Article

Not peer-reviewed version

Detecting Anomalies in IoT Devices: A Machine Learning-Based Solution

Rama Al Attar , mouhammd alkasassbeh * , Mu'awya Al-Dala'ien , Manar Alohaly

Posted Date: 8 April 2024

doi: 10.20944/preprints202404.0499.v1

Keywords: IoT; machine learning; security; ensemble; IDS; anomaly detection

Article

Detecting Anomalies in IoT Devices: A Machine Learning-Based Solution

Rama Al Attar [1], Mouhammd Alkasassbeh [1,*], Mu'awya Al-Dala'ien [1] and Manar Alohaly [2]

[1] Princess Sumaya University for Technology, Amman, Jordan; rama.m18@hotmail.com (R.A.); m.aldalaien@psut.edu.jo. (M. AD)
[2] Department of Computer Sciences, College of Computer and Information Sciences Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; mfalohaly@pnu.edu.sa (M.ALO)
* Correspondence: m.alkasassbeh@psut.edu.jo

Abstract: The growing shift toward Internet of Things (IoT)-based solutions expands the attack surface of systems by connecting an extensive network of heterogeneous devices and technologies. The heterogeneity of IoT devices and the scale of the network often make conventional security measures impractical. Therefore, recent research efforts have focused on machine learning (ML)-based device-agnostic IoT security solutions. However, most proposed solutions have focused solely on detecting malicious traffic in IoT networks. While this is important, further information about the attack is needed to provide a comprehensive defense before, during, and after the incident. To address this gap, we proposed an anomaly-based intrusion detection model that detects malicious traffic and identifies the attack category and subcategory with high accuracy, recall, precision, and F1-score. The proposed intrusion detection model is an ensemble model that integrates multiple ML models to produce more robust and reliable detections. For evaluation, we performed three sets of experiments: (1) a binary classification to detect malicious traffic; (2) a multi-class classification to detect the attack category; and (3) a multi-class classification to detect the attack subcategory. In the experiments, our model achieved an accuracy, recall, precision, and F1-score of 100% for the binary detection; a precision, recall, and F1-score of 99% and an accuracy of 100% for the multi-class category classification; and an accuracy, recall, precision, and F1-score of 88% for the multi-class subcategory classification. These results suggest that the proposed model can reliably detect anomalies in IoT devices.

Keywords: IoT; machine learning; security; ensemble; IDS; anomaly detection

1. Introduction

The Internet of Things (IoT) emerged in 1999 with the introduction of Radio Frequency Identification (RFID) tags [1]. Fast forward twenty-four years and the IoT landscape has grown exponentially, becoming increasingly in demand. According to Cambridge consultants, IoT connections across various sectors are projected to reach 155.7 million between 2016 and 2024 [2]. As IoT devices become more integrated into our daily lives, the attack surface for these devices expands. In fact, Kaspersky reported a staggering 140% increase in IoT breaches from January to June 2021 [3]. Researchers have been working tirelessly to address IoT vulnerabilities and enhance device security. However, due to IoT devices' limited processing power and memory, traditional security methods fall short in the IoT environment. Furthermore, as manufacturers prioritize rapid market entry over comprehensive security measures, security concerns in IoT applications are often overlooked. Unlike conventional computing methods, soft computing solutions require less computational power. Therefore, recent studies tend to adopt soft computing solutions that better suit resource-scarce IoT devices. As the number of interconnected devices rises, safeguarding them from potential attacks becomes increasingly challenging. This situation stresses the importance of scalable, efficient anomaly-based intrusion detection models driven by machine learning algorithms. Abomhara and

 doi:10.20944/preprints202404.0499.v1

Køien [4] identified various IoT security challenges, such as user privacy, identity management, data sharing and protection, authentication, and authorization. Intrusion detection systems (IDSs) are crucial to IoT security as they monitor networks for suspicious activities [5]. Intrusion detection models can be divided into two categories: signature-based and anomaly-based models. While signature-based IDSs offer high precision by matching network traffic to predefined known attacks, they struggle to detect less frequent or novel attacks, like zero-day threats. In contrast, anomaly-based IDSs improve recall by learning unusual behaviors and patterns but suffer from high false detection rates, which remains an ongoing research challenge.

a. Motivations and contributions

Several researchers have proposed anomaly detection techniques, most of which are ML-based. There are also several studies on combining different deep-learning methods for anomaly detection [6–8]. But still, three important research problems persist: combining ML algorithms like Decision Trees and Random Forests to harness their collective power, classifying the sub-categories of anomalous traffic using optimum datasets, and making predictions more efficient. Thus, an ensemble ML-based intrusion detection model is a promising solution that is independent of potential negligence from manufacturers and end users. Moreover, several existing studies were based on datasets that are not sufficiently representative of real-world IoT security issues. In this study, instead, we performed our experiments on the IoTID20 dataset, which resembles real-world security issues arising in IoT environments, to improve the practicality of the proposed solution. The main contributions of this research are summarized as follows:

- We developed an ensemble ML-based detection method against IoT attacks using the IoTID20 dataset. The proposed model makes three-faceted predictions: binary classification for anomalous or normal traffic, multi-class prediction for broad categorization of the attack, and classifying attacks within the sub-category. This layered approach provides a more nuanced understanding of the threat landscape, which can significantly aid in effective threat mitigation and response. To the best of our knowledge, this is the first work that applies an ensemble machine learning approach specifically to the IoTID20 dataset. This dataset represents a specific and complex IoT environment with varied network attacks, which poses unique challenges that our model effectively addresses.
- We provided a proof of concept implementation of the proposed ensemble ML-based solution, which consists of a diverse mix of individual classifiers (Decision Trees, Extra Trees Classifier, K-Nearest Neighbors, and Random Forests), creating a more robust and reliable system for anomaly detection. This particular combination is a novel contribution that enhances the performance and stability of the detection system.

The paper is structured as follows: Section 2 reviews relevant work on detecting IoT attacks using the IoTID20 dataset. Section 3 explains the methodology behind constructing the IDS model. Section 4 presents the results achieved by the proposed IDS model, and Section 5 concludes the paper.

2. Related Work

In our earlier work, we conducted an extensive review of recently published studies in IoT intrusion detection [6]. The published survey provided a detailed analysis of widely used IoT datasets. The results of this analysis motivated our decision to use the IoTID20 dataset. In addition, the survey compared the performance of existing ML-based solutions built using the IoTID20 dataset. In this section, we extend the findings of our earlier study with more recent related work utilizing the IoTID20 dataset and different approaches followed by researchers to mitigate IoT abuse.

a. **Mitigate IoT Abuse**

To overcome the shortcomings of anomaly detection methodologies that depended solely on cloud computing or single-source time series data, the authors [9] presented a distributed edge computing model for anomaly detection in IoT-based industrial sustainability. An Edge Computing based Anomaly Detection Algorithm (ECADA) was introduced to detect abnormalities in single-source and multi-source time series data. Experimental results showed that the algorithm was

 doi:10.20944/preprints202404.0499.v1

superior to its competitors when presented with novel anomalies. This research demonstrates the promise of edge computing in solving the transmission and processing difficulties of anomaly detection in real time.

CorrACC is a novel hybrid feature selection approach proposed by [10] to improve the accuracy of machine learning algorithms for identifying anomalous and malicious traffic in Smart Internet of Things (SIoT) networks. The authors emphasize the significance of feature selection for accurate anomaly and intrusion detection in IoT networks. Correlation Attribute Evaluation (CAE) and specific machine learning classifier accuracy (ACC) are used to select effective feature sets for a particular classifier which is used in their proposed method. The authors evaluate their approach on the Bot-IoT dataset and demonstrate that, relative to other feature selection techniques, their proposed approach obtains accuracy rates greater than 95%.

Another relevant study in distributed computing and mobile communication security presents DeepAutoD (D-AD), an innovative unpacking framework for secure feature extraction in distributed machine learning-based malicious code detection [11]. The authors address the challenge of malicious apps that employ advanced techniques to conceal high-risk code, which can hinder efficient mobile communication. By integrating deep deception call chains, D-AD effectively identifies mainstream Apps in the App-market and offers customizable algorithms for advanced Android systems. This work demonstrates that the proposed framework outperforms existing solutions regarding safety and effectiveness, making it a noteworthy contribution to the domain of Android security.

The authors [12] propose a novel method for identifying malicious traffic flows in IoT networks using machine learning algorithms. They demonstrate the efficacy of their approach by applying four distinct machine-learning algorithms to the Bot-IoT dataset. The experimental results indicate that the proposed method obtains an average accuracy of greater than 96%, which is superior to other feature selection techniques. The authors emphasize the importance of accurate feature selection for accurate malicious traffic detection in IoT networks and propose a new feature selection metric called CorrAUC, which eliminates inaccurate features and selects effective ones based on the area under the curve (AUC) metric. CorrAUC, a new feature selection algorithm based on the wrapper technique, accurately filters the features and selects the most effective ones for the chosen machine learning algorithm. Using an integrated TOPSIS and Shannon entropy based on a bijective soft set, the authors validate the desired features for identifying malicious traffic. This work significantly improves the security of IoT networks by providing an efficient and accurate method for detecting malicious traffic flows.

b. Recent Related Work Utilizing IoTID20 Dataset

Authors in [13] proposed a MidSoit IDS, which consists of three stages; the first stage identifies connected IoT devices, stage two identifies whether the traffic is benign or malicious, and stage three identifies the attack type of the malicious traffic identified in stage 2. Finally, the results of the third stage are sent to an action manager component that triggers necessary actions. The proposed model was evaluated on IoTID20, CIC-IDS-2017, and BOT-IoT. The authors then tested the model on nine machine learning classifiers, including LSVM, QSVM, K-Nearest Neighbor (KNN), Linear discriminant analysis (LDA), QDA, multilayer perceptron, LSTM, autoencoder, and Decision Tree (DT) classifier. The proposed model achieved class category accuracy of 99.15% using the IoTID20 dataset. Furthermore, the analysis was performed without sampling, under-sampling, and Synthetic Minority Over-Sampling Technique (SMOTE). It was evident that the model performed better without sampling.

Authors in [7] proposed a DCNN model consisting of 8 layers. The first layer is a 1D convolution layer followed by a max pooling layer, a 1D convolution layer, a max pooling layer, a flattened layer, and finally, three DNN layers. Before evaluating the model, the authors performed data preprocessing and the ETC feature selection method resulting in 62 selected features that achieved information gain greater than 0.001. The authors tested five optimizers, where Nadam was the top-performing optimizer. The proposed model achieved an accuracy of 99.84% and an F1-score of 99.34% for binary classification. As for class category classification, the model achieved an accuracy

 doi:10.20944/preprints202404.0499.v1

of 98.12% and an F1-score of 97.46%. Finally, for the multiclass subcategory classification, the model achieved an accuracy of 77.5% and an F1-score of 76%.

Table 1 provides a summary of the related studies on the IoTID20 dataset. Most of these studies have modeled the IDS problem as a binary classification problem. However, while the binary classification model is essential to distinguish suspicious from normal traffic, further information is needed to provide a comprehensive defense before, during, and after attacks. Therefore, this work focused on identifying the category and subcategory instances in the IoTID20 dataset.

Table 1. Summary of literature Accuracy using IoTID20 dataset.

Paper	**Binary**	**Multi – Class (Category)**	**Multi-Class (Subcategory)**
[14]	Training and Testing Accuracy, IoTID20 = 100%	X	X
[15]	Accuracy, IoTID20 = 100%	X	X
[16]	Accuracy, IotID20 = 94.7%	X	X
[13]	X	Accuracy DT = 99.15%	X
[17]	X	X	Accuracy DT = 88% Ensemble = 87% Random Forest = 84% Gaussian NB = 73% LDA = 70% Logistic Regression = 40% SVM = 40%
[18]	Accuracy, DoS = 99.95% MITM = 99.9761%, Scan = 99.96% Mirai Not covered in the paper	X	X
[19]	Accuracy: CNN = 96%, CNN-LSTM = 98% LSTM = 98.2%	X	X
[20]	X	Accuracy Proposed Model = 86.48%	X
[21]	X	X	Accuracy Proposed Model = 55.79%
[8]	Detection Rate, DT = 0.99, KNN = 0.98 RF = 0.98	X	X
[22]	X	X	Detection Rate

 doi:10.20944/preprints202404.0499.v1

			(Average) ANN = 96% DT = 94% SVM = 91.5% LR = 91.3%
[23]	Sequential Backward Processing *Highest Accuracy*: XGBoost: 99.31% Sequential Forward Processing *Highest Accuracy*: XGBoost: 99.3% Recursive Feature Elimination *Highest Accuracy*: XGBoost: 98.79%	X	X
[7]	Accuracy Proposed Model = 99.84%	Accuracy Proposed Model = 98.12%	Accuracy Proposed Model = 77.5%

As the IoTID20 dataset is relatively new and closest to a real-world IoT environment, there is more research potential on the dataset, specifically implementing Ensemble machine learning to IoT environments. Three out of the outlined literature review have applied ensemble machine learning, out of which two were applied to binary classification, and the third was applied to multi-class classification achieving an accuracy of 87%. This paper will focus on building an Ensemble-based IDS model that can detect attacks targeting IoT devices using the dataset IoTID20.

3. Methodology

This section presents the proposed approach to detecting IoT-based attacks along with the selected features and implementation requirements of the intrusion detection model.

a. IoTID20 Testbed Environment

To safeguard against security risks that face IoT devices, it is essential to understand the underlying architecture. The IoT architecture is divided preliminarily into three layers [5]. The perception layer is the physical layer which consists of edge devices such as sensors, CCTV cameras, and actuators. This layer is responsible for gathering information from its surroundings, making it a sensitive layer as it is relatively exposed. The network layer is responsible for connecting IoT devices to network devices and servers by acting as a median to transfer data from one point to another via various network protocols. The application layer is responsible for providing professional services to the end user. In this paper, a recent IoT dataset was chosen, which includes possible attacks on all three layers of the IoT environment. The studied IoT dataset includes four attack types as follows:

- Denial of Service Attack: an attack that tries to compromise the availability of its target by making it unresponsive.
- Mirai Attack: an evolved type of DoS attack named DDoS. DDoS attacks use the same method as DoS. However, this time a group of attackers are attacking the victim simultaneously.
- Man in the Middle Attack: an attack that tries to eavesdrop on communications taking place between two nodes, where the attacker starts acting as a proxy.

 doi:10.20944/preprints202404.0499.v1

- Probing Attack: an attack that scans the network looking for vulnerabilities that could be exploited to help perform an attack on the device.
- Probing Attack: an attack that scans the network looking for vulnerabilities that could be exploited to help perform an attack on the device.

The IoTID20 dataset was created using Pcap files from the IoT Network Intrusion dataset [6,17] to detect malicious activities in IoT networks. The testbed resembles an environment similar to a smart home with a number of interconnected IoT devices. The IoTID20 testbed consists of an AI speaker (SKT NGU), security camera (EZVIZ Wi-Fi Camera), smartphones and laptops connected to an access point. The dataset consists of 625,783 records and 86 features, of which 83 are network features and 3 are label features. Table 2 outlines the three label features and the number of records for each class.

Table 2. IoTID20 Dataset Distribution.

Binary	**Multi-Class (Category)**	**Multi-Class (Subcategory)**
Normal (40,073)	Normal (40,073)	Normal (40,073)
Anomaly (585,710)	DoS (59,391)	Syn-flooding(59,391)
	MITM (35,377)	ARP Spoofing (35,377)
	Mirai (415,677)	Brute Force (121,181)
		HTTP Flooding (55,818)
		UDP Flooding (183,554)
		ACK Flooding (55,124)
	Scan (75,265)	Host Port (22,192)
		Port OS (53,073)

Figure 1 outlines the general approach of the proposed IoT intrusion detection model. To build the model, we used a Windows 10 pro computer machine. The technical specifications of the CPU are (Ryzen 5 5600x, 6 cores 12 threads) and (16GB 3600 MHZ) RAM. The model was built on a Visual Studio Code environment. Python was used as a programming language utilizing Scikit-learn, Numpy, and Pandas libraries.

Data preprocessing was the first stage, followed by extracting the most relevant network features. The quality of data has a direct impact on the machine learning models' performance. The detailed description of this stage is as follows:

- Converted all data types to integer values to ensure data is understood by the machine learning model.
- Removed any infinity values found and replaced them with NaN values instead.
- Replaced NaN values utilizing the mean strategy ‘simple imputer’ imputation transformer from sklearn, which replaces NaN values by the mean column value.
- Performed label encoding on three columns (Binary, Multi-Class Category, and Multi-Class Subcategory) which transforms categorical columns into numerical values.
- Dropped the remaining categorical features to reduce margin of error (Flow_ID, Dst_IP, Src_IP, Timestamp)
- Dropped constant features as per the correlation matrix of the IoTID20 dataset; they contain only one value for all the outputs in the dataset and provide no information.

 doi:10.20944/preprints202404.0499.v1

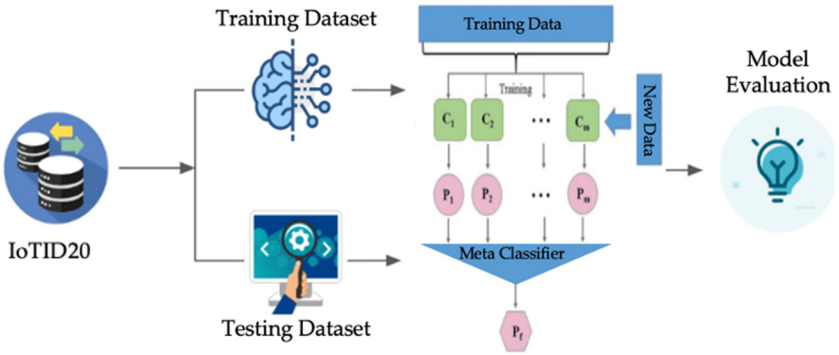

Figure 1. The proposed IDS model.

b. **Overall IDS Model Design Phase**

Phase Two of the proposed approach includes testing the impact of two sampling techniques that could potentially enhance the machine learning model by solving the challenge of the imbalance found in the IoTID20 dataset. A random under-sampling technique [24] was utilized where the majority class is reduced to the minority class by randomly deleting instances to result in a balanced dataset. In addition, the SMOTE oversampling technique [25] was utilized to produce new synthetic data samples. The output of this phase will result in a balanced dataset.

Phase three of the suggested model studies the effect of dimensionality reduction techniques that reduce the number of input variables in a dataset where important features are chosen to help reduce computational time, cost of modeling, save storage space, improve the detection model, and avoid the overfitting of the ML model. Principle component analysis [26] was utilized as a dimensionality reduction method where principal components with a cumulative variance > 99% were chosen.

Phase four of the suggested model approach splits the dataset into training and testing datasets using stratified K-fold cross-validation with a K value of 10. Stratified cross-validation enhances K-fold cross-validation by preserving the class distribution in the training and testing datasets [27]. The training dataset is then fed into the machine learning model to train the model with possible attacks, while the test dataset is used to test and evaluate the machine learning model.

c. **Ensemble Machine Learning Model Selection Phase**

Ensemble machine learning combines multiple base learners to train the dataset, creating a model with improved stability and performance. [28] outlined three reasons why ensemble learning outperforms single classifiers; (1) the training data may provide insufficient information to choose a single best classifier. The ensemble algorithm solves this by averaging their votes, which reduces the likelihood of selecting the incorrect classifier. (2) the learning algorithm's search process may be imperfect; ensemble learning solves this as the search would be originating from different points, which could lead to better accuracy. And finally, (3) the hypothesis space being searched may not express a true function. However, applying various weights of the hypothesis makes it possible to enlarge the space of the expressible function. As previously outlined, the IoTID20 dataset is an imbalanced dataset. Consequently, ensemble methods are more stable than single models and less affected by dataset class imbalance [29].

During the development of the ensemble machine learning model, multiple classifiers were tested to achieve optimum results. Of the eleven classifiers, Decision Tree, Random Forest, KNN, and Extra Tree achieved better results and are used to build an advanced ensemble ML classifier. Advanced ensemble methods are divided into three types: (1) Bagging, (2) Boosting, and (3) Stacking. Figure 2 displays the results of the best-performing ML ensemble classifier for the binary, multi-class category, and multi-class subcategory classification.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 April 2024 doi:10.20944/preprints202404.0499.v1

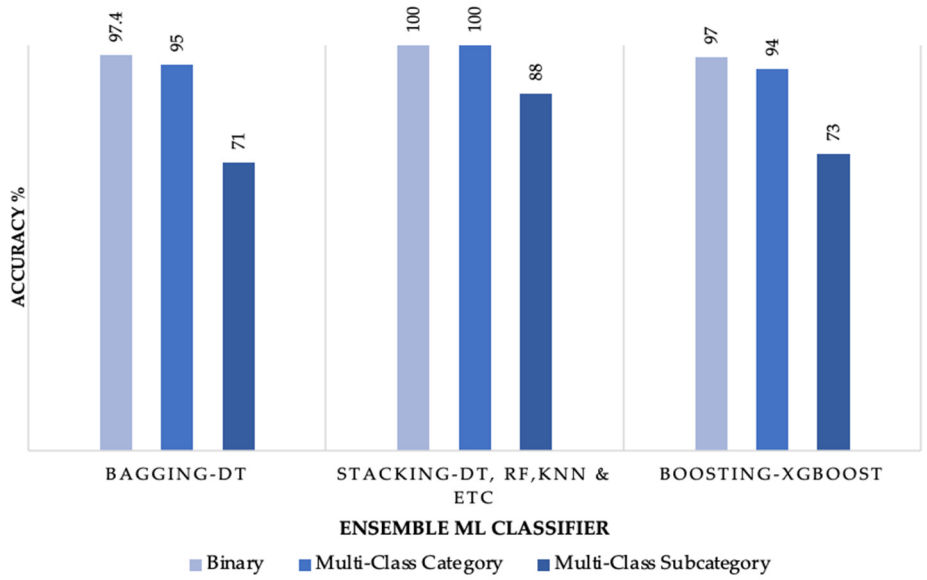

Figure 2. Comparison between advanced ensemble models.

As depicted in Figure 2, the stacking ensemble model performed better than bagging and boosting. Stacking has the advantage of utilizing multiple ML classifiers as base learners, followed by a meta-classifier. The meta-classifier combines the different predictions made by the base learners and improves them, yielding better results. Figure 3 outlines the stacking ensemble model utilized throughout this paper.

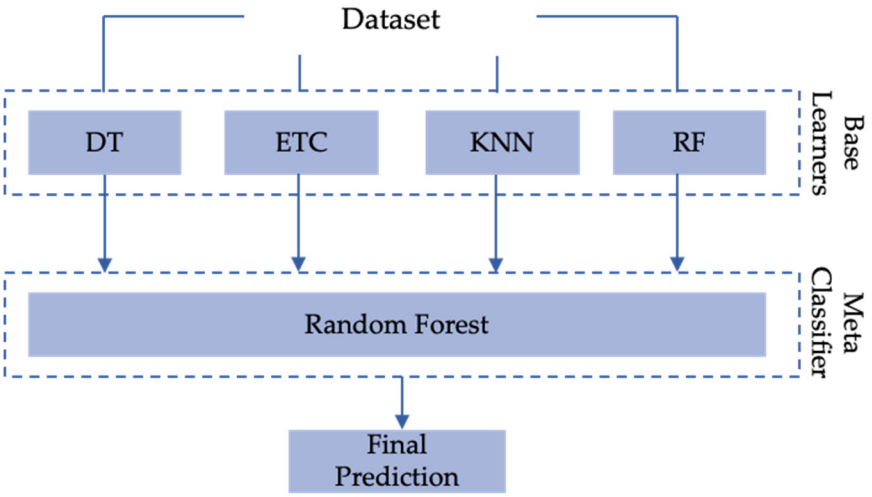

Figure 3. Proposed Stacking Ensemble IDS Model.

4. Experiments and Results

The experiments and findings obtained for binary, category, and subcategory classifications are described in this section. In addition, the impact of sampling and dimensionality reduction on the machine learning model was tested in each experiment. Accuracy is a widely used metric to assess the performance of machine learning models and will be used to compare the results of this study with the previously outlined literature. However, accuracy is a misleading metric when dealing with imbalanced datasets as it would be predicting based on the majority class, yielding high accuracies [24]. Instead, in imbalanced datasets F1-Score is the metric to rely on.

a. **Binary Classification**

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 April 2024 doi:10.20944/preprints202404.0499.v1

In binary classification, the IDS model predicts whether the traffic is malicious or benign. Two experiments were performed in this stage. The first experiment was to test the effect of the dataset imbalance on the IDS model, followed by testing the effect of dimensionality reduction utilizing PCA. For dataset imbalance, a comparison was conducted between a balanced model utilizing random under-sampling and an imbalanced model. We found that the IDS model achieved an accuracy of 100% in both instances; however, a higher F1-score and recall in the imbalanced model of 100% compared with 97.1% and 98.1% in the balanced model. As the IDS model achieved better results while keeping the dataset imbalanced, further analysis has been performed by applying dimensionality reduction, phase three of the proposed model. After applying PCA dimensionality reduction, we noticed that the model performed better by utilizing all features of the IoTID20 dataset, as the F1-Score decreased from 100% to 93.70%. Likewise, the recall has also decreased from 100% to 92.40%. Hence, the IDS model for binary classification was built by utilizing all features of the IoTID20 dataset.

Table 3 outlines the detailed results of the stacking IDS model for binary classification per class type, and Figure 4 outlines the confusion matrix for binary classification.

Table 3. Binary Classification Results.

Class	Recall	Precision	F1-score
Normal	100%	99%	99.2%
Attack	100%	100%	100%

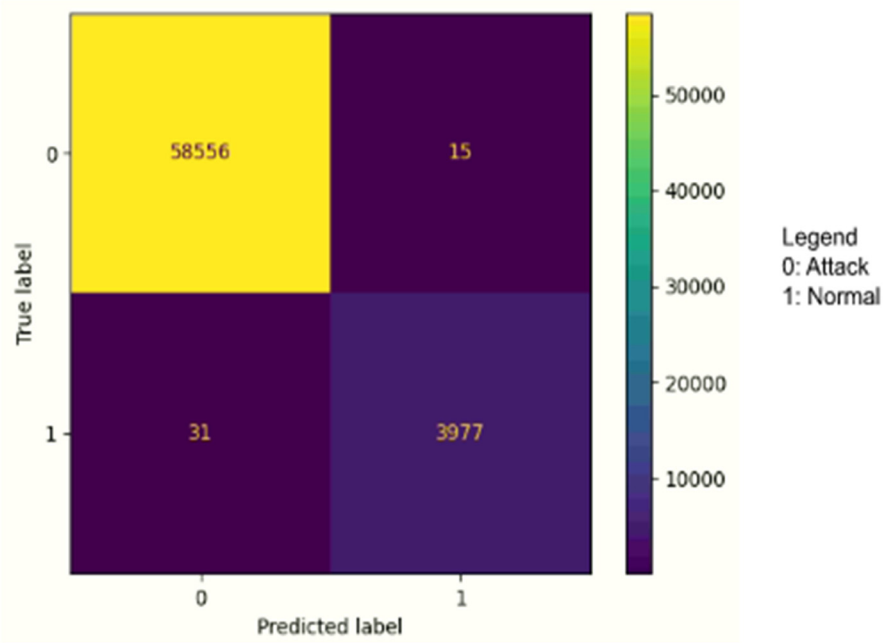

Figure 4. Confusion Matrix for Binary Classification.

In binary classification, the Ensemble IDS model achieved a recall of 100%, which means it can correctly identify benign and attack instances. Furthermore, the model achieved a precision of 99% of positive predictions in benign instances and 100% of positive predictions in attack instances. As the dataset is imbalanced F1-score is used as a measure to calculate the accuracy of the model, where it achieved 99%.

b. **Multi-Class Classification (Categories)**

In multi-class category classification, the IDS model predicts whether the traffic is benign, DoS, Mirai, MITM or a Scan attack. Similar to binary classification two experiments were performed in this stage, the first experiment was to test the effect of the dataset imbalance on the IDS model. Followed by testing the effect of dimensionality reduction utilizing PCA. To test the impact of the imbalanced

 doi:10.20944/preprints202404.0499.v1

dataset on the IDS model, both under sampling and SMOTE oversampling techniques were utilized to achieve a balanced dataset and a comparison was made with a model built utilizing an imbalanced dataset. Table 4 outlines the six different methods tested for multi-class category classification. In each method the model was under sampled and/or oversampled to reach a value of one of the class types. The IDS model achieved good results on all methods where the difference between them was nearly non-existent. However, after PCA dimensionality reduction was applied it was noticed that the IDS model performed better utilizing all features of the IoTID20 dataset. The accuracy of the balanced datasets de-creased by 17-20%, while the F1-score and recall decreased by 32-37%.

Table 4. Multi-Class Category Sampling Distribution.

Target Class Type	Sampling Method
MITM	All instances were under sampled to 35,377
Normal	Instances < 40,073 were oversampled using SMOTE Instances > 40,073 were under sampled
DoS	Instances < 59,391 were oversampled using SMOTE Instances > 59,391 were under sampled
Scan	Instances < 75,265 were oversampled using SMOTE Instances > 75,265 were under-sampled
Mirai	All instances were oversampled to 415,677 using SMOTE
No Sampling	All instances are kept as their original distribution within the IoTID20 dataset

Table 5 outlines the detailed results of the stacking IDS model per class type, and Figure 5 that outlines the confusion matrix for category classification.

Table 5. Multi-Class Category Classification Results.

Class	Recall	Precision	F1-score
Normal	99%	98%	99%
DoS	100%	100%	100%
MITM	99%	90%	93%
Mirai	99%	100%	99%
Scan	99%	98%	99%

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 April 2024 doi:10.20944/preprints202404.0499.v1

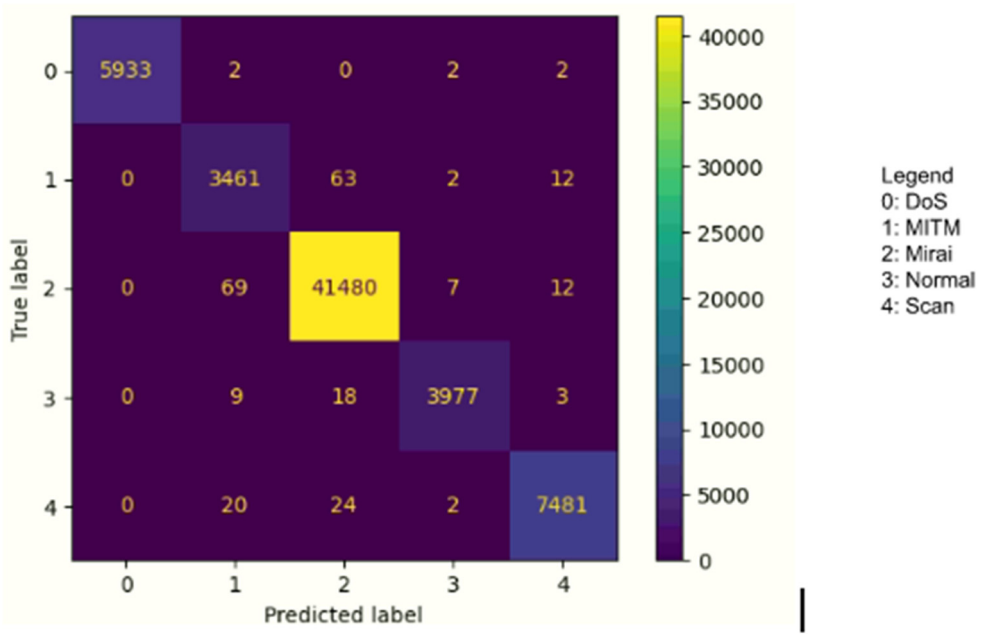

Figure 5. Confusion Matrix for Category Classification.

In multi-class category classification, the Ensemble IDS model achieved a recall of 99% for the normal class, which means the model was able to identify 99% of benign instances and was able to detect attack instances where it achieved 100% for DoS, followed by 99% for MITM, Mirai, and Scan attack instances. Furthermore, the model achieved an overall precision of 97.2% of positive predictions. As the dataset is imbalanced F1-score is used as a measure to calculate the accuracy of the model, where it achieved an overall of 98%.

c. **Multi-Class Classification (Sub-Categories)**

In multi-class subcategory classification, the IDS model predicts whether the traffic is benign, DoS, MITM, Brute Force, HTTP Flooding, UDP Flooding, ACK Flooding, Host Port or Port OS attacks. Similar to both binary and category classification two experiments were performed in this stage, the first experiment was to test the effect of the dataset imbalance on the IDS model. Followed by testing the effect of dimensionality reduction utilizing PCA. To test the impact of the imbalanced dataset on the IDS model, both under sampling and SMOTE oversampling techniques were utilized to achieve a balanced dataset and a comparison was made with a model built utilizing an imbalanced dataset. Table 6 outlines the ten different methods tested for multi-class subcategory classification. In each method the model was under sampled and/or oversampled to reach a value of one of the class types. The IDS model performed optimum results without sampling; however, the results were very close between the various methods tested. On the other hand, after applying PCA dimensionality reduction the IDS model performed better utilizing all features of the IoTID20 dataset and out of all class classifications, subcategory class was mostly affected by PCA where the accuracy decreased from 88% to 55%.

Table 6. Multi-Class Subcategory Sampling Distribution.

Target Class Type	Sampling Method
Scan - Host Port	All instances were under sampled to 22,192

 doi:10.20944/preprints202404.0499.v1

MITM	Instances < 35,377 were oversampled using SMOTE Instances > 35,377 were under-sampled
Normal	Instances < 40,073 were oversampled using SMOTE Instances > 40,073 were under-sampled
Scan - Port OS	Instances < 53,073 were oversampled using SMOTE Instances > 53,073 were under-sampled
Mirai - ACK Flooding	Instances < 55,124 were oversampled using SMOTE Instances > 55,124 were under-sampled
Mirai - HTTP Flooding	Instances < 55,818 were oversampled using SMOTE Instances > 55,818 were under-sampled
DoS	Instances < 59,391 were oversampled using SMOTE Instances > 59,391 were under-sampled
Mirai - Brute Force	Instances < 121,181 were oversampled using SMOTE Instances > 121,181 were under-sampled
Mirai - UDP Flooding	All instances were oversampled to 183,554 using SMOTE

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 April 2024 doi:10.20944/preprints202404.0499.v1

No Sampling	All instances are kept as their original distribution within the IoTID20 dataset

Analyzing the previous results, it is concluded that a stacking IDS model with no sampling and feature selection yields better results. To be able to secure IoT devices it is vital that the model achieves optimum results. Table 7 outlines the detailed results of the stacking IDS model built for subcategory classification. In addition, Figure 6 that outlines the confusion matrix for subcategory classification.

Table 7. Multi-Class Subcategory Classification Results.

Class	Recall	Precision	F1-score
Normal	**99%**	**100%**	**99%**
DoS	**100%**	**100%**	**100%**
MITM	97%	97%	97%
Mirai - Ack Flooding	65%	67%	66%
Mirai - Brute Force	98%	93%	96%
Mirai - HTTP Flooding	65%	67%	66%
Mirai - UDP Flooding	89%	91%	90%
Scan - Host Port	62%	80%	70%
Scan - Port OS	93%	86%	89%

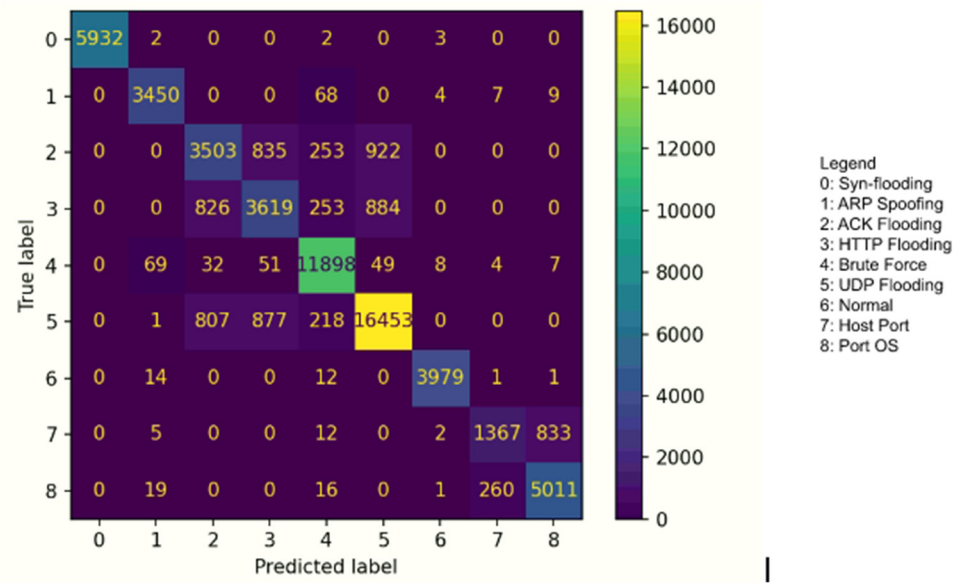

Figure 6. Confusion Matrix for Subcategory Classification.

d. **Ensemble Techniques**

The built stacking model achieved results better than other ensemble methods as previously outlined in Figure 2. Furthermore, the proposed stacking model was compared with individual ML classifiers

 doi:10.20944/preprints202404.0499.v1

where stacking achieved better results than all individual classifiers especially in multiclass subcategory classification. Figures 7–9 outline the comparison between the proposed stacking model and individual ML classifiers where all models were tested on no sampling methods for a fair comparison.

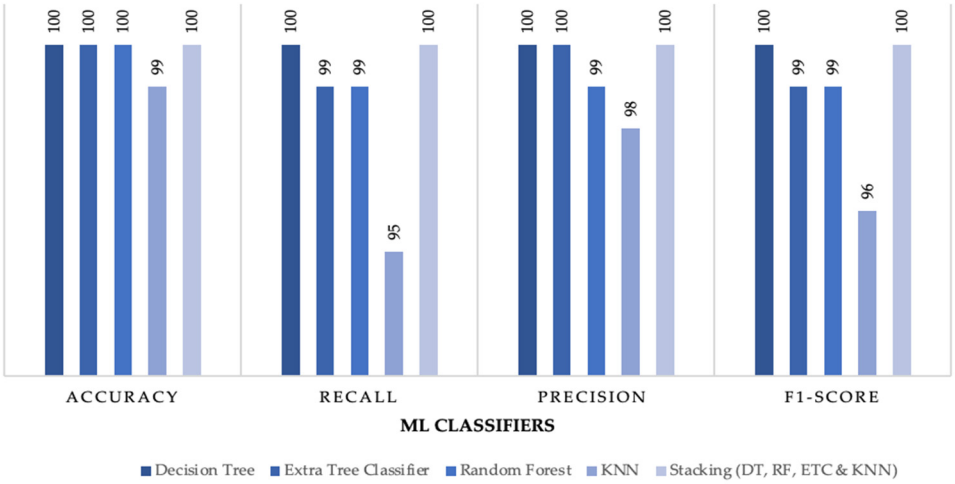

Figure 7. Binary Classification Base Classifiers and Stacking Model Comparison.

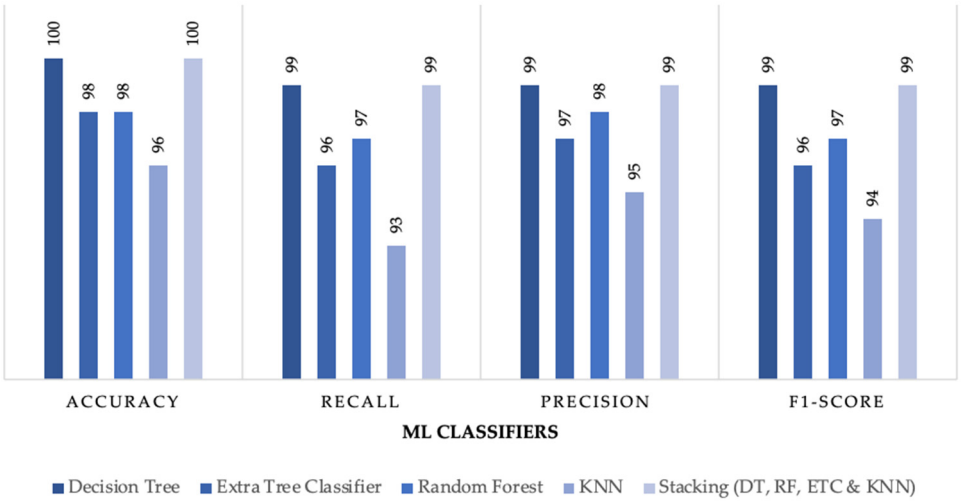

Figure 8. Category Classification Base Classifiers and Stacking Model Comparison.

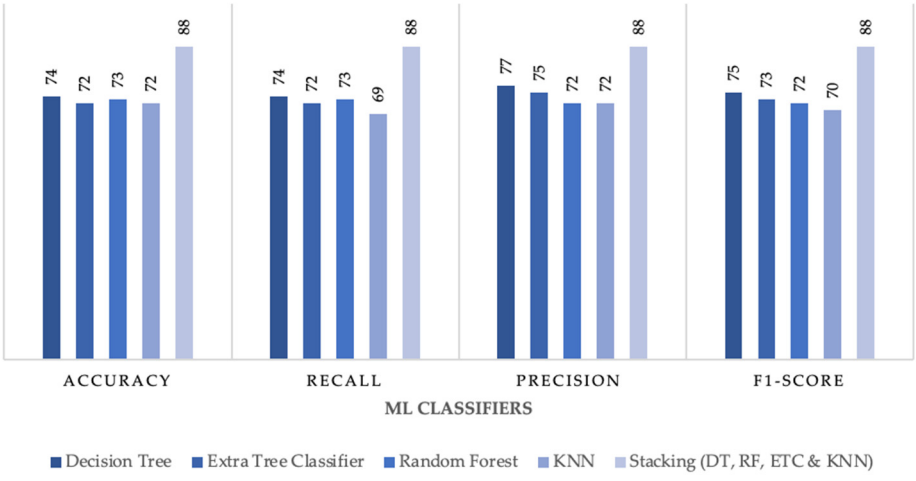

Figure 9. Subcategory Classification Base Classifiers and Stacking Model Comparison.

 doi:10.20944/preprints202404.0499.v1

e. **Comparison with Related Previous Work**

This section outlines the comparison between the built stacking IDS model with the previously outlined literature as follows:

1. Binary Classification: Out of the previously outlined literature, nine papers evaluated their models against binary classification. Figure 10 summarizes the results of the literature in comparison with the proposed IDS model. The proposed stacking IDS model performed better than some of the literature, and in some instances, the performance was similar with 100% accuracy.

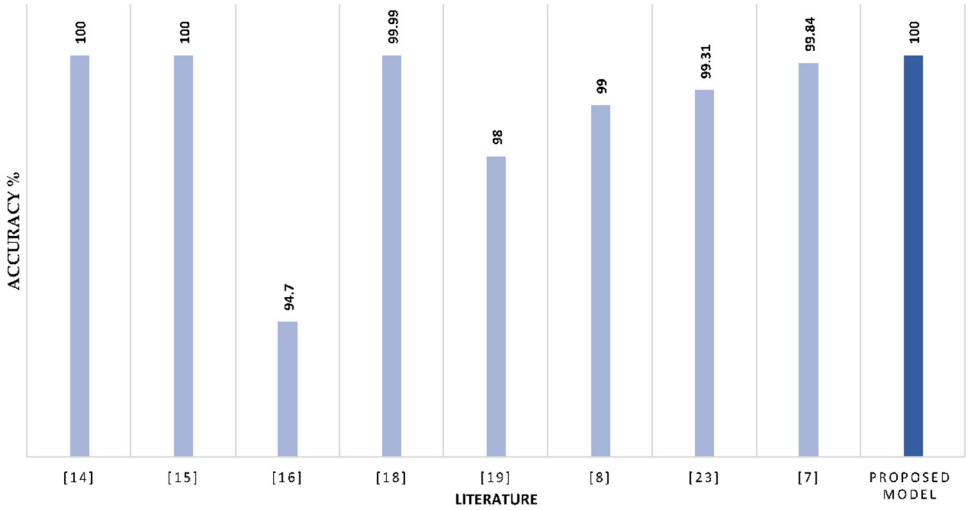

Figure 10. Binary Classification Literature Comparison.

2. Multi-Class Category Classification: Out of the previously outlined literature, only three papers evaluated their models against multi-class category classification. Figure 11 outlines the results of the literature in comparison with the proposed IDS model. The proposed stacking IDS model performed better than the literature in the category classification task. The proposed model achieved an accuracy of 100% and recall, precision, and F1-score of 99%.

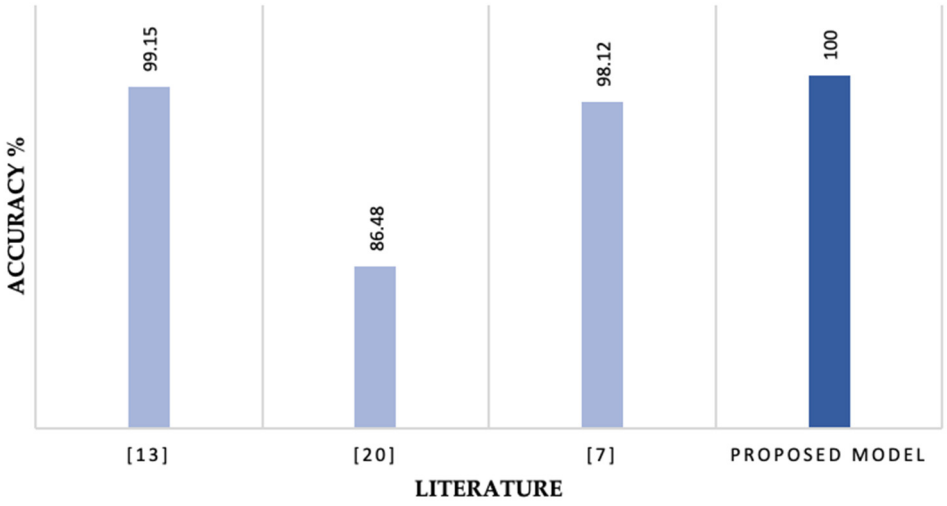

Figure 11. Category Classification Literature Comparison.

 doi:10.20944/preprints202404.0499.v1

3. Multi-Class Subcategory Classification: Out of the previously outlined literature, four papers evaluated their models against multi-class subcategory classification. Figure 12 presents the results of the literature in comparison with the proposed IDS model.

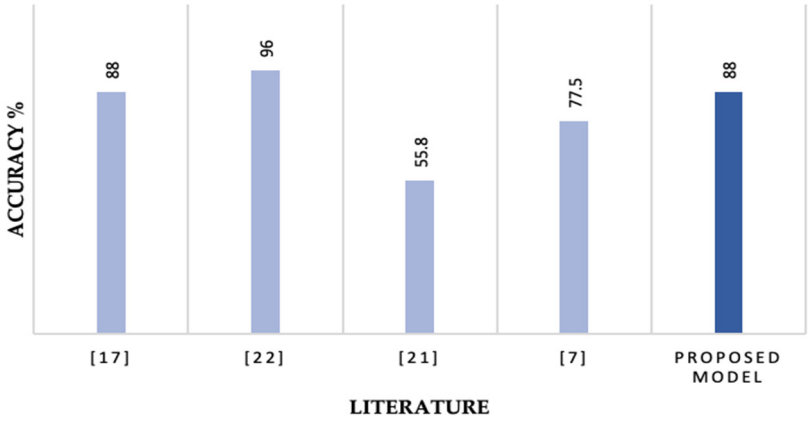

Figure 12. Subcategory Multi-Class Classification Literature Comparison.

The main focus of the paper is to detect subcategory classification to be able to secure IoT environments. We should be able to detect what type of attack was initiated in order to better respond. A summarized comparison between the proposed models and earlier studies is given below.

- Researchers in [17] evaluated the IoTID20 dataset on six ML classifiers. Out of the six classifiers, the decision tree achieved the best result with 88%, followed by an ensemble method with 87%.
- Researchers in [22] evaluated the IoTID20 dataset on four classifiers. Of the four, DT performed the highest detection rate in detecting 6 of the nine classes. Figure 13 displays the ROC-AUC curve for the multi-class subcategory classification of this study. The proposed Ensemble model in this study achieved similar results to the DT classifier in [22] in terms of AUC except for the Scan host port class, where in [22], AUC is 96 compared to 81 in this study.
- The proposed IDS model achieved better results than both [21] and [7].

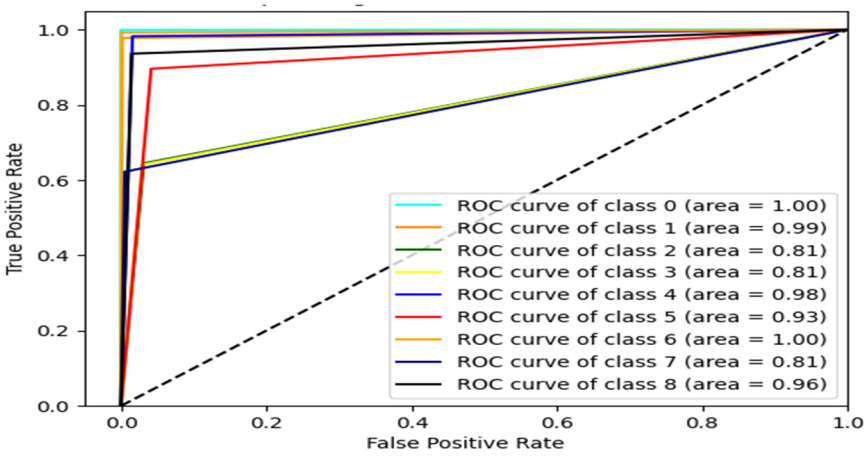

Figure 13. ROC-AUC curve of proposed IDS model.

5. Conclusions

In this work, we developed a machine learning-based intrusion detection approach to detect malicious traffic in IoT networks and identify further details about the attack, which we referred to as the attack category and subcategory. We trained different ML algorithms on the IoTID20 dataset to perform binary and multi-class anomaly detection tasks. Our experiments led to the development

 doi:10.20944/preprints202404.0499.v1

of a staking ensemble anomaly detection model that achieved robust detection results. In particular, our binary classifier surpasses the results reported by earlier studies. Moreover, our multi-class classification models achieved F1-scores of 99% and 88% in the multi-class category and subcategory classification tasks, respectively. These results suggest the proposed ensemble model can reliably detect anomalies in IoT devices. While IoT devices produce massive data that enhance the potential of machine learning-powered IoT security solutions, the practical applicability of these solutions is highly challenged by the class imbalance with noisy data problems. Hence, our foreseen future work is focused on alleviating the adverse effect of class imbalance, using various preprocessing and feature selection techniques.

References

1. Chin, J.; Callaghan, V.; Allouch, S.B. The Internet-of-Things: Reflections on the Past, Present and Future from a User-Centered and Smart Environment Perspective. *Journal of Ambient Intelligence and Smart Environments* **2019**, *11*, 45–69, doi:10.3233/AIS-180506.
2. Winchcomb, T.; Massey, S.; Beastall, P. *Review of Latest Developments in the Internet of Things*; 2017;
3. Cyrus, C. IoT Cyberattacks Escalate in 2021, According to Kaspersky Available online: https://www.iotworldtoday.com/security/iot-cyberattacks-escalate-in-2021-according-to-kaspersky (accessed on 26 April 2023).
4. Abomhara, M.; Køien, G.M. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. *Journal of Cyber Security and Mobility* **2015**, 65–88, doi:10.13052/jcsm2245-1439.414.
5. Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering* **2017**, *2017*, e9324035, doi:10.1155/2017/9324035.
6. Al-Attar, R.; Alkasassbeh, M.; Al-Dala'Ien, M. A Survey: Soft Computing for Anomaly Detection to Mitigate IoT Abuse. In Proceedings of the 2022 International Conference on Engineering & MIS (ICEMIS); July 2022; pp. 1–6.
7. Ullah, S.; Ahmad, J.; Khan, M.A.; Alkhammash, E.H.; Hadjouni, M.; Ghadi, Y.Y.; Saeed, F.; Pitropakis, N. A New Intrusion Detection System for the Internet of Things via Deep Convolutional Neural Network and Feature Engineering. *Sensors* **2022**, *22*, 3607, doi:10.3390/s22103607.
8. Omar, M.; George, L. Toward a Lightweight Machine Learning Based Solution against Cyber-Intrusions for IoT. In Proceedings of the 2021 IEEE 46th Conference on Local Computer Networks (LCN); October 2021; pp. 519–524.
9. Yu, X.; Yang, X.; Tan, Q.; Shan, C.; Lv, Z. An Edge Computing Based Anomaly Detection Method in IoT Industrial Sustainability. *Applied Soft Computing* **2022**, *128*, 109486, doi:10.1016/j.asoc.2022.109486.
10. Shafiq, M.; Tian, Z.; Bashir, A.K.; Du, X.; Guizani, M. IoT Malicious Traffic Identification Using Wrapper-Based Feature Selection Mechanisms. *Computers & Security* **2020**, *94*, 101863, doi:10.1016/j.cose.2020.101863.
11. Lu, H.; Jin, C.; Helu, X.; Du, X.; Guizani, M.; Tian, Z. DeepAutoD: Research on Distributed Machine Learning Oriented Scalable Mobile Communication Security Unpacking System. *IEEE Transactions on Network Science and Engineering* **2022**, *9*, 2052–2065, doi:10.1109/TNSE.2021.3100750.
12. Shafiq, M.; Tian, Z.; Bashir, A.K.; Du, X.; Guizani, M. CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques. *IEEE Internet Things J.* **2021**, *8*, 3242–3254, doi:10.1109/JIOT.2020.3002255.
13. Dat-Thinh, N.; Xuan-Ninh, H.; Kim-Hung, L. MidSiot: A Multistage Intrusion Detection System for Internet of Things. *Wireless Communications and Mobile Computing* **2022**, *2022*, e9173291, doi:10.1155/2022/9173291.
14. Islam, N.; Farhin, F.; Sultana, I.; Shamim Kaiser, M.; Sazzadur Rahman, Md.; Mahmud, M.; S. M. Sanwar Hosen, A.; Hwan Cho, G. Towards Machine Learning Based Intrusion Detection in IoT Networks. *Computers, Materials & Continua* **2021**, *69*, 1801–1821, doi:10.32604/cmc.2021.018466.
15. Indrasiri, P.L.; Lee, E.; Rupapara, V.; Rustam, F.; Ashraf, I. Malicious Traffic Detection in Iot and Local Networks Using Stacked Ensemble Classifier. *Computers, Materials and Continua* **2022**, *71*, 489–515, doi:10.32604/cmc.2022.019636.
16. Song, Y.; Hyun, S.; Cheong, Y.-G. Analysis of Autoencoders for Network Intrusion Detection. *Sensors* **2021**, *21*, 4294, doi:10.3390/s21134294.
17. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In Proceedings of the Advances in Artificial Intelligence; Goutte, C., Zhu, X., Eds.; Springer International Publishing: Cham, 2020; pp. 508–520.

18. Maniriho, P.; Niyigaba, E.; Bizimana, Z.; Twiringiyimana, V.; Mahoro, L.J.; Ahmad, T. Anomaly-Based Intrusion Detection Approach for IoT Networks Using Machine Learning. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM); November 2020; pp. 303–308.
19. Alkahtani, H.; Aldhyani, T.H.H. Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms. *Complexity* **2021**, *2021*, e5579851, doi:10.1155/2021/5579851.
20. Qaddoura, R.; M. Al-Zoubi, A.; Faris, H.; Almomani, I. A Multi-Layer Classification Approach for Intrusion Detection in IoT Networks Based on Deep Learning. *Sensors* **2021**, *21*, 2987, doi:10.3390/s21092987.
21. Qaddoura, R.; Al-Zoubi, A.M.; Almomani, I.; Faris, H. Predicting Different Types of Imbalanced Intrusion Activities Based on a Multi-Stage Deep Learning Approach. In Proceedings of the 2021 International Conference on Information Technology (ICIT); July 2021; pp. 858–863.
22. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432, doi:10.3390/s21196432.
23. Krishnan, S.; Neyaz, A.; Liu, Q. IoT Network Attack Detection Using Supervised Machine Learning. **2021**.
24. Mohammed, R.; Rawashdeh, J.; Abdullah, M. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. In Proceedings of the 2020 11th International Conference on Information and Communication Systems (ICICS); April 2020; pp. 243–248.
25. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* **2002**, *16*, 321–357, doi:10.1613/jair.953.
26. Pearson, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **1901**, *2*, doi:https://doi.org/10.1080/14786440109462720.
27. Kohavi, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. **2001**, *14*.
28. Rincy, T.N.; Gupta, R. Ensemble Learning Techniques and Its Efficiency in Machine Learning: A Survey. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA); February 2020; pp. 1–6.
29. Yin, X.; Liu, Q.; Pan, Y.; Huang, X.; Wu, J.; Wang, X. Strength of Stacking Technique of Ensemble Learning in Rockburst Prediction with Imbalanced Data: Comparison of Eight Single and Ensemble Models. *Nat Resour Res* **2021**, *30*, 1795–1815, doi:10.1007/s11053-020-09787-0.