INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 2, PP. 465-472
Manuscript received November 8, 2023; revised June, 2024. doi: 10.24425/ijet.2024.149567

Ensemble learning approach to enhancing binary classification in Intrusion Detection System for Internet of Things

Soni, Muhammad Akmal Remli, Kauthar Mohd Daud, and Januar Al Amien

Abstract—**The Internet of Things (IoT) has experienced significant growth and plays a crucial role in daily activities. However, along with its development, IoT is very vulnerable to attacks and raises concerns for users. The Intrusion Detection System (IDS) operates efficiently to detect and identify suspicious activities within the network. The primary source of attacks originates from external sources, specifi-cally from the internet attempting to transmit data to the host network. IDS can identify unknown attacks from network traffic and has become one of the most effective network security. Classification is used to distinguish between normal class and attacks in binary classification problem. As a result, there is a rise in the false positive rates and a decrease in the detection accuracy during the model's training. Based on the test results using the ensemble technique with the ensemble learning XGBoost and LightGBM algorithm, it can be concluded that both binary classification problems can be solved. The results using these ensemble learning algorithms on the ToN IoT Dataset, where binary classification has been performed by combining multiple devices into one, have demonstrated improved accuracy. Moreover, this ensemble approach ensures a more even distribution of accuracy across each device, surpassing the findings of previous research.**

Keywords—**Binary classification; XGBoost; ToN IoT dataset; ensemble technique**

I. Introduction

THE Internet of Things (IoT) is reshaping our daily experiences by enabling the control of physical devices at the periphery and connecting both tangible and digital entities to enhance our everyday routines. This applies to various applications including Industrial IoT (IIoT), intelligent residences and urban environments, as well as advanced power distribution networks [1][2]. In recent years, the Internet of Things (IoT) has experienced significant expansion and integration into various aspects of daily life [3][4]. The Internet of Things (IoT) is a technology that allows physical objects to be interconnected and equipped with internet connectivity and processing capabilities. IoT has the potential to provide versatile and effective solutions in various fields, including healthcare, environmental monitoring, and industrial control systems [5]. IoT devices are something that is very much intensified by all groups, because IoT makes it easier for humans to carry out tasks and helps humans in terms of automation[6].

The cybersecurity domain has recently seen increased attention toward discussing the potential dangers posed to IoT applications and the imperative of mitigating these risks. Certain IoT applications, often referred to as Industrial IoT within the context of Industry 4.0 advancements, encompass critical functions like overseeing industrial control and infrastructure systems, demanding a heightened level of protection. It has been reported that in a recent cyberattack targeting IIoT applications, multiple electrical substations in Ukraine were breached, leading to a power outage that impacted approximately 225,000 customers [7]. Cybercriminal activities on the internet primarily involve communication attacks. Detecting malicious network traffic while minimizing costs has become a challenging problem for information security experts [8]. Intrusion detection systems (IDS) are widely used as: second line of defense for monitoring a system or network events to detect possible successful malicious activity circumvent security perimeters (eg, firewalls) [9].

Figure 1 illustrates the Internet of Things that is connected to the network, in the figure there are several hardware and software devices that are related to each other.

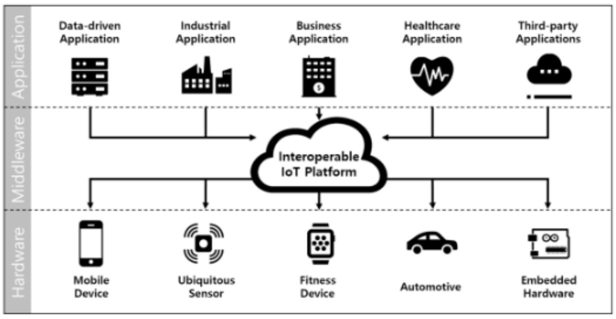

Fig. 1. Internet of Things applications

Soni is with Faculty of Computer Sciences, Universitas Muhammadiyah Riau, Pekanbaru, Riau Indonesia and Faculty of Data Science and Computing, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, 16100 Kota Bharu, Kelantan, Malaysia (e-mail: soni@umri.ac.id).

Muhammad Akmal Remli is with Faculty of Data Science and Computing, Universiti Malaysia Kelantan and Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, 16100 Kota Bharu, Kelantan, Malaysia (e-mail: akmal@umk.edu.my).

Kauthar Mohd Daud is with Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi Selangor, Malaysia (e-mail: kauthar.md@ukm.edu.my).

Januar Al Amien is Universitas Muhammadiyah Riau, Pekanbaru, Riau Indonesia and Faculty of Data Science and Computing, Universiti Malaysia Kelantan, City Campus, Pengkalan Chepa, 16100 Kota Bharu, Kelantan, Malaysia (e-mail: januaralamien@umri.ac.id).

With the increase in crime on the internet network, it is necessary to do further research. In detecting crimes that occur, many methods are used, such as a machine learning approach. Machine Learning (ML) approach is used to study datasets in classifying algorithms and to predict accurate ones.

Unbalanced data occurs when one class has a significantly larger sample size than another, leading to the possibility of minority samples being disregarded as noise during the classification process, which can result in less than satisfactory algorithmic outcomes. Addressing unbalanced data involves employing oversampling techniques to create synthetic minority samples and implementing undersampling techniques to reduce the majority class, effectively tackling the issue of class imbalance [10].

Because there is an imbalance in the ToN IoT Dataset, the author uses the ADASYN technique to balance the data in the imbalanced class. Presently, numerous specialists have integrated the SMOTE over-sampling algorithm into the realm of intrusion detection systems. Nonetheless, when generating a minority sample, SMOTE does not take into account the context surrounding the minority sample point[11]. When dealing with a minority sample, if there are numerous samples from the same class in its vicinity, that particular sample might be considered noise. When newly generated samples overlap significantly with the neighboring majority samples, it can pose challenges during the classification process. SMOTE generates an equal number of new samples for each minority sample, whereas the ADASYN algorithm employs a sample distribution to automatically calculate the number of synthesized samples for each minority sample. This approach aims to prevent sample overlap. [10]. The performance of the over-sampling technique is better overall. Among the over-sampling techniques, ADASYN's performance is relatively better [12].

This Adasyn method enhances learning about data distribution in two ways. Reduces bias caused by unbalance classes and adjusts boundary classification decisions for difficult examples. Synthetic oversampling such as the adasyn technique adds samples to produce samples of the synthetic minority class. It can improve classifier efficiency[13].

The TON_IoT dataset is a dataset that can be used as a comparison of various classification methods for system intrusion detection. This dataset has 7 datasets consisting of several IoT devices. Feature selection is one of the important pre-processes for reducing datasets by removing unimportant features from the TON_IoT dataset. Not all features in the dataset have an influence on the label class. Therefore eliminating unimportant attributes with class labels is an important thing to do to improve classifier performance. The main purpose of feature selection is to select important features and remove unimportant and less important features from class labels in order to improve classifier performance, namely increasing accuracy and reducing computation time. The dataset has a problem, namely the IoT data set which shows normal and attack classes, and types which indicate attack subclasses targeting IoT devices for binary classification problems.

XGBoost is an improved algorithm rooted in gradient boosting decision trees, designed to efficiently construct boosted trees and work concurrently. It serves as a machine learning method to address regression and classification challenges, relying on the principles of Gradient Boosting Decision Trees [14].

In this study, researchers used the ToN IoT Dataset. From the results of combining datasets, there are class imbalance problems, categorical features, and missing values. The TON_IoT dataset is a dataset that can be used as a comparison of various classification methods for system intrusion detection. This dataset has 7 datasets consisting of several IoT devices [15]. Modbus, Motion Light, Thermostat, and Weather. The ToN-IoT dataset poses a binary classification challenge, characterized by the issue of class imbalance. To address this problem, various techniques including oversampling, under-sampling, and hybrid methods have been proposed as potential solutions. Oversampling entails duplicating minority class instances. However, it's important to note that the ToN-IoT dataset presents several challenges, including class imbalance, the presence of categorical features, and missing values [16].

This study assesses the effectiveness of data-driven intrusion detection techniques in addressing binary classification challenges. It leverages various supervised machine learning approaches utilizing datasets derived from the ToN IoT Dataset. Subsequently, the data from all IoT devices are amalgamated into a single dataset referred to as the Combined IoT dataset, which is then evaluated for binary classification tasks. The primary focus of this research contribution is:

a. Evaluate the performance of machine learning methods in binary-classification problems on the ToN IoT Dataset
b. Comparing the xgboost and lightgbm ensemble learning model with previous machine learning using evaluation metric accuracy, precision, recall, F-measure.

II. LITERATURE REVIEW

Most of the recently published datasets [15][1]. Classifies IoT devices with seven datasets for 7 datasets consisting of Garage Door, Fridge , GPS Tracker, Modbus, Motion Light, Fridge, Thermostat, and Weather. In the machine learning approach, many techniques are used such as Logistic Regression (LR) result in 0.61% accuracy, Linear Discriminant Analysis (LDA) result in 0.68% accuracy, k-Nearest Neighbour (kNN) result in 0.84% accuracy, Random Forest (RF) result in, 0.85% accuracy, Classification and Regression Trees (CART) result in 0.88% accuracy, Naïve Bayes (NB) result in 0.62% accuracy, Support Vector Machine (SVM) result in 0.61% accuracy and Long Short-Term Memory (LSTM) result in 0.81% accuracy [15].

Subsequent research conducted classification using the Voting Classifier on the ToN IoT dataset. The results demonstrated remarkable accuracy, with Global Positioning System (GPS) sensors and weather sensors achieving up to 96% and 97%, respectively, while other machine learning algorithms reached up to 85% and 87% accuracy, respectively[1]. Research conducted by [17] using the Ensemble Learning model and producing a performance analysis accuracy of 94.5%. Other research conducted [16] using models LR, NB, DT, RF, AdaBoost, kNN, SVM, XGBoost and produce Accuracy XGBoost 98%. The following are some previous studies that have been conducted on IoT datasets and classifications using machine learning. These studies can serve as references and are summarized in Table I below.

TABLE I
TABLE OF PREVIOUS RESEARCH ON DATASETS

No	Model	Dataset	Performance Analisis	References
1	LR, LDA, kNN, RF, CART, NB, SVM, LSTM	ToN IoT Dataset	CART 88%	[15]
2	Ensemble Learning	RPL-NIDDS17	94.5%	[17]
3	LR, NB, DT, RF, AdaBoost, kNN, SVM, XGBoost	Ton IoT Network Dataset	XGBoost 98%	[16]
4	XGBoost	NSL-KDD, UNSW-NB15,	95.55	[18]
5	XGBoost	ToN IoT Network	96.35%	[19]
6	DT-RFkNN-NB, DT-RFNB,DT-RFkNN	ToN IoT Dataset	76%	[1]
7	DT-RFkNN-NB, DT-RFNB, DT-RFkNN	ToN IoT Network Dataset	88%	[1]

III. RESEARCH METHODOLOGY

The framework offers detailed step-by-step procedures implemented at every stage of the research. Additionally, it introduces the required datasets for testing and evaluating the methods. Finally, various metrics for assessing the performance of the proposed method are presented, as depicted in Figure 2.

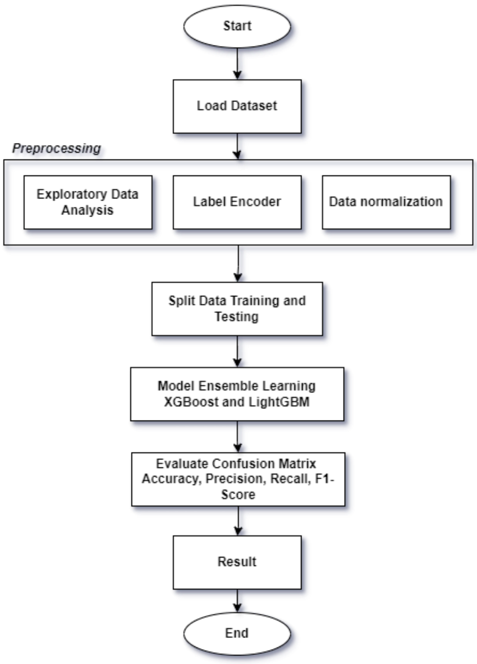
Start
Load Dataset
Preprocessing
Exploratory Data Analysis
Label Encoder
Data normalization
Split Data Training and Testing
Model Ensemble Learning XGBoost and LightGBM
Evaluate Confusion Matrix Accuracy, Precision, Recall, F1-Score
Result
End

Fig. 2 Method Ensemble Learning

A. Dataset

The dataset used is the ToN IoT Dataset, which presents challenges related to class imbalance in both binary and multiclass classification problems. It comprises seven IoT training datasets, encompassing Fridge, Garage Door, GPS Tracker, Modbus, Motion Light, Thermostat, and Weather. The ToN-IoT datasets can be accessed at ToN-IoT repository. TON_IoT datasets Provided by Nour Mustafa on https://research.unsw.edu.au/projects/toniot-datasets [15] [20].

B. Data Preprocessing

Preprocessing is carried out to classify the data in this dataset. The dataset is not well-structured due to the presence of non-numeric data. In order for the data to be usable, it needs to go through several preprocessing stages. Additionally, there are additional steps that can be taken, such as handling features that contain categories (both main categories and subcategories) using Label Encoder.

The steps involved in this process are:

1. Identifying non-numeric data: Firstly, identifying variables in the dataset that are not numeric or contain text, categories, or symbols.
2. Initial pre-processing: Performing initial pre-processing steps, such as removing missing or duplicate data, handling missing values, and cleaning inconsistent data.
3. Converting categories into numeric: Using Label Encoder or similar methods to convert categorical variables (such as category and subcategory types) into numeric representations that can be used by machine learning algorithms.
4. Handling variable scales: Checking and adjusting the scales of variables in the dataset to have a similar range, so that no variable dominates over others.
5. Additional pre-processing: If necessary, performing additional preprocessing steps such as data normalization, addressing outliers, or data transformation to improve distribution.

These steps are taken to ensure that the data in the dataset is ready for use in the classification process, enabling machine learning models to produce more accurate results.

C. Model Ensemble Learning XGBoost and LightGBM

The ensemble method is a machine learning approach that combines multiple fundamental models to diminish the occurrence of false positives and enhance accuracy compared to using a single model. To tackle this challenge, our solution incorporates an ensemble classifier to mitigate bias among diverse training datasets. This fusion of feature selection and ensemble classifier is aimed at enhancing the stability and accuracy of the IDS while maintaining low computational complexity and time requirements[21].

1. XGBoost (eXtreme Gradient Boosting)

XGBoost is founded on the principles of gradient-boosted decision trees and is recognized for its exceptional speed and high performance compared to other machine learning techniques. It serves as a technique to enhance the capabilities of machine learning models, with a specific focus on tree boosting methods. XGBoost, which stands for Extreme Gradient Boosting, excels in efficient memory and hardware resource utilization, contributing to algorithm improvement and model refinement. By utilizing Taylor expansion on the cost function with consideration of the second derivative, XGBoost ensures heightened result accuracy. This approach optimizes the objective function through an iterative training process, where each subsequent phase's optimization relies on the outcomes of the preceding stage[16] [22][23].

Equation's ensemble model [24], which has parameters that are functions, makes it impossible to optimize it using conventional Euclidean-space techniques. The model is instead trained in an additive way. Formally, if $\hat{y}_i^{(t)}$ represents the forecast for the *i*-th instance at the *t*-th iteration, we must add $\mathrm{f_t}$ to reduce the next goal.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} I\left(\hat{y}_{\mathrm{i}}, y_{\mathrm{i}}{}^{(t-1)} + f_t(x_{\mathrm{i}})\right) + \Omega(f_t). \quad (1)$$

Utilizing a second-order approximation allows for the efficient optimization of the objective in a broader context.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} [l\left(\hat{y}, y_{\mathrm{i}}{}^{(t-1)}\right) + g_{\mathrm{i}}\, f_t(x_{\mathrm{i}}) + \frac{1}{2}\, hi\; ft^2(x_{\mathrm{i}})] + \Omega(f_t). \quad (2)$$

Where $gi = \partial\hat{y}^{(t-1)} l\left(\hat{y}, y_{\mathrm{i}}{}^{(t-1)}\right)$ and $hi = \partial^2\hat{y}^{(t-1)} l\left(\hat{y}, y_{\mathrm{i}}{}^{(t-1)}\right)$ are the loss function's first and second order gradient statistics. At step t, we can eliminate the constant terms to get the next simplified aim:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} (g_{\mathrm{i}} f_t(x_{\mathrm{i}}) + \frac{1}{2} h_{\mathrm{i}} f_t^2(x_{\mathrm{i}})] + \Omega(f_t) \quad (3)$$

For a xed structure q(x), we may determine the ideal leaf j weight w_j^* by

$$w_j^* = -\frac{(\sum_{i\in Ij} gi)^2}{\sum_{i\in Ij} hi + \lambda} \quad (4)$$

and determine the associated ideal value by

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i\in Ij} gi)^2}{\sum_{i\in Ij} hi + \lambda} + \gamma T \quad (5)$$

Typically, it's not feasible to list down every potential tree configuration *q*. A Instead, a greedy algorithm is employed, commencing with a solitary leaf and systematically appending branches to construct the tree. Assume that I_L dan I_R are the instance sets of left and right nodes after the split. Lettting I $=I_L \cup I_R$, then the loss reduction after the split is given by

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in Lj} gi)^2}{\sum_{i\in IL} hi + \lambda} + \frac{(\sum_{i\in Lj} gi)^2}{\sum_{i\in IR} hi + \lambda} - \frac{(\sum_{i\in Lj} gi)^2}{\sum_{i\in} hi + \lambda}\right] - \gamma \quad (6)$$

In practice, this equation is commonly employed to assess the suitability of potential split candidates.

2. LightGBM

LightGBM is a Gradient Boosting Decision Tree (GBDT) model that integrates two techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). Traditional GBDT models are often characterized by lengthy training times, with a significant portion of this duration dedicated to identifying the optimal split points. To address this challenge, LightGBM adopts the histogram algorithm for both feature selection within decision trees and the determination of split points. This algorithm discretizes the initial continuous feature values into bins, which are subsequently employed in constructing the model. As a result, the histogram approach significantly reduces the time required for selecting split points, leading to improved efficiency in both the training and prediction processes[25]. The algorithm that relies on histogram construction within LightGBM is illustrated in:

Pseudo-code of LightGBM [26]

```
Input: I:training data
       d:max depth
       m:feature dimension
Initialize:nodeSet ← {0}:tree nodes in current level
Initialize:rowSet ← {{1, 2, 3, ...}}:data indices in tree nodes
for i = 1, . . . , d do
    for node in nodeSet do
        usedRows ← rowSet[node]
        for k = 1, . . . , m do
            H ← new Histogarm()
        > Build histogram
            for j in useRows do
                bin ←I.f[k][j].bin
                H[bin].y ←
                H[bin].y+I.y[j]
                H[bin].n ← H[bin].n+1
            end for
            Find the best split on histogram H
        end for
end for
Update rowSet and nodeSet according to the best split points
end for
```

D. Evaluate Confusion Matrix

Confusion matrix is an instrument used to evaluate performance of the resulting classification model. These results will then be used calculate accuracy, precision, recall and f1-score values [27] [28]. According to [29][30], Several measurement metrics are relevant. Given the widespread consensus that accuracy alone does not offer a sufficient means of performance assessment, we furnish values for a majority of these metrics, particularly when the datasets feature a surplus of positive examples over negative ones.

Accuracy (AC): represents the proportion of accurate model classifications relative to the total number of classifications made. In the context of binary scenarios:

$$Accuracy\ \% = \frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

Precision (PR): denotes the relationship between accurate predictions and the total predictions made for a specific class. A high Precision value is associated with a reduced occurrence of false alarms. In the context of binary situations:

$$Precision\ \% = \frac{TP}{TP+FP} \quad (2)$$

Recall (RC): represents the relationship between accurate predictions and the total instances within a specific class. A high Recall value suggests that the majority of samples in a class have been correctly identified. In the context of binary scenarios:

$$Recall\ \% = \frac{TP}{TP+FN} \quad (3)$$

F1 Score (F1): PR and RC metrics pose conflicting demands because improving one may lead to a compromise in the other. The F1 Score is the harmonic mean of these two metrics. In the context of binary situations:

$$F1 = 2 * \frac{Presisi * Recall}{Presisi + Recall} \quad (4)$$

The false negative rate (FNR) is the proportion of false negative instances to the total count of actual positive cases. It signifies the system's inability to identify genuine positive outcomes. When this value is elevated, it means that legitimate threats go undetected, rendering the system susceptible to exploitation by malicious users and jeopardizing its security. Consequently, there is a pressing need to minimize the FNR to the utmost extent possible.

$$FNR\ \% = \frac{FN}{TP+FN} \tag{5}$$

The false positive rate (FPR), also known as the false alarm rate (FAR), signifies the ratio of false positive occurrences compared to the total number of authentic negative instances. When this metric consistently remains high, it may cause security analysts to intentionally disregard system alerts, which could ultimately lead to the system becoming vulnerable or precarious. [31]. Hence, it is imperative to minimize it to the utmost extent possible.

$$FPR\ \% = \frac{FP}{FP+TN} \tag{6}$$

When taking a broader view of metrics, accuracy emerges as the paramount factor. It signifies the ratio of correctly assigned individual samples to the total samples, reflecting the confidence level of the classification. Consequently, it should be maximized to its fullest potential.

IV. EXPERIMENTAL AND DISCUSSION

The results of the entire dataset tested on Binary Classification which displays Accuracy, Precision, Recall, F-Score, Train Time and Test Time. The accuracy obtained from the seventh dataset can be seen in the fol-lowing table which shows performance evaluation metrics in machine learning. The classification report is used to show precision, recall, f1-score, and support in table II below.

TABLE II

BINARY CLASSIFICATION RESULTS FOR EACH TON OF IOT DATASET

Dataset	Our Propose	Accuracy %	Precision %	Recall %	F1-score	Time Sec
Fridge sensor	XGBoost	1.00	1.00	1.00	1.00	0.59
	LightGBM	1.00	1.00	1.00	1.00	0.14
Garage door	XGBoost	1.00	1.00	1.00	1.00	1.24
	LightGBM	1.00	1.00	1.00	1.00	0.85
GPS Tracker	XGBoost	0.95	0.96	0.93	0.94	5.97
	LightGBM	0.92	0.92	0.93	0.92	0.19
Modbus	XGBoost	0.98	0.99	0.95	0.97	8.57
	LightGBM	0.72	0.81	0.54	0.50	0.18
Motion Light	XGBoost	0.92	0.92	0.89	0.90	10.19
	LightGBM	0.92	0.93	0.92	0.92	0.23
Thermostat	XGBoost	0.95	0.94	0.91	0.93	12.76
	LightGBM	0.95	0.95	0.93	0.94	0.19
Weather	XGBoost	0.99	0.99	0.98	0.99	18.01
	LightGBM	0.89	0.89	0.88	0.88	0.20

This section describes the results of proposed model, namely the XGBoost algo-rithm and compares it with the results of previous studies. The eXtreme Gradient Boosting (XGBoost) algorithm can be used to classify IoT device datasets such as: fridge, garage door, gps sensor, motion light, modbus, thermostat, and weather. The seven datasets are combined into 1 dataset and given the name Combined IoT. In the binary classification test, the accuracy is 99%. When compared with previous research, it can be said that the eXtreme Gradient Boosting algorithm is able to pro-duce better and maximum accuracy than other algorithms that have been studied previously. The results of the overall comparison can be seen in table III below.

TABLE III

COMPARISON TABLE OF BINARY CLASSIFICATION ON EACH DATASET WITH PREVIOUS RESEARCH

		Result at machine learning [15]								**Different combinations** [1]			**Our proposed Ensemble Learning models**	
Datasets	**Evaluation metrics**	**LR**	**LDA**	**kNN**	**RF**	**CART**	**NB**	**SVM**	**LSTM**	**(DT-RF-kNN-NB)**	**(DT-RF-NB)**	**(DT-RF-kNN)**	**XGBoost**	**LightGBM**
Fridge sensor	Accuracy	0.57	0.77	0.99	0.97	0.97	0.50	0.81	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	Precision	0.34	0.79	0.99	0.97	0.97	0.53	0.86	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	Recall	0.58	0.77	0.99	0.97	0.97	0.51	0.82	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	F1-score	0.43	0.77	0.99	0.97	0.97	0.51	0.80	1.00	1.00	1.00	1.00	**1.00**	**1.00**
Garage door	Accuracy	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**
	F1-score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	**1.00**
GPS Tracker	Accuracy	0.8	0.86	0.88	0.85	0.84	0.84	0.86	0.88	0.96	0.95	0.96	**0.95**	**0.92**
	Precision	0.8	0.88	0.89	0.85	0.85	0.86	0.88	0.89	0.96	0.96	0.96	**0.96**	**0.92**
	Recall	0.8	0.86	0.88	0.85	0.85	0.85	0.87	0.88	0.96	0.96	0.96	**0.93**	**0.93**
	F1-score	0.8	0.87	0.88	0.85	0.85	0.86	0.87	0.88	0.96	0.96	0.96	**0.94**	**0.92**
Modbus	Accuracy	0.6	0.67	0.77	0.90	0.98	0.67	0.67	0.67	0.97	0.96	0.96	**0.98**	**0.71**
	Precision	0.4	0.46	0.77	0.90	0.99	0.46	0.46	0.47	0.97	0.97	0.97	**0.99**	**0.81**
	Recall	0.6	0.68	0.78	0.98	0.98	0.68	0.68	0.68	0.97	0.97	0.97	**0.95**	**0.54**
	F1-score	0.5	0.55	0.77	0.98	0.99	0.55	0.55	0.55	0.97	0.97	0.97	**0.97**	**0.50**
Motion Light	Accuracy	0.5	0.58	0.54	0.58	0.58	0.58	0.58	0.59	0.58	0.58	0.58	**0.92**	**0.92**
	Precision	0.3	0.34	0.34	0.34	0.34	0.34	0.34	0.35	0.34	0.35	0.34	**0.92**	**0.95**
	Recall	0.5	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.58	**0.89**	**0.92**
	F1-score	0.4	0.43	0.43	0.43	0.43	0.43	0.43	0.44	0.43	0.44	0.43	**0.90**	**0.92**
Thermostat	Accuracy	0.6	0.66	0.60	0.66	0.59	0.66	0.66	0.66	0.66	0.66	0.66	**0.95**	**0.95**
	Precision	0.4	0.44	0.56	0.59	0.56	0.44	0.44	0.45	0.78	0.54	0.61	**0.94**	**0.95**
	Recall	0.6	0.66	0.61	0.66	0.59	0.66	0.66	0.67	0.66	0.66	0.67	**0.91**	**0.93**

		Result at machine learning [15]								**Different combinations** [1]			**Our proposed Ensemble Learning models**	
Datasets	**Evaluation metrics**	**LR**	**LDA**	**kNN**	**RF**	**CART**	**NB**	**SVM**	**LSTM**	**(DT-RF-kNN-NB)**	**(DT-RF-NB)**	**(DT-RF-kNN)**	**XGBoost**	**LightGBM**
	F1-score	0.5	0.53	0.57	0.53	0.57	0.53	0.53	0.54	0.53	0.53	0.65	**0.93**	**0.94**
Weather	Accuracy	0.5	0.60	0.81	0.84	0.87	0.59	0.63	0.82	0.97	0.97	0.98	**0.99**	**0.89**
	Precision	0.6	0.59	0.81	0.84	0.88	0.72	0.68	0.82	0.98	0.97	0.98	**0.99**	**0.89**
	Recall	0.5	0.60	0.81	0.84	0.87	0.59	0.63	0.81	0.97	0.97	0.98	**0.98**	**0.88**
	F1-score	0.5	0.53	0.81	0.84	0.87	0.67	0.55	0.80	0.97	0.97	0.98	**0.99**	**0.88**

From the given table of evaluation metrics for different machine learning models on various datasets, we can draw the following conclusions:

1. Fridge sensor and garage door dataset: LSTM, (DT-RF-kNN-NB), (DT-RF-NB), (DT-RF-kNN), XGBoost, and LightGBM models perform exceptionally well, achieving high accuracy, precision, recall, and F1-score. LDA model also performs well with high precision, recall, and F1-score.
2. GPS Tracker dataset: RF, XGBoost, and LightGBM models exhibit the best per-formance with high accuracy, precision, recall, and F1-score. LSTM, (DT-RF-kNN-NB), (DT-RF-NB), and (DT-RF-kNN) models also provide good results with high accuracy, precision, recall, and F1-score.
3. Modbus: RF, XGBoost, and LightGBM models deliver the best performance with high accuracy, precision, recall, and F1-score. NB and SVM models also provide good results with high accuracy, precision, recall, and F1-score
4. Motion Light dataset: XGBoost and LightGBM models demonstrate superior performance with high accuracy, precision, recall, and F1-score.
5. Thermostat dataset: LSTM, XGBoost, and LightGBM models achieve the best performance with high accuracy, precision, recall, and F1-score.
6. Weather dataset: RF, XGBoost, and LightGBM models exhibit the best perfor-mance with high accuracy, precision, recall, and F1-score.

Overall, performance of machine learning models varies depending on the dataset used. However, certain models consistently yield good results across multiple da-tasets, such as RF, XGBoost, and LightGBM. Every IoT dataset is amalgamated into a single CSV file called " Combined ToN IoT Dataset." An automated Python script is employed to perform the consolidation of all IoT datasets into this CSV file. Subsequently, the median value for each column is utilized as an imputation method to populate missing values within that particular column. The selection of the median is recommended due to its reduced susceptibility to outlier influences when compared to using the mean for imputation. The distribution of classes within the " Combined ToN IoT Dataset" is presented in Table IV.

TABLE IV
COMPARISON TABLE OF BINARY CLASSIFICATION ON COMBINED TON IOT DATASET WITH PREVIOUS RESEARCH

		Performance of the state of art machine learning [15]								**Model with different combinations**[1]			**Our proposed model Ensemble learning**	
Dataset	**Evaluation metrics**	**LR**	**LDA**	**kNN**	**RF**	**CART**	**NB**	**SVM**	**LSTM**	**(DT-RF-kNN-NB)**	**(DT-RF-NB)**	**(DT-RF-kNN)**	**XGBoost**	**LightGBM**
Combined ToN IoT Dataset	Accuracy	0.61	0.68	0.84	0.85	0.88	0.62	0.61	0.81	0.87	0.88	0.88	0.99	1.0
	Precision	0.37	0.74	0.85	0.87	0.90	0.63	0.37	0.83	0.90	0.90	0.89	0.99	1.0
	Recall	0.61	0.68	0.84	0.85	0.88	0.62	0.61	0.81	0.88	0.88	0.88	0.99	1.0
	F1-score	0.46	0.62	0.84	0.85	0.88	0.51	0.46	0.80	0.87	0.88	0.88	0.99	1.0

Based on the performance table that presents state-of-the-art machine learning models applied to the Combined ToN IoT Dataset, the following conclusions can be inferred:

1. Accuracy: The models achieve varying levels of accuracy, ranging from 0.61 to 1.0. LightGBM demonstrates the highest accuracy of 1.0, indicating its effectiveness in correctly classifying instances. XGBoost also performs well with an accuracy of 0.99. Logistic Regression (LR) and Naive Bayes (NB) models show relatively lower accuracy compared to others.
2. Precision: Precision measures the ability of a model to correctly identify positive instances. XGBoost and LightGBM models achieve high precision scores of 0.99 and 1.0, respectively. Linear Discriminant Analysis (LDA) and SVM models also exhibit good precision with scores of 0.74 and 0.83, respectively.
3. Recall: Recall reflects the model's ability to correctly identify positive instances out of all true positive instances. XGBoost 0.99, (DT-RF-NB) 0.88, and LightGBM models achieve the highest recall score of 1.0. This indicates their effectiveness in capturing most positive instances. Other models such as LDA, kNN, and RF also demonstrate relatively high recall scores.
4. F1-score: The F1-score provides a balanced measure of a model's precision and recall. XGBoost 0.99, and LightGBM models achieve the highest F1-scores of 1.0, indicating a goods balance between precision and recall. Other models exhibit lower F1-scores, but still demonstrate reasonable performance.

Overall, XGBoost and LightGBM consistently perform well across multiple evaluation metrics, showing their effectiveness in the classification task on the Combined ToN IoT Dataset. These models demonstrate high accuracy, precision, recall, and F1-score, indicating their superiority compared to other state-of-the-art machine learning models for this particular dataset.

V. Conclusion

The dataset ToN IoT dataset, which combines 7 pieces of dataset, normal related to IoT and other network traffic. With features labels that show attack streams, at-tack categories and subcategories for possible binary classification. From the ToN IoT dataset that uses the XGBoost method, the distribution is 80% for training data and 20% for tes Based on the testing results using ensemble techniques with the eX-treme Gradient Boosting (XGBoost) and LightGBM algorithms, it can be concluded that they can be used for classifying the 7 IoT device datasets, namely fridge, garage door, GPS sensor, motion light, Modbus, thermostat, and weather. These algorithms successfully handle both binary classification problems.

In the binary classification testing, the following accuracy results were obtained for each dataset: fridge sensor and garage door achieved 100% accuracy, GPS sensor achieved 92% accuracy, Modbus achieved 98% accuracy, motion light achieved 92% accuracy, thermostat achieved 95% accuracy, and weather achieved 99% ac-curacy.

Furthermore, in the testing of a combined dataset where the seven IoT datasets were merged, XGBoost achieved an accuracy 99%, while LightGBM achieved an accuracy 100%. Comparing these results with previous research, it can be concluded that the ensemble learning algorithms are capable of producing better and optimal accuracy compared to other algorithms that have been previously studied.

Overall, the ensemble learning algorithms, namely XGBoost and LightGBM, demonstrate their effectiveness in classifying the IoT datasets, achieving high accu-racy rates in both binary classification scenarios. These algorithms outperform other previously researched algorithms in terms of accuracy and performance.

References

[1] M. A. Khan *et al.*, "Voting Classifier-Based Intrusion Detection for IoT Networks," pp. 313–328, 2022, https://doi.org/10.1007/978-981-16-5559-3_26.

[2] A. Azmoodeh, A. Dehghantanha, and K. K. R. Choo, "Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigenspace Learning," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 88–95, 2019, https://doi.org/10.1109/TSUSC.2018.2809665

[3] M. M. Islam, A. Rahaman, and M. R. Islam, "Development of Smart Healthcare Monitoring System in IoT Environment," *SN Comput. Sci.*, vol. 1, no. 3, pp. 1–11, 2020, https://doi.org/10.1007/s42979-020-00195-y

[4] G. Mois, S. Folea, and T. Sanislav, "Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2056–2064, 2017, https://doi.org/10.1109/TIM.2017.2677619

[5] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018, https://doi.org/10.1109/TII.2018.2852491.

[6] L. Nie *et al.*, "Intrusion Detection in Green Internet of Things: A Deep Deterministic Policy Gradient-Based Algorithm," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 778–788, 2021, https://doi.org/10.1109/TGCN.2021.3073714.

[7] G. Falco, C. Caldera, and H. Shrobe, "IIoT Cybersecurity Risk Modeling for SCADA Systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4486–4495, 2018, https://doi.org/10.1109/JIOT.2018.2822842

[8] E. Farzadnia, H. Shirazi, and A. Nowroozi, "A novel sophisticated hybrid method for intrusion detection using the artificial immune system," *J. Inf. Secur. Appl.*, vol. 58, no. February, p. 102721, 2021, https://doi.org/10.1016/j.jisa.2020.102721

[9] A. H. Azizan *et al.*, "A machine learning approach for improving the performance of network intrusion detection systems," *Ann. Emerg. Technol. Comput.*, vol. 5, no. Special issue 5, pp. 201–208, 2021, https://doi.org/10.33166/AETiC.2021.05.025

[10] N. Moustafa, M. Ahmed, and S. Ahmed, "Data Analytics-enabled Intrusion Detection: Evaluations of ToN IoT Linux Datasets," in *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*, 2020, pp. 727–735, https://doi.org/10.1109/TrustCom50675.2020.00100

[11] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput. Secur.*, vol. 106, p. 102289, 2021, https://doi.org/10.1016/j.cose.2021.102289

[12] A. Kumar, A. Abdelhadi, and C. Clancy, "Novel anomaly detection and classification schemes for Machine-to-Machine uplink," in *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 2019, pp. 1284–1289, https://doi.org/10.1109/BigData.2018.8622142

[13] B. Cao, C. Li, Y. Song, and X. Fan, "Network Intrusion Detection Technology Based on Convolutional Neural Network and BiGRU," *Comput. Intell. Neurosci.*, vol. 2022, 2022, https://doi.org/10.1155/2022/1942847.

[14] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Comput. Commun.*, vol. 166, pp. 110–124, 2021, https://doi.org/https://doi.org/10.1016/j.comcom.2020.12.003.

[15] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. N. Anwar, "TON-IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, https://doi.org/10.1109/ACCESS.2020.3022862.

[16] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," *IEEE Access*, vol. 9. pp. 142206–142217, 2021, https://doi.org/10.1109/ACCESS.2021.3120626.

[17] N. Mane, A. Verma, and A. Arya, "A Pragmatic Optimal Approach for Detection of Cyber Attacks using Genetic Programming," in *20th IEEE International Symposium on Computational Intelligence and Informatics, CINTI 2020 - Proceedings*, 2020, pp. 71–76, https://doi.org/10.1109/CINTI51262.2020.9305844.

[18] A. O. Alzahrani and M. J. F. Alenazi, “Designing a network intrusion detection system based on machine learning for software defined networks,” *Futur. Internet*, vol. 13, no. 5, 2021, https://doi.org/10.3390/fi13050111.

[19] P. Kumar, G. P. Gupta, and R. Tripathi, “An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks,” *Comput. Commun.*, vol. 166, pp. 110–124, 2021, https://doi.org/10.1016/j.comcom.2020.12.003

[20] A. R. Gad, A. A. Nashat, and T. M. Barkat, “Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset,” *IEEE Access*, vol. 9, no. October, pp. 142206–142217, 2021, https://doi.org/10.1109/ACCESS.2021.3120626

[21] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, “Building an efficient intrusion detection system based on feature selection and ensemble classifier,” *Comput. Networks*, vol. 174, p. 107247, 2020, https://doi.org/10.1016/j.comnet.2020.107247.

[22] H. Jiang, Z. He, G. Ye, and H. Zhang, “Network Intrusion Detection Based on PSO-Xgboost Model,” *IEEE Access*, vol. 8, pp. 58392–58401, 2020, https://doi.org/10.1109/ACCESS.2020.2982418

[23] X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu, “Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning,” *Electron. Commer. Res. Appl.*, vol. 31, pp. 24–39, 2018, https://doi.org/10.1016/j.elerap.2018.08.002

[24] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 785–794, 2016, https://doi.org/10.1145/2939672.2939785

[25] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 3147–3155, 2017.

[26] G. Pietropolli, L. Manzoni, A. Paoletti, and M. Castelli, “Combining Geometric Semantic GP with Gradient-Descent Optimization,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2022, vol. 13223 LNCS, pp. 19–33, https://doi.org/10.1007/978-3-031-02056-8_2.

[27] H. C. Husada and A. S. Paramita, “Analisis Sentimen Pada Maskapai Penerbangan di Platform Twitter Menggunakan Algoritma Support Vector Machine (SVM),” *Teknika*, vol. 10, no. 1, pp. 18–26, 2021, https://doi.org/10.34148/teknika.v10i1.311.

[28] A. Agarwal, P. Sharma, M. Alshehri, A. A. Mohamed, and O. Alfarraj, “Classification model for accuracy and intrusion detection using machine learning approach,” *PeerJ Comput. Sci.*, vol. 7, pp. 1–22, 2021, https://doi.org/10.7717/PEERJ-CS.437.

[29] X. Liu *et al.*, “NADS-RA: Network Anomaly Detection Scheme Based on Feature Representation and Data Augmentation,” *IEEE Access*, vol. 8, pp. 214781–214800, 2020, https://doi.org/10.1109/ACCESS.2020.3040510.

[30] P. Henrique *et al.*, “Impact of Feature Selection Methods on the Classification of DDoS Attacks using XGBoost,” *J. Commun. Inf. Syst.*, vol. 36, no. 1, 2021.

[31] I. Benmessahel, K. Xie, and M. Chellal, “A new evolutionary neural networks based on intrusion detection systems using multiverse optimization,” *Appl. Intell.*, vol. 48, no. 8, pp. 2315–2327, 2018, https://doi.org/10.1007/s10489-017-1085-y