting. [21] [22]

Ensemble methods are a set of learning algorithms that can be used to solve different types of machine learning problems. The main idea behind ensemble methods is to combine the predictions of several models to get better results than any single model. This is achieved by training multiple models on different parts of the dataset and then combining their predictions.

In majority voting, the final prediction is the class that has been predicted most frequently by the individual models. This is a form of hard voting where each model makes a final decision (either class or value) and the most common decision is taken as the final prediction.

In this project, we used the LGBM model with majority voting, which is a form of boosting. Boosting is a technique where a sequence of weak learners are trained in a way that each subsequent model focuses on the mistakes made by the previous ones. This means that each model is trained on the residuals (errors) of the previous model. The final prediction is a weighted sum of the predictions of each model, where the weights are determined by the performance of the models.

The parameters of LGBM model are:

- boosting_type: This is set to 'gbt', which stands for Gradient Boosted Trees. This is the default boosting type for LGBM and it is a type of gradient boosting where the model is built by adding new trees to the ensemble that aim to correct the mistakes of the existing trees.
- max_depth: This is set to 2, which means that the maximum depth of the trees in the model is 2. This is a parameter that controls the complexity of the model. A smaller value will make the model simpler and less likely to overfit, but it might also miss important patterns in the data.
- n_estimators: This is set to 100, which means that the model consists of 100 trees. The number of trees in the model can affect the model's performance and computational cost. More trees can often lead to better performance, but they also increase the computational cost and the risk of overfitting.
- num_leaves: This is set to 3, which means that each tree in the model has 3 leaves. This is a parameter that controls the complexity of the individual trees in the model. A smaller value will make the trees simpler and less likely to overfit, but it might also miss important patterns in the data.
- learning_rate: This is set to 0.05, which means that the contribution of each tree to the final prediction is reduced by 5%. The learning rate is a parameter that controls the learning rate of the model. A smaller value will make the model learn more slowly and might result in a better fit to the data, but it might also increase the risk of overfitting.

Benefits of LGBM Ensemble Model

- Speed: LGBM is known for its high efficiency and speed. It uses a novel technique called Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value, which makes the algorithm run faster.
- Efficiency: LGBM uses a technique called Exclusive Feature Bundling (EFB) to group the categorical features into bins, which reduces the memory usage and speeds up the training process.
- High Accuracy: LGBM often achieves high accuracy on a wide range of datasets, which makes it a good choice for many machine learning tasks

Overall Perfomance of Proposed IDS-Model The proposed Intrusion Detection System (IDS)-Model demonstrated exceptional performance across various evaluation metrics, indicating its effectiveness in identifying and classifying intrusions. The model achieved an impressive accuracy rate of 99.5%, showcasing its ability to correctly classify instances. Additionally, with a high ROC score of 0.99, the model exhibited strong discriminatory power, effectively distinguishing between normal and intrusive activities.The runtime of the model was notably efficient, completing its analysis in approximately 1.5 seconds, highlighting its computational efficiency and suitability for real-time applications.

Confusion Matrix: The confusion matrix provides a comprehensive overview of the model's predictions, illustrating the counts of true positive, true negative, false positive, and false negative predictions. It helps in understanding the model's strengths and areas of improvement in classification tasks.

Precision, Recall, and F1 Score: Precision represents the proportion of correctly identified positive instances out of all instances classified as positive by the model. Recall, also known as sensitivity, denotes the proportion of actual positive instances correctly predicted by the model. The F1 score is a harmonic mean of precision and recall, providing a balanced assessment of a model's performance in binary classification tasks.

The detailed breakdown of these metrics is presented in the following table and figure 6 and 7

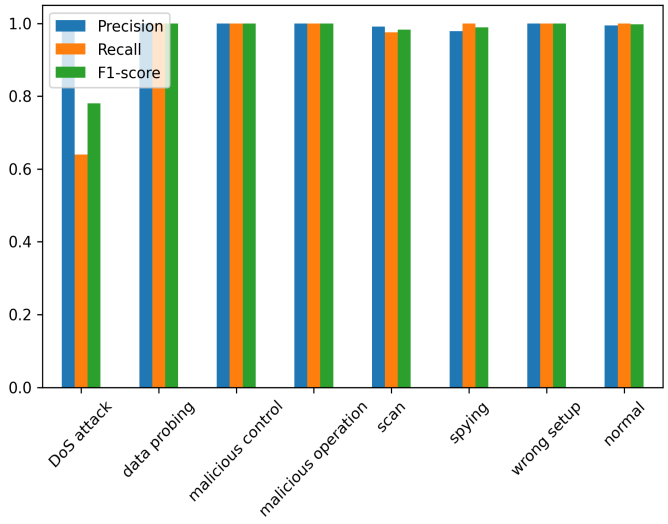
Fig. 6. Confusion matrix of LGBM-IDS

VI. Conclusion

This project built an effective Intrusion Detection System (IDS) for IoT-enabled smart homes using machine learning, specifically the Light Gradient Boosting Machine (LGBM). By combining LGBM with ensemble techniques like majority voting, it achieved 99.5% accuracy, a 0.99 ROC score, and a quick runtime of 1.5 seconds. These results show its high accuracy and speed, ideal for real-time security.

The study explored different machine learning classifiers, used dimensionality reduction to enhance efficiency, and tested the

Fig. 7. Classification of prediction rate in LGBM



model using the DS2OS dataset, showing promising performance in accuracy and efficiency.

This work contributes a robust model for IoT security, but there's room for improvement. Future research could focus on optimizing the model further and exploring different algorithms. Also, having more comprehensive databases for testing would be helpful.

In essence, this project demonstrates how machine learning, particularly the LGB-IDS model, greatly improves IoT security. While impressive, ongoing research is crucial to refine the model and tackle the specific challenges of IoT security.

REFERENCES

[1] Y. Shah and S. Sengupta, "A survey on classification of cyber-attacks on iot and iiot devices," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2020, pp. 0406–0413.

[2] Z. Shouran, A. Ashari, and T. Priyambodo, "Internet of things (iot) of smart home: privacy and security," *International Journal of Computer Applications*, vol. 182, no. 39, pp. 3–8, 2019.

[3] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[4] R. M. Aziz, M. F. Baluch, S. Patel, and A. H. Ganie, "Lgbm: a machine learning approach for ethereum fraud detection," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3321–3331, 2022.

[5] FrancoisXA, "Ds2os traffic traces," https://www.kaggle.com/francoisxa/ds2ostraffictraces, 2023, [Online; accessed 19-December-2023].

[6] C. Hazman, A. Guezzaz, S. Benkirane *et al.*, "Toward an intrusion detection model for iot-based smart environments," *Multimedia Tools and Applications*, 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-16436-0

[7] P. Kumar, G. P. Gupta, and R. Tripathi, "Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for iot networks," *Arabian Journal for Science and Engineering*, vol. 46, pp. 3749–3778, 2021.

[8] A. Huč, J. Šalej, and M. Trebar, "Analysis of machine learning algorithms for anomaly detection on edge devices," *Sensors*, vol. 21, no. 14, p. 4946, 2021.

[9] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Anomaly based network intrusion detection for iot attacks using deep learning technique," *Computers and Electrical Engineering*, vol. 107, p. 108626, 2023.

[10] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671–2701, 2019.

[11] E. Rehman, M. Haseeb-ud Din, A. J. Malik, T. K. Khan, A. A. Abbasi, S. Kadry, M. A. Khan, and S. Rho, "Intrusion detection based on machine learning in the internet of things, attacks and counter measures," *The Journal of Supercomputing*, pp. 1–35, 2022.

[12] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer networks*, vol. 174, p. 107247, 2020.

[13] O. D. Okey, S. S. Maidin, P. Adasme, R. Lopes Rosa, M. Saadi, D. Carrillo Melgarejo, and D. Zegarra Rodríguez, "Boostedenml: Efficient technique for detecting cyberattacks in iot systems using boosted ensemble machine learning," *Sensors*, vol. 22, no. 19, p. 7409, 2022.

[14] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for iot security using decision tree and gradient boosting," *The Journal of Supercomputing*, vol. 79, no. 3, pp. 3392–3411, 2023.

[15] Y. K. Saheed, "Performance improvement of intrusion detection system for detecting attacks on internet of things and edge of things," in *Artificial Intelligence for Cloud and Edge Computing*. Springer, 2022, pp. 321–339.

[16] R. G. Bace, P. Mell *et al.*, "Intrusion detection systems," 2001.

[17] M. S. Yadav and R. Kalpana, "Data preprocessing for intrusion detection system using encoding and normalization approaches," in *2019 11th International Conference on Advanced Computing (ICoAC)*. IEEE, 2019, pp. 265–269.

[18] C. Ordonez, "Data set preprocessing and transformation in a database system," *Intelligent Data Analysis*, vol. 15, no. 4, pp. 613–631, 2011.

[19] M. A. H. Dalfi, S. Chaabouni, and A. Fakhfakh, "Breast cancer detection using random forest supported by feature selection," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 2s, pp. 223–238, 2024.

[20] J. D. Novaković, A. Veljović, S. S. Ilić, Ž. Papić, and M. Tomović, "Evaluation of classification models in machine learning," *Theory and Applications of Mathematics & Computer Science*, vol. 7, no. 1, p. 39, 2017.

[21] D. Rani, N. S. Gill, P. Gulia, F. Arena, and G. Pau, "Design of an intrusion detection model for iot-enabled smart home," *IEEE Access*, vol. 11, pp. 52 509–52 526, 2023.

[22] P. Kumar, G. P. Gupta, and R. Tripathi, "A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks," *Journal of ambient intelligence and humanized Computing*, vol. 12, pp. 9555–9572, 2021.