Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 3, July 2024, pp. 543∼556
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i3.3885

A Comparison of Enhanced Ensemble Learning Techniques for Internet of Things Network Attack Detection

Edi Ismanto , Januar Al Amien , Vitriani
Universitas Muhammadiyah Riau, Pekanbaru, Indonesia

Article Info

Article history:

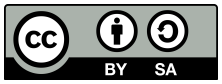
Received February 20, 2024
Revised May 05, 2024
Accepted June 03, 2024

Keywords:

Attack detection
Ensemble learning
Internet of Things network

ABSTRACT

Over the past few decades, the Internet of Things (IoT) has become increasingly significant due to its capacity to enable low-cost devices and sensor communication. Implementation has opened many new opportunities for efficiency, productivity, convenience, and security. However, it has also brought about new privacy and data security challenges, interoperability, and network reliability. The research issue was that IoT devices are frequently open to attacks. Certain machine learning (ML) algorithms still struggle to handle imbalanced data and have weak generalization skills compared to ensemble learning. **The research aimed** to develop security for IoT networks based on enhanced ensemble learning using Grid Search and Random Search techniques. **The method used** was the ensemble learning approach, which consists of Random Forest (RF), Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM), and Extreme Gradient Boosting (XGBoost). This study used the UNSW-NB15 IoT dataset. **The study's findings** demonstrated that XGBoost performs better than other methods at identifying IoT network attacks. By employing Grid Search and Random Search optimization, XGBoost achieves an accuracy rate of 98.56% in binary model measurements and 97.47% on multi-class data. The findings underscored the efficacy of XGBoost in bolstering security within IoT networks.



Corresponding Author:

Edi Ismanto,
Department of Informatics,
Universitas Muhammadiyah Riau, Pekanbaru, Indonesia,
Email: edi.ismanto@umri.ac.id.

How to Cite:
E. Ismanto, J. Al Amien, and V. Vitrian, "A Comparison of Enhanced Ensemble Learning Techniques for Internet of Things Network Attack Detection", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 3, pp. 543-556, July, 2024.

1. INTRODUCTION

The Internet of Things (IoT) enables items to communicate, sense their environment, and share data online. IoT has grown significantly in importance over the past few decades as a result of its ability to facilitate low-cost device and sensor communication [1, 2]. Aside from data connectivity infrastructures, analytics, and people, the IoT is propelled by intelligent assets that exchange and store data [3–5]. These IoT smart systems are susceptible to a variety of cyberattacks, including denial-of-service (DDoS), man-in-the-middle (MiM), backdoor, and infiltration attacks [6, 7]. Attacks like this have the potential to compromise the integrity and confidentiality of data on that network [8]. Therefore, a detection system is a must in any effective security solution to keep them safe from these kinds of attacks. One security tool for monitoring data traffic is an intrusion detection system (IDS) [9, 10]. It protects the network by acting as a second line of defense. IDS monitors the networks at every point of entry, looks for any unauthorized activity in the packets moving through the channel, and notifies the appropriate authority if it finds any [11–13]. IDS is typically installed after a firewall, which seems to be its optimal placement. To detect known attacks, they compare the data packet with a predefined list or database containing the pattern or signatures of known attacks [14, 15, 15]. IDS offers useful safeguards to protect IoT from multiple outbreaks [16, 17].

A literature review was undertaken to provide an in-depth understanding of using Machine Learning (ML) techniques in IoT network attack detection. This paper proposes a One-Class Bidirectional GRU Autoencoder and Ensemble Learning-Based Lightweight Intelligent NIDS. In addition to correctly classifying data as normal or abnormal, it can also classify unknown attacks as the kind that most closely resemble known attacks [18]. For an intrusion detection system using the Information-Centric Networking (ICN) dataset, this paper presents a Gradient Boosting Decision Tree (GBDT)-paralleled quadratic ensemble learning method. The results demonstrate that the method achieves significantly better performance when compared with existing methods [19]. Using the LightGBM classifier, propose a dynamic ensemble algorithm for anomaly detection in IoT environments [20, 21]. The ensemble has been constructed using three extensively used models: Artificial Neural Networks (ANN), Random Forests (RF), and Decision Trees (DT), based on all the taken into consideration metrics, the system can detect nine different categories of attacks with high performance, the experimental evaluation carried out on the CIC-IDS2017 public dataset [13, 22, 23]. Lightweight ML models, such as decision trees (DT), logistic regression (LR), and Gaussian naive Bayes(GNB) as the base classifier and stochastic gradient descent (SGD) as the meta-classifier, are used to build this ensemble model [21]. Three datasetsKDD Cup 1999, UNSW-NB15, and CIC-IDS 2017are used to train and assess the performance of this proposed model and the individual classifiers that went into building the ensemble model. The empirical results unequivocally demonstrate the great benefits of using an ensemble classifier in intrusion detection systems with unbalanced datasets [23]. Stacked ensemble technology is being used to create improved IDS that safeguard information networks [4, 24–26]. Four base ML algorithms, namely K-Nearest Neighbour (KNN), Nave Bayes (NB), Support Vector Machine (SVM), and Decision Tree (DT), were trained on pertinent features from the UNSW-15NB intrusion detection dataset to construct base-predictive models [6, 9, 27, 28]. 3.0% more accurate classifications were made thanks to the stacked ensemble recording [27]. To classify the network flow as normal or anomalous, the suggested ML pipeline combines the bagging and boosting algorithms through voting between an RF classifier and an XGBoost classifier [7, 10, 29–32]. The UNSW-NB15, NSL-KDD, and BoTIoT benchmark datasets are used to train the proposed model; the accuracy of the results is extremely high [2, 11, 33–35]. Shallow learning techniques like extreme gradient boosting (XGBoost), K-nearest Neighbours (KNN), and Random forest (RF) are used to further train and validate the Message queuing telemetry transport (MQTT) dataset [5, 12]. Better attack detection rates are achieved in experimental results compared to the control group. The dual ensemble model presented in this work combines gradient-boosting decision trees (GBDT) and bagging, two popular ensemble techniques. The evaluation of GBDT algorithms, including LightGBM, CatBoost, XGBoost, J48 Decision Tree, and Gradient Boosting Machine (GBM), is conducted on several publicly accessible data sets, including NSL-KDD, UNSW-NB15, and HIKARI-2021 [35–37]. According to the results, the suggested method appears to be a workable answer for the anomaly-based IDS task.

A gap in the literature that has not yet been addressed is the overfitting problem that arises when complex models are applied to small or unbalanced data sets. Ensemble learning can help reduce overfitting problems when complex models are used on small or unbalanced data sets. Nevertheless, selecting the best model to add to an ensemble and modifying the parameters to achieve optimal performance can be challenging. In this work, we suggest enhancing an ensemble model's performance to detect attacks on IoT networks. **This study differs from others** in that it optimizes the ensemble learning model's parameters to increase the accuracy of attack detection by using Grid Search and Random Search optimization techniques.

One issue with this research is that because of the architecture's widespread use and the insufficient application of technological security, IoT devices are susceptible to different kinds of cyberattacks. Conventional Machine learning (ML) models still tend to perform less well in generalization. In general, conventional models are less adaptable when it comes to managing abrupt shifts or data drifts. This has an impact on low detection results. Determining an appropriate method to safeguard these networks against cyberattacks is desirable to improve the dependability of these systems [18]. **The goal of this** work is to create and enhance machine

learning ensemble models to identify IoT network device attacks. This study used four ensemble models: eXtreme Gradient Boosting (XGBoost), Gradient Boosting Machine (GBM), Adaptive Boosting (AdaBoost), and Random Forest (RF). The Grid Search and Random Search hyperparameter optimization techniques are used to raise the ensemble model's performance.

To address the aforementioned issues, we have presented a cyberattack detection framework combining an enhanced ensemble learning model with a Grid Search and Random Search optimization algorithm. The ensemble design makes use of four popular and effective ML techniques: RF, GBM, XGBoost, and AdaBoost [38, 31, 35]. **The primary contribution** of the work lies in developing an ensemble learning model-based framework tailored for IoT cyberattack detection. This framework incorporates the latest classifier models like XGBoost, AdaBoost, GBM, and RF, which are utilized to construct the ensemble learning model. The approach also focuses on improving attack detection accuracy by leveraging Grid Search and Random Search algorithms to optimize the model parameters, thereby enhancing the overall efficacy of the detection system.

2. RESEARCH METHOD

In this study, experimental methods are employed. The suggested method for attack detection in IoT environments is presented here in block diagram form, utilizing an improved ensemble learning technique. Several functional units make up the suggested architecture. Figure 1 displays the employed research methodology.

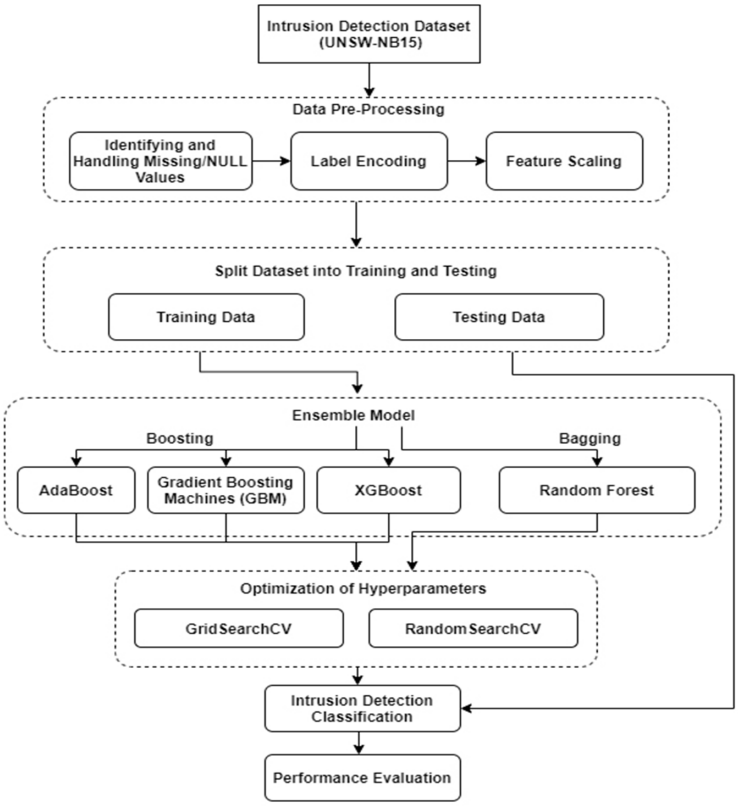


Figure 1. Methods of research

The UNSW-NB15 intrusion detection dataset underwent preprocessing to prepare it for the creation of ensemble learning intrusion detection models. Feature scaling, label coding, and handling of missing values were applied to the data set. After that, the data is separated into training and testing sets. After that, the data is trained and tested using the ensemble learning model. Grid Search and Random Search techniques were used to improve the performance of the ensemble learning process and optimize the model hyperparameters.

2.1. Datasets

The UNSW-NB 15 IoT dataset was the source used in this study. UNSW-NB 15 is the most current intrusion detection research dataset made available by the Australian Centre for Cyber Security (ACCS) Cyber Range Laboratory. The UNSW-NB15 dataset has the following advantages over NSLKDD [32]: First, it includes contemporary synthesized attack activities and actual modern normal behaviors. Secondly, there is a similarity in the training and testing sets' probability distributions. Thirdly, it employs a collection of characteristics from the packet header and payload to represent the network packets effectively. The analysis of the UNSW-NB15 on existing classification systems showed complex patterns in the dataset, suggesting that it can be used to evaluate both novel and well-established classification methods with precision and effectiveness [33, 21, 39, 27, 28]. Table 1 displays the distribution and description of its nine attack categories and normal connections. The training and testing sets consist of 82,332 and 175,341 records, respectively.

Table 1. An Explanation of the Various Attack Methods Found in the UNSW-NB15 Dataset

Type	No of Records	Description
Normal	93	Innate transaction information
Analysis	2.674	This is an intrusion attack against a web application using port-based penetration.
Backdoors	2.329	This is an attempt to access a system without authorization through a penetration test.
DoS	16.353	This attack strips authorized network users of their entitlement to the memory and storage space that the system requires to carry out computation tasks.
Exploits	44.525	This is a type of penetration attack where an operating system glitch, bug, or vulnerability is exploited using a series of instructions or code.
Fuzzers	24.244	This is a scan attack that uses software testing techniques to systematically bombard the victim's system with multiple requests until it crashes to find security holes and program flaws in the operating system, network, or program.
Generic	58.871	This is a penetration attack technique that doesn't take the block cipher's structure into account and can be used against any block cipher.
Reconnaissance	13.987	This is a type of probe attack that collects data about a network's configuration to circumvent security measures.
Shellcode	1.511	This is a penetration attack that compromises the victim's computer by using a small program that gets its instructions from a shell.
Worms	174	This is a self-replicating malicious code scan attack that propagated to other computers, primarily via a computer network, without joining itself to a program like a virus.

This data set has nine distinct attack categories, with normal denoting non-attack. The attack data categories that appear most frequently in data are "Generic," "Exploits," "Fuzzers," "DoS," and "Reconnaissance". The distribution of attacks that took place is depicted in Figure 2.

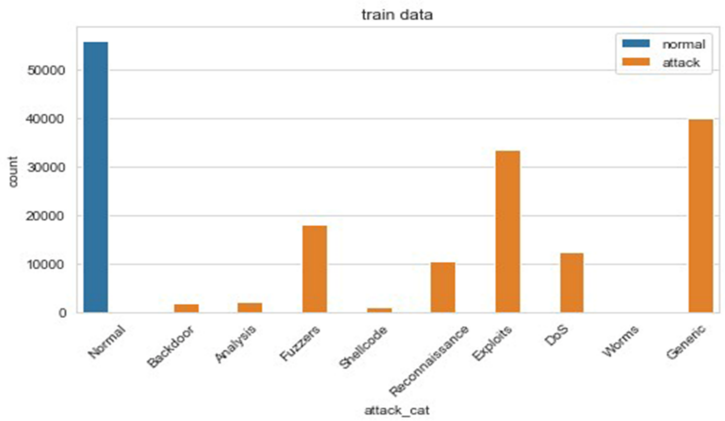


Figure 2. The distribution of attacks

2.2. Pre-Processing

Following the dataset's selection, data-cleaning procedures are carried out to normalize the features and remove noise [15]. Several approaches are used for normalization; however, the min-max normalization approach is employed in this study because it

is more effective in terms of scaling and resolving problems with z-score normalization related to outliers [40]. Values normalized within the interval [0, 1] by min-max scaling. Equation (1) can be used to calculate min-max normalization [41].

$$Z_i = \frac{Y_i - min(Y)}{max(Y) - min(Y)} \tag{1}$$

The number of features is denoted by Y = (Y1, Y2, Y3, ..., Yn), the feature we wish to normalize is Yi, and the normalized feature is Zi. As a result, all features are now within the same scope and have equal weights.

Before the label encoding process was initiated, duplicate and inconsistent values were eliminated from the datasets [26]. The nominal attributes had to be changed to numerical values in the following step. This is because the back-end computations of ensemble learning algorithms use numerical values rather than nominal values. We must first complete this data encoding step to pass data to the suggested model. The two groups of the dataset consist of the binary class labels and multi-class labels sections. For Binary Classification, a duplicate of DataFrame is made. There are two categories for the "label" attribute: "normal" and "abnormal." Labels are encoded using LabelEncoder() and stored in the binary dataset 'label', which consists of 81173 rows and 61 columns. For multi-class classification, a copy of the DataFrame is made. The nine categories that make up the 'attack_cat' attribute are 'Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Normal, Reconnaissance, and Worms.' Attack_cat is encoded using LabelEncoder(), and the encoded labels are saved in a label. Multi-class dataset with 69 columns and 81173 rows.

2.3. Ensemble Model

This paper aims to improve the attack classification recognition rate using an Ensemble Learning model that uses Random Forest (RF) as the bagging model and boosting models like the AdaBoost algorithm, Gradient Boosting Machines (GBM), and XGBoost.

2.4. The Random Forest (RF) Algorithm

The Random Forest (RF) algorithm is an ensemble learning method used in ML for classification and regression. To generate more precise and reliable predictions, RF constructs several decision trees during training and merges the predictions from these trees [25, 42]. RF uses bootstrap sampling on the training set of data before constructing each tree [10]. Bootstrap sampling is the process of selecting random samples from training data and replacing them [11]. This can lead to some samples appearing more than once and others not appearing at all.

2.5. The Adaptive Boosting (AdaBoost) Algorithm

The AdaBoost algorithm is an ensemble learning technique that combines multiple weak models (weak learners) into a single strong model (strong learners) [42, 30]. It is particularly useful for classification [11]. AdaBoost has the advantage of being able to work well with complex data and handle binary and multi-class classification problems [25].

2.6. The Gradient Boosting Machines (GBM) Algorithm

Gradient boosting machines (GBM) are popular ML algorithms that work well for regression and classification problems. GBM is an ensemble method that focuses on data points that were not thoroughly examined in the past to correct earlier model errors iteratively [37]. Among GBM's benefits are its capacity to work with high-dimensional data, handle a broad range of feature kinds, and typically generate very predictive models [31].

2.7. The eXtreme Gradient Boosting (XGBoost) Algorithm

XGBoost has emerged as one of the most well-liked and successful algorithms in ML competitions. It is also extensively utilized in the industry for a range of prediction tasks, such as ranking, regression, and classification [10, 11]. XGBoost offers a strong regularisation method to lessen overfitting. With XGBoost, missing data is automatically handled at any split point in the tree, allowing for accurate prediction even when missing data is present. XGBoost makes use of memory distribution and parallelism to optimize performance.

2.8. Optimizing Hyperparameters in Models

Hyperparameter model optimization is the process of enhancing the hyperparameter values of an Ensemble Learning model [43]. Hyperparameters are variables that impact the model's performance and complexity and are not altered during the training process [37]. In this study, Grid Search and Random Search were the two hyperparameter optimization techniques employed. This method involves methodically searching through various combinations of predefined hyperparameters. Grid search will assess every possible combination of hyperparameters and choose the combination that yields the best results [44, 45]. The grid search mathematical notation IN Equation (2) [46].

$$Grid = R_1 * R_2 * \ldots * R_n \tag{2}$$

R is the range of values that are possible for every hyperparameter. As a result, the set of all potential pairings of hyperparameter values for evaluation will be generated. The function of evaluation f is used to evaluate each hyperparameter combination and choose the set of hyperparameters that yields the best outcomes according to the given evaluation standards. This approach chooses a set of hyperparameters at random for evaluation. Random search is defined as a stochastic search algorithm that selects randomly from a set of hyperparameter values and evaluates the objective function for each randomly selected combination [47, 48]. The random search is expressed mathematically in Equation (3) [49].

$$i = 1, 2, \ldots, N \tag{3}$$

To begin the random search, choose how many N iterations to run. From the set H, choose one hyperparameter combination at random. Evaluation of the function of f in the selected hyperparameter combination. Based on the given evaluation criteria, choose the hyperparameter combination that yields the best results.

2.9. Evaluation of Performance

Evaluating use model performance is the central component of any experimental study. For this reason, the standard evaluation metrics that are the focus of this study are accuracy, recall, precision, and f1-score [10, 35, 36]. Equations (4), (5), (6), and (7) can be used to calculate accuracy, recall, precision, and F1-Score, respectively [50].

$$Accuracy(Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Accuracy (Acc) is a variable used to calculate accuracy. TP (True Positive) is the number of positive cases that are correctly identified by the model as positive. TN (True Negative) is the number of negative cases that are correctly identified by the model as negative. FP (False Positive) is the number of negative cases falsely detected by the model as positive. FN (False Negative) is the number of positive cases incorrectly detected by the model as negative.

$$Recall = \frac{TP}{(TP + FN)} \tag{5}$$

Recall is used to measure how well a model detects positive cases. TP (True Positive) is the number of positive cases that are correctly detected by the model as positive. FN (False Negative) is the number of positive cases incorrectly detected by the model as negative.

$$Precision(Prec) = \frac{TP}{(TP + FP)} \tag{6}$$

The Precision (Prec) variable is used to measure how well a model detects true positive cases. TP (True Positive) is the number of positive cases that are correctly detected by the model as positive. FP (False Positive) is the number of negative cases falsely detected by the model as positive.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

F1-Score measures the detection model's performance by considering both precision (accuracy) and recall (sensitivity). Precision is the ratio of the number of positive cases correctly detected by the model (TP) to the total cases detected as positive by the model (TP + FP). Recall (Sensitivity) is the ratio of the number of positive cases correctly detected by the model (TP) to the total positive cases that should have been detected by the model (TP + FN).

3. RESULT AND ANALYSIS

3.1. Environment of Experimentation

Research experiments were carried out using Python programming. With an 11th-generation Intel core i7 processor operating at 3.00GHz and 64 GB of RAM. The actual UNSW-NB15 dataset, which includes recent attacks and non-malicious data, is used to evaluate the efficacy of this IoT attack detection system. In random order, the binary data will be divided into two categories: training (80%) and testing (20%). Similarly, 80% of the multi-class data is set aside for training and 20% for testing. The parameters of the ensemble learning model are optimized with an estimator of (50, 100, and 150) and a learning rate of (0.1, 0.5, and 1.0).

3.2. Analysis of Experimental Results

This subsection evaluates a proposed attack detection system that guards against cyberattacks in IoT environments by employing an ensemble learning technique. The ensemble learning models, RF, AdaBoost, GBM, and XGBoost, were trained and tested in the first phase without parameter optimization. The model measurement results in the binary class without optimization are shown in Table 2.

Table 2. The Binary Class Model Measurement Results Without Optimization

Models	Acc (%)	Precision	Recall	F1-Score
RF	98.34	0.98	0.97	0.98
AdaBoost	98.35	0.98	0.98	0.98
GBM	98.32	0.98	0.98	0.98
XGBoost	**98.54**	0.99	0.99	0.99

Experiments on four ensemble models in the binary class without optimization show that the XGBoost model outperforms the other models in terms of accuracy. The accuracy percentage attained by the XGBoost model was 98.54%. The binary class's attack detection classification outcomes for the XGBoost model are shown in Figure 3.

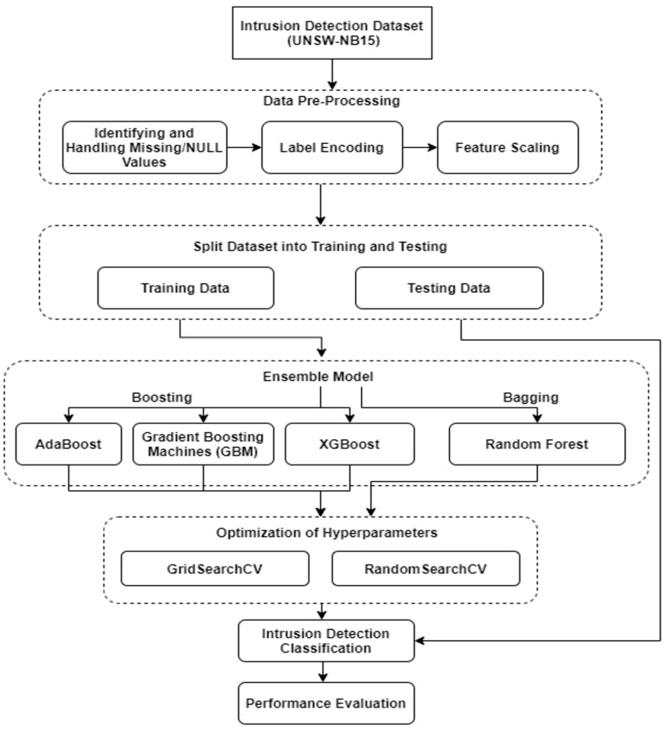


Figure 3. The XGBoost model classified attacks in the binary class.

In Figure 3, the number of cases that truly belong to the ”abnormal” class and were correctly detected by the XGBoost model is 3770. The cases that should have been ”abnormal” but were detected as ”normal” by the XGBoost model amount to 139. The cases of ”normal” that were correctly detected by the XGBoost model total 12229. Meanwhile, the cases that should have been ”normal” but were detected as ”abnormal” by the XGBoost model are 97. Testing of ensemble models on multiclass data was also done without optimization. Table 3 displays the model measurement results for the multi-class without optimization. Four ensemble models were used for evaluation.

Table 3. The Multi-Class Model Measurement Results Without Optimization

Models	Acc (%)	Precision	Recall	F1-Score
RF	97.31	0.97	0.97	0.97
AdaBoost	97.17	0.96	0.97	0.96
GBM	97.49	0.97	0.97	0.97
XGBoost	**97.51**	0.97	0.98	0.97

When dealing with multi-class data, the XGBoost model continues to outperform other models. When compared to other models, the multi-class XGBoost accuracy of 97.51% was the highest. The results of the XGBoost model's attack detection classification on multi-class data are shown in Figure 4.

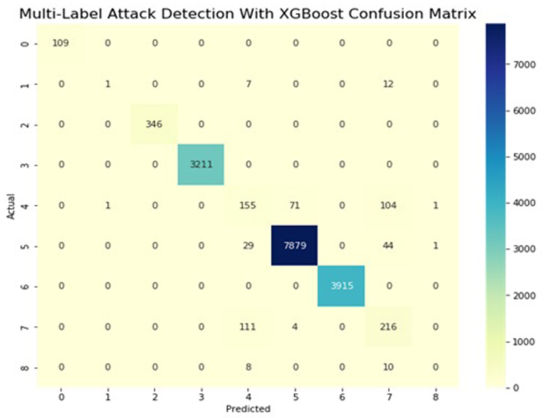


Figure 4. The Classification Outcomes of the XGBoost Model's Attack Detection on Multi-Class Data

Based on Figure 4 for multi-class data, the XGBoost model correctly detects 109 instances as 'Analysis.' It detects 1 instance as 'Backdoor'. This model accurately identifies 346 instances as 'DoS'. It correctly detects 3211 instances as 'Exploits'. The model detects 71 as 'Fuzzers' and 104 as 'Normal,' whereas they are actually 'Fuzzers.' This model identifies 7879 as 'Generic,' 44 as 'Reconnaissance,' and 1 as 'Worms,' whereas they are actually 'Generic.' The model accurately detects 3915 instances as 'Normal'. It predicts 111 as 'Reconnaissance,' 4 as 'Generic,' and 216 as 'Worms,' whereas they are actually 'Reconnaissance.' Additionally, the model detects 8 as 'Generic' and 10 as 'Reconnaissance,' whereas they are actually 'Worms.' Test results on binary and multi-class classes form the basis for improving the model's performance. Subsequently, four ensemble models were optimized on binary and multi-class data using Grid Search and Random Search methods. The model measurement results for the binary with optimization Grid Search are shown in Table 4.

Table 4. The Binary Model Measurement Results with Optimization Grid Search

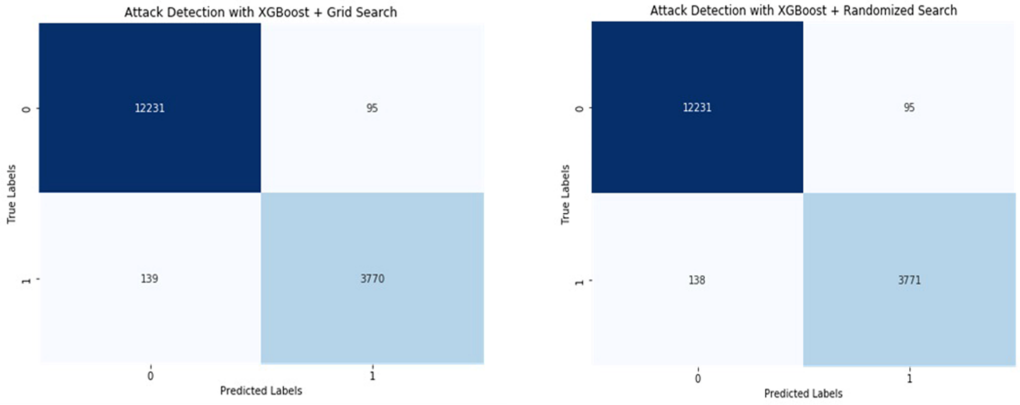
Models	Acc (%)	Precision	Recall	F1-Score
RF + Grid Search	98.33	0.98	0.98	0.98
AdaBoost + Grid Search	98.36	0.98	0.98	0.98
GBM + Grid Search	98.47	0.98	0.98	0.98
XGBoost + Grid Search	**98.56**	0.98	0.98	0.98

XGBoost accuracy still yields the best results based on the binary model measurement results with Grid Search optimization for the four ensemble models. After optimization with Grid Search and Random Search, XGBoost yielded an accuracy value of

98.56%. The model measurement results for the binary with optimization Random Search are shown in Table 5. The results of the XGBoost model's attack detection classification in the binary class using Grid Search and Random Search optimization are shown in Figure 5.

Table 5. The Binary Model Measurement Results with Optimization Random Search

Models	Acc (%)	Precision	Recall	F1-Score
RF + Random Search	98.3	0.98	0.98	0.98
AdaBoost + Random Search	98.29	0.98	0.98	0.98
GBM + Random Search	98.46	0.98	0.98	0.98
XGBoost + Random Search	**98.56**	0.98	0.98	0.98



(a) (b)

Figure 5. The XGBoost Model uses Grid Search and Random Search Optimization to Classify Attacks in the binary class

In Figure 5, the XGBoost detection results for binary classes with Grid Search optimization show that there were 12231 instances correctly detected as the normal class (0) by the model. The number of instances incorrectly detected as the abnormal class (1) by the model is 95. Additionally, there were 139 instances incorrectly detected as the normal class (0) by the model, while the number of instances correctly detected as the abnormal class (1) by the model is 3770. Similarly, for the XGBoost detection results for binary classes with Random Search optimization, there were 12231 instances correctly detected as the normal class (0) by the model. The number of instances incorrectly detected as the abnormal class (1) by the model is 95. Moreover, there were 138 instances incorrectly detected as the normal class (0) by the model, whereas the number of instances correctly detected as the abnormal class (1) by the model is 3771.

Additionally, ensemble model testing was done on multi-class data using Grid Search and Random Search for optimization. The model measurement results for the multi-class with optimization Grid Search is shown in Table 6.

Table 6. The Multi-Class Model Measurement Results with Optimization Grid Search

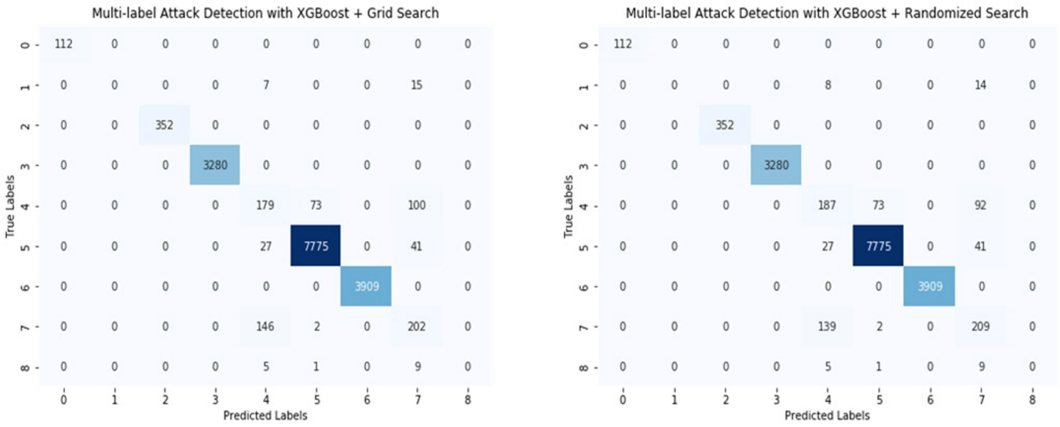
Models	Acc (%)	Precision	Recall	F1-Score
RF + Grid Search	97.46	0.97	0.97	0.97
AdaBoost + Grid Search	97.35	0.97	0.97	0.97
GBM + Grid Search	97.43	0.97	0.97	0.97
XGBoost + Grid Search	**97.47**	0.97	0.97	0.97

The best accuracy value was found for the XGBoost model based on the outcomes of testing the ensemble model using Grid Search optimization on multi-class data. Using Random Search optimization, ensemble model testing was also performed on multi-class data, and the best model was determined to be XGBoost. A 97.47% accuracy was attained by both. The model measurement results for the multi-class with optimization Random Search are shown in Table 7.

Table 7. The Multi-Class Model Measurement Results with Optimization Random Search

Models	Acc (%)	Precision	Recall	F1-Score
RF + Random Search	97.39	0.97	0.97	0.97
AdaBoost + Random Search	97.35	0.97	0.97	0.97
GBM + Random Search	97.38	0.97	0.97	0.97
XGBoost + Random Search	**97.47**	0.97	0.97	0.97

The results of the XGBoost model's attack detection classification using Grid Search and Random Search optimization on multi-class data are shown in Figure 6. In terms of classification results for attack detection in IoT networks, the XGBoost model outperforms the RF, AdaBoost, and GBM models. The experimental results indicate that optimization techniques such as Grid Search and Random Search significantly impact model performance.



(a) (b)

Figure 6. The Classification Results of the XGBoost Model's Attack Detection using Grid Search and Random Search Optimization on Multi-Class Data

In Figure 6, the XGBoost detection results for multi-class with Grid Search optimization show several instances correctly classified into each class, such as 112 instances in the 'Analysis' class (True Positive for class 0), 352 instances in the 'DoS' class (True Positive for class 2), 3280 instances in the 'Exploits' class (True Positive for class 3), 179 instances in the 'Fuzzers' class (True Positive for class 4), 7775 instances in the 'Generic' class (True Positive for class 5), 3909 instances in the 'Normal' class (True Positive for class 6), 202 instances in the 'Reconnaissance' class (True Positive for class 7), and 9 instances in the 'Worms' class (True Positive for class 8). However, some instances were misclassified, such as 73 instances misclassified as 'Generic' and 100 instances misclassified as 'Reconnaissance' in the 'Fuzzers' class (False Positive for class 4), as well as other cases like 27 instances misclassified as 'Fuzzers' and 41 instances misclassified as 'Reconnaissance' in the 'Generic' class (False Positive for class 5). In the XGBoost detection results for multi-class with Random Search optimization, several classes were correctly identified by the model, such as 'Analysis' with 112 True Positives, 'DoS' with 352 True Positives, 'Exploits' with 3280 True Positives, 'Fuzzers' with 187 True Positives, 'Generic' with 7775 True Positives, 'Normal' with 3909 True Positives, 'Reconnaissance' with 209 True Positives, and 'Worms' with 9 True Positives. However, there were also detection errors, such as 73 False Positives for the 'Generic' class and 92 False Positives for the 'Reconnaissance' class from the 'Fuzzers' class, as well as 27 False Positives for the 'Fuzzers' class and 41 False Positives for the 'Reconnaissance' class from the 'Generic' class.

3.3. Discussion

The study's findings indicate that based on binary model measurements with Grid Search and Random Search optimization, XGBoost accuracy produces the best results compared to the three ensemble models: RF, AdaBoost, and GBM. XGBoost produced an accuracy result of 98.56% following optimization using Grid Search and Random Search. Similarly, XGBoost performs better than other models based on the results of the ensemble model testing that uses Grid Search and Random Search optimization on

multi-class data. XGBoost achieves a 97.47% accuracy rate on multi-class data. The findings of this study are consistent with the research [4, 25], which found that the ensemble learning approach effectively identifies IoT network attacks. Comparing its accuracy to research findings [28] using an ML algorithm, it was only 89%. However, the recommended method's accuracy levels in this study were 98.56% and 97.47%. Compared to the results of a study [35] that used an ensemble algorithm, and it only attained an accuracy of 83.73%. Meanwhile, the method this study recommended produced accuracy results of 98.56% and 97.47%.

In line with Research [10], the XGBoost algorithm excels in handling complex and large-scale data due to its ability to address overfitting and underfitting issues. This is achieved through regularization techniques applied during the boosting process, such as incorporating penalty terms in the objective function and pruning the resulting decision trees. Furthermore, optimization techniques like Grid Search and Random Search also play a crucial role in enhancing the overall performance. By adjusting model parameters for better detection accuracy, these methods significantly improve the algorithm's ability to handle a variety of datasets.

4. CONCLUSION

The results of this research are that XGBoost accuracy provides the best results based on binary model measurements with Grid Search and Random Search optimization when compared to the three ensemble models, namely RF, AdaBoost, and GBM. After optimization with Grid Search and Random Search, XGBoost yielded an accuracy value of 98.56%. Similarly, according to the ensemble model testing results utilizing Grid Search and Random Search optimization on multi-class data, XGBoost demonstrates superior performance compared to other models. The accuracy attained by XGBoost on multi-class data is 97.47%. This research contributes to developing security in IoT networks based on enhanced ensemble learning utilizing Grid Search and Random Search approaches. The underperformance of ensemble models in previous research can be attributed to a lack of studies on improving their performance. Based on the experimental results of four ensemble learning models, namely RF, AdaBoost, GBM, and XGBoost, it can be concluded that the XGBoost model exhibits the best performance in detecting attacks on IoT networks. This research provides insights into the effectiveness of enhanced ensemble learning techniques in detecting IoT network attacks while highlighting the importance of continuously developing appropriate security strategies to protect the increasingly interconnected network infrastructure associated with IoT devices. Future research must test with a dataset that contains a higher number of variants of attack types and other algorithms that were not tested in this study.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

Three writers, assigned to different tasks, collaborated to compile this research. Edi Ismanto did data gathering, analysis, and interpretation. January Al Amien oversaw the research's conception and design, critically evaluated the article, and gave the go-ahead for publication. Vitriani was in charge of preparing the articles.

FUNDING STATEMENT

This study was granted a Muhammadiyah Batch VII National Research Grant 2024 under contract number 0258.509/I.3/D/2024.

COMPETING INTEREST

I have nothing to disclose about competing institutional, general, and financial interests.

REFERENCES

[1] B. Lal, S. Ravichandran, R. Kavin, N. Anil Kumar, D. Bordoloi, and R. Ganesh Kumar, "IOT-BASED cyber security identification model through machine learning technique," *Measurement: Sensors*, vol. 27, no. December 2022, p. 100791, 2023, https://doi.org/10.1016/j.measen.2023.100791.

[2] A. Alhowaide, I. Alsmadi, and J. Tang, "Ensemble Detection Model for IoT IDS," *Internet of Things (Netherlands)*, vol. 16, no. June 2021, p. 100435, 2021, https://doi.org/10.1016/j.iot.2021.100435.

[3] R. A. Yunmar, "Hybrid Intrusion Detection System using Fuzzy Logic Inference Engine for SQL Injection Attack," *Kursor*, vol. 9, no. 3, pp. 83–94, 2018, https://doi.org/10.28961/kursor.v9i3.147.

[4] R. Lazzarini, H. Tianfield, and V. Charissis, "A stacking ensemble of deep learning models for IoT intrusion detection," *Knowledge-Based Systems*, vol. 279, p. 110941, 2023, https://doi.org/10.1016/j.knosys.2023.110941.

[5] J. B. Awotunde, S. O. Folorunso, A. L. Imoize, J. O. Odunuga, C. C. Lee, C. T. Li, and D. T. Do, "An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks," *Applied Sciences (Switzerland)*, vol. 13, no. 4, 2023, https://doi.org/10.3390/app13042479.

[6] P. Bajaj, S. Mishra, and A. Paul, "Comparative Analysis of Stack-Ensemble-Based Intrusion Detection System for Single-Layer and Cross-layer DoS Attack Detection in IoT," *SN Computer Science*, vol. 4, no. 5, 2023, https://doi.org/10.1007/s42979-023-02105-4.

[7] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Computer Communications*, vol. 166, no. October 2020, pp. 110–124, 2021, https://doi.org/10.1016/j.comcom.2020.12.003.

[8] A. Arqane, O. Boutkhoum, H. Boukhriss, and A. EL Moutaouakkil, "Intrusion Detection System using Ensemble Learning Approaches: A Systematic Literature Review," *International journal of online and biomedical engineering*, vol. 18, no. 13, pp. 160–175, 2022, https://doi.org/10.3991/ijoe.v18i13.33519.

[9] K. Johnson Singh, D. Maisnam, and U. S. Chanu, "Intrusion Detection System with SVM and Ensemble Learning Algorithms," *SN Computer Science*, vol. 4, no. 5, 2023, https://doi.org/10.1007/s42979-023-01954-3.

[10] A. Parashar, K. S. Saggu, and A. Garg, "Machine learning based framework for network intrusion detection system using stacking ensemble technique," *Indian Journal of Engineering and Materials Sciences*, vol. 29, no. 4, pp. 509–518, 2022, https://doi.org/10.56042/ijems.v29i4.46838.

[11] S. A. Hussein, A. A. Mahmood, and E. O. Oraby, "Network Intrusion Detection System Using Ensemble Learning Approaches," *Webology*, vol. 18, no. Special Issue, pp. 962–974, 2021, https://doi.org/10.14704/WEB/V18SI05/WEB18274.

[12] S. Hariprasad and T. Deepa, "An Ensemble Intrusion Detection System based on Acute Feature Selection," *Multimedia Tools and Applications*, pp. 8267–8280, 2023, https://doi.org/10.1007/s11042-023-15788-x.

[13] 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, no. March, 2020, https://doi.org/10.1016/j.comnet.2020.107247.

[14] R. Bingu and S. Jothilakshmi, "Design of Intrusion Detection System using Ensemble Learning Technique in Cloud Computing Environment," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, pp. 751–764, 2023, https://doi.org/10.14569/IJACSA.2023.0140580.

[15] M. Nobakht, R. Javidan, and A. Pourebrahimi, "DEMD-IoT: a deep ensemble model for IoT malware detection using CNNs and network traffic," *Evolving Systems*, vol. 14, no. 3, pp. 461–477, 2023, https://doi.org/10.1007/s12530-022-09471-z.

[16] S. Raut, A. Poojary, A. Naiknaware, S. Vairat, and S. R. Khonde, "Ensemble Based Intrusion Detection System for Multi attack Environment," vol. 6, no. 11, pp. 687–690, 2020.

[17] R. Ahmad, I. Alsmadi, W. Alhamdani, and L. Tawalbeh, "A Deep Learning Ensemble Approach to Detecting Unknown Network Attacks," *Journal of Information Security and Applications*, vol. 67, p. 103196, 2022, https://doi.org/10.1016/j.jisa.2022.103196.

[18] W. Yao, L. Hu, Y. Hou, and X. Li, "A Lightweight Intelligent Network Intrusion Detection System Using One-Class Autoencoder and Ensemble Learning for IoT," *Sensors*, vol. 23, no. 8, pp. 1–25, 2023, https://doi.org/10.3390/s23084141.

[19] J. Yang, Y. Sheng, and J. Wang, "A GBDT-paralleled quadratic ensemble learning for intrusion detection system," *IEEE Access*, vol. 8, pp. 175 467–175 482, 2020, https://doi.org/10.1109/ACCESS.2020.3026044.

[20] A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, and J. Ahmad, "A New Ensemble-Based Intrusion Detection System for Internet of Things," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 1805–1819, 2022, https://doi.org/10.1007/s13369-021-06086-5.

[21] N. Thockchom, M. M. Singh, and U. Nandi, "A novel ensemble learning-based model for network intrusion detection," *Complex and Intelligent Systems*, vol. 9, no. 5, pp. 5693–5714, 2023, https://doi.org/10.1007/s40747-023-01013-7.

[22] J. Jiang, F. Liu, Y. Liu, Q. Tang, B. Wang, G. Zhong, and W. Wang, "A dynamic ensemble algorithm for anomaly detection in IoT imbalanced data streams," *Computer Communications*, vol. 194, no. April, pp. 250–257, 2022, https://doi.org/10.1016/j.comcom.2022.07.034.

[23] A. Arshad, M. Jabeen, S. Ubaid, A. Raza, L. Abualigah, K. Aldiabat, and H. Jia, "A novel ensemble method for enhancing Internet of Things device security against botnet attacks," *Decision Analytics Journal*, vol. 8, no. June, p. 100307, 2023, https://doi.org/10.1016/j.dajour.2023.100307.

[24] A. M. Mahfouz, A. Abuhussein, F. S. Alsubaei, and S. G. Shiva, "Toward A Holistic, Efficient, Stacking Ensemble Intrusion Detection System using a Real Cloud-based Dataset," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, pp. 950–962, 2022, https://doi.org/10.14569/IJACSA.2022.01309110.

[25] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, no. June, p. 100306, 2023, https://doi.org/10.1016/j.array.2023.100306.

[26] Y. Alotaibi and M. Ilyas, "Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security," *Sensors*, vol. 23, no. 12, 2023, https://doi.org/10.3390/s23125568.

[27] O. O. Olasehinde, "A Stacked Ensemble Intrusion Detection Approach for the Protection of Information System," *International Journal for Information Security Research*, vol. 10, no. 1, pp. 910–923, 2020, https://doi.org/10.20533/ijisr.2042.4639.2020.0105.

[28] D. Srivastav and P. Srivastava, "A two-tier hybrid ensemble learning pipeline for intrusion detection systems in IoT networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3913–3927, 2023, https://doi.org/10.1007/s12652-022-04461-0.

[29] D. Tiwari, B. S. Bhati, B. Nagpal, S. Sankhwar, and F. Al-Turjman, "An enhanced intelligent model: To protect marine IoT sensor environment using ensemble machine learning approach," *Ocean Engineering*, vol. 242, no. November, p. 110180, 2021, https://doi.org/10.1016/j.oceaneng.2021.110180.

[30] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi, and M. Alazab, "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors (Switzerland)*, vol. 20, no. 9, pp. 1–37, 2020, https://doi.org/10.3390/s20092559.

[31] R. Golchha, A. Joshi, and G. P. Gupta, "Voting-based Ensemble Learning approach for Cyber Attacks Detection in Industrial Internet of Things," *Procedia Computer Science*, vol. 218, pp. 1752–1759, 2022, https://doi.org/10.1016/j.procs.2023.01.153.

[32] R. Gangula, M. M. Vutukuru, and M. Ranjeeth Kumar, "Intrusion Attack Detection Using Firefly Optimization Algorithm and Ensemble Classification Model," *Wireless Personal Communications*, vol. 132, no. 3, pp. 1899–1916, 2023, https://doi.org/10.1007/s11277-023-10687-8.

[33] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, "An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning," *Expert Systems with Applications*, vol. 213, no. PA, p. 118745, 2023, https://doi.org/10.1016/j.eswa.2022.118745.

[34] Q. Abbas, S. Hina, H. Sajjad, K. S. Zaidi, and R. Akbar, "Optimization of predictive performance of intrusion detection system using hybrid ensemble model for secure systems," *PeerJ Computer Science*, vol. 9, 2023, https://doi.org/10.7717/peerj-cs.1552.

[35] M. H. L. Louk and B. A. Tama, "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," *Expert Systems with Applications*, vol. 213, no. PB, p. 119030, 2023, https://doi.org/10.1016/j.eswa.2022.119030.

[36] M. M. Otoom, K. N. A. Sattar, and M. A. Sadig, "Ensemble Model for Network Intrusion Detection System Based on Bagging Using J48," *Advances in Science and Technology Research Journal*, vol. 17, no. 2, pp. 322–329, 2023, https://doi.org/10.12913/22998624/161820.

[37] D. Mishra, B. Naik, J. Nayak, A. Souri, P. B. Dash, and S. Vimal, "Light gradient boosting machine with optimized hyperparameters for identification of malicious access in IoT network," *Digital Communications and Networks*, vol. 9, no. 1, pp. 125–137, 2023, https://doi.org/10.1016/j.dcan.2022.10.004.

[38] V. Agate, F. M. D'Anna, A. De Paola, P. Ferraro, G. Lo Re, and M. Morana, "A Behavior-Based Intrusion Detection System Using Ensemble Learning Techniques," *CEUR Workshop Proceedings*, vol. 3260, pp. 207–218, 2022.

[39] Y. Cao, Z. Wang, H. Ding, J. Zhang, and B. Li, "An intrusion detection system based on stacked ensemble learning for IoT network," *Computers and Electrical Engineering*, vol. 110, no. March 2023, p. 108836, 2023, https://doi.org/10.1016/j.compeleceng.2023.108836.

[40] Q. A. Al-Haija and M. Al-Dala'ien, "ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, 2022, https://doi.org/10.3390/jsan11010018.

[41] H. C. Lin, P. Wang, K. M. Chao, W. H. Lin, and Z. Y. Yang, "Ensemble learning for threat classification in network intrusion detection on a security monitoring system for renewable energy," 2021, https://doi.org/10.3390/app112311283.

[42] D. N. Mhawi and S. H. Hashim, "Proposed Hybrid Ensemble Learning Algorithms for an Efficient Intrusion Detection System," *Iraqi Journal of Computer, Communication, Control and System Engineering*, vol. 22, no. 2, pp. 73–84, 2022, https://doi.org/10.33103/uot.ijccce.22.2.7.

[43] A. G. Sooai, S. D. B. Mau, Y. C. H. Siki, D. J. Manehat, S. C. Sianturi, and A. H. Mondolang, "Optimizing Lantana Classification: High-Accuracy Model Utilizing Feature Extraction," *Jurnal Ilmiah Kursor*, vol. 12, no. 2, pp. 49–58, 2023, https://doi.org/10.21107/kursor.v12i2.347.

[44] H. Alibrahim and S. A. Ludwig, "Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization," *IEEE Congress on Evolutionary Computation, CEC 2023*, pp. 1–9, 2021.

[45] Y. Sun, S. Ding, Z. Zhang, and W. Jia, "An improved grid search algorithm to optimize SVR for prediction," *Soft Computing*, vol. 25, no. 7, pp. 5633–5644, 2021, https://doi.org/10.1007/s00500-020-05560-w.

[46] A. Allawala, K. Rutherford, and P. Wadhwa, "Rotated Grid Search for Hyperparameter Optimization," *International Journal of Machine Learning and Computing*, vol. 12, no. 5, 2022, https://doi.org/10.18178/ijmlc.2022.12.5.1110.

[47] L. Villalobos-Arias, C. Quesada-López, J. Guevara-Coto, A. Martínez, and M. Jenkins, "Evaluating hyper-parameter tuning using random search in support vector machines for software effort estimation," *PROMISE 2020 - Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, Co-located with ESEC/FSE 2020*, pp. 31–40, 2020, https://doi.org/10.1145/3416508.3417121.

[48] H. Mohammadi, M. Soltanolkotabi, and M. R. Jovanović, "Learning the model-free linear quadratic regulator via random search," *Proceedings of Machine Learning Research*, vol. 120, no. 2, pp. 531–539, 2020.

[49] T. Sandev, V. Domazetoski, A. Iomin, and L. Kocarev, "Diffusionadvection equations on a comb: Resetting and random search," *Mathematics*, vol. 9, no. 3, pp. 1–24, 2021, https://doi.org/10.3390/math9030221.

[50] U. AlHaddad, A. Basuhail, M. Khemakhem, F. E. Eassa, and K. Jambi, "Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks," *Sensors*, vol. 23, no. 17, 2023, https://doi.org/10.3390/s23177464.