Efficient IoT Security: Weighted Voting for BASHLITE and Mirai Attack Detection

Marwan Abu-Zanona
Department of Management Information Systems College of Business Administration
King Faisal University, Al-Ahsa 31982, Saudi Arabia

Abstract—**The increasing number of devices in the Internet of Things (IoT) has exposed various vulnerabilities, such as BASHLITE and Mirai attacks, making it easier for cyber threats to emerge. Due to these vulnerabilities, developing innovative detection and mitigation strategies is essential. Our proposed solution is an ensemble-based weighted voting model that combines different classifiers, including Random Forest, eXtreme Gradient Boosting (XGBoost), Gradient Boosting, K-nearest neighbor (KNN), Multilayer Perceptron (MLP), and Adaptive Boosting (AdaBoost), using artificial intelligence and machine learning. We evaluated our model on the N-BaIoT dataset, a benchmark in this domain. Our results show that the weighted voting approach has exceptional accuracy, precision, recall, and F1-Score. This highlights the effectiveness of our model in classifying various attack instances within the IoT security context. Our approach performs better than other state-of-the-art methods, achieving a remarkable accuracy of 99.9955% in detecting and preventing BASHLITE and Mirai cyber-attacks on IoT devices.**

Keywords*—*Internet of Things; IoT security; BASHLITE; Mirai attacks; ensemble learning

I. Introduction

IoT devices have transformed our daily lives and work [1]. However, the rapid increase in IoT devices has also exposed serious vulnerabilities in the IoT ecosystem [2]. With a vast attack surface, IoT networks have become prime targets for cybercriminals who seek to compromise devices and launch large-scale attacks [3]. Addressing these security concerns and understanding the potential effects of IoT attacks on vital infrastructure, data privacy, and overall societal well-being is essential [4]. IoT devices are vulnerable to numerous security threats due to their constrained resources, diverse communication protocols, and heterogeneous architectures [5]. These vulnerabilities include weak or default credentials, inadequate mechanisms, and the absence of timely software updates [6]. The complexity of security management is further compounded by the heterogeneous nature of IoT networks that comprise devices from various manufacturers [7]. IoT systems face many threats, while the BASHLITE and Mirai attacks are among the most widespread. These attacks have been responsible for numerous large-scale Distributed Denial of Service (DDoS) incidents [8]. The BASHLITE and Mirai attacks have significantly impacted the IoT security landscape, posing a severe threat to connected devices and networks [9]. These cause financial losses by creating powerful botnets. These attacks have become more sophisticated, with malware and botnet operators evolving [10]. Organizations have implemented attack triage systems to combat these threats that help security operators identify and analyze new attack patterns [11]. Analysis of the behavior of these botnets reveals that they rely on infrastructure providers and target specific victims [12]. By understanding the characteristics of these attacks, we can gain valuable insights into the challenges that require innovative detection and mitigation approaches. To effectively secure IoT systems, it is necessary to quickly detect and prevent threats such as BASHLITE and Mirai attacks. Traditional security methods often fail in IoT environments because of the specific features of these devices, such as limited computational resources and constrained communication capabilities [13]. Therefore, efficient detection methods are essential for IoT security. Artificial intelligence and machine learning ensembles can create a more resilient defense against IoT-related threats [14]. Innovative approaches, such as ensemble-based weighted voting, should be adopted to improve detection accuracy and efficiency [15]. Ensemble-based voting optimizes predictive outcomes by fusing diverse algorithms to mitigate inherent uncertainties and strengthen model robustness [16].

Based on the details presented, we will apply the three research questions:

1) How can we effectively detect and prevent BASHLITE and Mirai attacks to enhance the security of IoT systems?
2) What strategies keep up with the changing landscape of BASHLITE and Mirai attacks on IoT networks?
3) How can we improve the accuracy and efficiency of IoT security systems in the face of unique device constraints by applying ensemble-based weighted voting and other artificial intelligence techniques?

The objective of this work is to detect and prevent BASHLITE and Mirai attacks. To achieve this, the researchers developed an ensemble-based weighted voting model that combines machine learning classifiers such as Random Forest, eXtreme Gradient Boosting (XGBoost), Gradient Boosting, K-nearest neighbor (KNN), Multilayer Perceptron (MLP), and Adaptive Boosting (AdaBoost). The model is evaluated through extensive testing on the N-BaIoT dataset and shows different accuracy measurements in identifying attacks in IoT environments. Then compares contemporary techniques and establishes the proposed approach as a state-of-the-art solution. The research focuses on IoT security, explicitly targeting the detection and classification of BASHLITE and Mirai attacks. The key contributions are:

- Ensemble-based weighted voting approach, leveraging machine learning classifiers (Random Forest, XGBoost, KNN, MLP, and AdaBoost).

- Validation on the N-BaIoT Dataset showcased the model's heightened performance in accurately identifying attacks, demonstrating improved accuracy, precision, recall, and F1-Score.
- Comparative analysis of the proposed approach with contemporary techniques, establishing its prowess as a state-of-the-art solution for IoT security.
- The findings provide valuable insights into strengthening IoT networks, showing practical implications for enhancing security measures against growing cyber threats.

The rest of this work is organized as follows: Section II delves into existing related works in the field. Section III details the methodology, covering the N-BaIoT dataset, defining the single classifiers, and defining performance metrics. Subsequently, Section IV details the proposed weighted voting approach, the training process, and the specifics of the ensemble weighted voting technique. The evaluation and results derived from the experimental setup are expounded in Section V. Section VI concludes findings and outlines future research efforts.

II. Related Work

With the increasing threats to IoT networks, there has been a noteworthy rise in research attempts towards IoT security. This section comprehensively reviews the relevant literature on IoT security and attack detection. We analyze the previous studies and approaches that have addressed the challenges IoT devices pose and their vulnerabilities. By synthesizing and analyzing the existing body of work, we aim to gain a deeper understanding of the current state of IoT security research. Ensemble learning techniques have been proposed to identify and classify attacks on IoT networks [17], [18]. These techniques implement machine learning to enhance the detection of security breaches and recommend appropriate mitigation strategies [19]. Ensemble models trained on realistic data have shown promising results in accuracy [20]. Additionally, efficient and lightweight machine learning-based detection systems have been developed to counter attacks on IoT devices [21]. Ensemble learning approaches offer potential solutions for improving attack detection and securing IoT networks.

Alothman et al. [22] proposed an approach using machine learning to detect IoT botnet attacks. This approach was proposed to differentiate malicious traffic from normal traffic and identify botnet types. They utilized the Bot-IoT dataset containing various attack categories and applied preprocessing techniques like Synthetic Minority Over-sampling Technique (SMOTE). Using the Bot-IoT dataset, they tested multiple classifiers and reported the results of the best three classifiers, J48, Random Forest, and MLP. The results showed that the RF and J48 classifiers had superior accuracies of 0.960 and 0.963, respectively.

Alkahtani et al. [23] proposed a hybrid deep learning algorithm to detect botnet attacks on IoT networks. The experimental results showed that the proposed model using the N-BaIoT dataset, achieved near 90% accuracy in detecting attacks from doorbells. For thermostat devices, the proposed system achieved an accuracy of 88.53% in identifying botnet attacks. The proposed system also exhibited high accuracies in detecting botnet attacks from security cameras, achieving from 87.19% to 89.64%.

Karanja et al. [24] designed a method for identity malware in the IoT using Haralick image texture features and three machine learning classifiers. They converted the data to gray scale images and computed the gray-level co-occurrence matrix (GLCM) for each image. Then, they calculated five Haralick features that were used to classify malware using classifiers. The experimental results showed that their approach achieved three results 80%, 89%, and 95% accuracy based on their findings.

Alsamiri et al. [25] used different machine learning algorithms to quickly and effectively detect IoT network attacks. They utilized the Bot-IoT dataset for the evaluation process. During the implementation phase, several algorithms were applied, and most achieved high performance. Additionally, new features were extracted from the Bot-IoT dataset which resulted in better results. Based on the experimental results, the KNN algorithm was found to be the most effective with 99% accuracy.

A research study [26] produced a MedBIoT dataset containing both typical and botnet traffic in the IoT network. The dataset includes data from the primal phase of botnet preparation and features real botnet malware such as Mirai, BASHLITE, and Torii. Machine learning models, both supervised and unsupervised, were built using the data to demonstrate the effectiveness of machine learning-based botnet classification and intrusion detection systems. The experimental results showed that the RF algorithm was the most effective, with an accuracy of 96.17%. The collected dataset has proven suitable and reliable for botnet detection using machine learning techniques.

In [27], the authors proposed new algorithms designed to encrypt data streams in an efficient IoT environment that meets the security requirements of 5G networks. They demonstrated that their algorithms resist various types of attacks, including quantum attacks, eavesdropping, plaintext attacks, chosen ciphertext attacks, and public critical attacks. The authors compared their proposed algorithm with leading post-quantum (PQ) cryptography algorithms such as LWE, LIZARD, and NTRU. According to their findings, the symmetric algorithm they proposed is 70 times faster than the aforementioned symmetric algorithms, and their asymmetric algorithm is ten times faster than the above-stated asymmetric algorithms. Additionally, both the proposed algorithms require 6000 times less memory.

When it comes to IoT devices, we face a challenge: they tend to need more energy and power. Post-quantum cryptography is usually more computationally intensive than the current cryptographic standards. In study [28], authors used the post-quantum digital signature scheme CRYSTALS-Dilithium to authenticate Message Queue Telemetry Transport (MQTT) and measure CPU, memory, and disk usage. They also explored using a key encapsulation mechanism (KEM) trick suggested in 2020 for transport-level security (TLS), which can save up to 90% of CPU cycles. They utilized the post-quantum KEM scheme CRYSTALS-KYBER to compare the resulting CPU, memory, and disk usage with traditional authentication. The study found that using KEM for authentication resulted in a 25 ms speed increase and a 71% savings. Although there were some additional costs for memory, they were minimal enough to be acceptable for most IoT devices.

Some maintain the trade-offs among cost, performance, and security, especially when considering resource-constrained IoT devices. The authors in [29] discussed various S-boxes used in the popular LWC algorithms by their input–output bit-sizes and highlighted their strengths and limitations. Then, it focuses on the proposed 5-bit S-box design. The novel design uses a chaotic mapping theory to offer a random behavior of the element in the proposed S-box. The experimental results from ASIC implementation reveal two essential characteristics of the proposed S-box, cost and performance, and further compare it with 4/5-bit competitors. It demonstrates the security strength of the proposed 5-bit S-box through cryptanalyses such as bijective, nonlinearity, linearity, differential cryptanalysis, differential style boomerang attack, avalanche effect, bit and independence criterion. Also, a comparison is carried out to exhibit the superiority of the proposed 5-bit S-box over its 5-bit competitors.

The article in [1] presented an efficient design method for PSCA-resistant ciphers implemented in hardware using high-level synthesis. The focus is on lightweight block ciphers that use addition, rotation, and XOR-based permutations. They also studied the effects of threshold implementation, which is a secure countermeasure against power side-channel attacks, on the behavioral descriptions of ciphers, along with changes in high-level synthesis scheduling. The proposed method successfully improves the resistance against power side-channel attacks for all addition/rotation/XOR-based ciphers used as benchmarks, as demonstrated by the results obtained using Welch's t-test.

III. Methodology

A. The N-BaIoT Dataset

The N-BaIoT dataset, introduced by Meidan et al. [30], comprises 115 attributes obtained through port mirroring of IoT devices. It includes benign data captured immediately after network setup, featuring two types of packet sizes, packet counts, and jitters. These data samples are categorized based on source IP, source MAC-IP, channel, socket, and total, with attributes such as packet, packet count, and time between packet arrivals. Statistical measures like mean, variance, integer values, magnitude, radius, covariance, and correlation coefficient are covered across 23 features for each of the five-time windows. The dataset incorporates injected BASHLITE and Mirai attacks into various IoT devices, each associated with specific device types and model names. BASHLITE, also known as gafgyt, is a botnet for DDoS attacks on Linux-based IoT devices, employing flooding attacks like UDP and TCP attacks. In contrast, Mirai, developed by Paras, executes large-scale attacks on IoT devices by scanning for vulnerabilities and launching flooding attacks. This dataset offers a comprehensive understanding of IoT device behavior under benign and malicious conditions, providing insights into the impact of different attacks on various device types.

B. Single Classifiers

Our ensemble model leverages the unique features of each classifier. Each classifier has a distinct architecture, hyperparameters, and beneficial features for IoT security. Our approach employs various machine learning models, including the Random Forest Classifier, XGBoost, Gradient Boosting Classifier, KNN Classifier, MLP Classifier, and AdaBoost Classifier, to classify attacks and anomalies in the proposed data. Every model plays an essential role in our approach to enhancing attack detection in IoT ecosystems.

1) Random Forest Classifier: The Random Forest Classifier is a machine learning algorithm for classification and regression tasks [31]. It is a reliable and robust technique that enhances predictive accuracy by utilizing multiple decision trees while minimizing overfitting. The algorithm creates a collection of decision trees using bootstrapping to build each tree from a subset of the dataset. During the creation of each tree, a random subset of features is considered for splitting at each node. This diversity is necessary to ensure that the forest is not overly dependent on a particular set of features, promoting robustness and flexibility in the predictive system [32]. The prediction process involves aggregating the outputs of individual trees, where each tree casts a vote for the class label. Overall, the Random Forest Classifier is a powerful tool for machine learning tasks that require high accuracy and flexibility. Let T denote the set of decision trees in the forest, and $h_i(\mathbf{x})$ represent the prediction of the i-th tree for input vector $\mathbf{x}$. The final prediction $\hat{y}$ is determined by a majority vote:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left(\sum_{i=1}^{|T|} \mathbb{I}\{h_i(\mathbf{x}) = y\} \right) \tag{1}$$

where, $\mathbb{I}\{\cdot\}$ is the indicator function, and $|T|$ signifies the total number of trees in the Random Forest. This approach allows Random Forest to make accurate predictions by combining decisions from multiple trees, improving adaptability, and reducing overfitting.

2) Gradient Boosting Classifier: The Gradient Boosting Classifier is an advanced machine learning algorithm that constructs predictive models sequentially [33]. It achieves this by addressing the errors of its predecessors, which refine the model's accuracy and make it particularly effective for classification tasks. The core idea behind this algorithm is to combine the outputs of multiple weak learners, usually decision trees, in a weighted manner [34]. Each tree is trained to rectify the mistakes of the preceding ones, optimizing the overall predictive performance. The algorithm employs a scheme that minimizes a composite objective function. It contains a loss term that quantifies prediction errors and regularization terms for controlling model complexity. Through gradient descent, the classifier adjusts the parameters of each weak learner to improve its predictive capabilities. This approach enables Gradient Boosting to excel at capturing intricate patterns within data, making it a valuable tool for various real-world applications. In the Gradient Boosting Classifier, the prediction is formulated as an additive ensemble of weak learners, typically decision trees. The overall prediction $F(x)$ is a weighted sum of these weak learners, given by the equation:

$$F(x) = \sum_{t=1}^{T} \alpha_t f_t(x) \tag{2}$$

where, T represents the total number of weak learners in the ensemble, α_t denotes the weight assigned to the t-th weak learner, and $f_t(x)$ is the prediction made by the t-th weak learner. Gradient boosting works by adding weak learners sequentially. Each new learner addresses the residuals (errors) of the combined model from the previous iterations. This approach helps in improving the overall model accuracy gradually by focusing on the previously misclassified instances. During the training process, the weights (α_t) are determined, which depend on the contribution of each weak learner to minimizing the overall loss function.

3) XGBoost Classifier: XGBoost is a highly efficient and versatile ensemble learning algorithm that is widely used for predictive modeling [35]. It is a gradient boosting method that builds a series of weak learners, such as decision trees, and adaptively improves their predictive performance. XGBoost includes regularization terms in its objective function, which promotes model simplicity and reduces overfitting [36]. XGBoost optimizes predictive accuracy with its innovative approach to tree construction and parallel processing. The XGBoost classifier algorithm aims to improve the predictive capabilities of an ensemble of weak learners, typically decision trees. It does this by minimizing the objective function through an iterative boosting process that adjusts the parameters of each weak learner. The final prediction is determined by aggregating the weighted contributions of individual learners, where weights reflect the influence of each learner on the overall model. The success of XGBoost in achieving superior performance across diverse classification scenarios is due to its gradient descent optimization and regularization. The XGBoost classifier is defined as:

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i) \tag{3}$$

While, $\hat{y}_i$ represents the predicted output for the i-th instance, $\mathbf{x}_i$ denotes the feature vector, K is the total number of weak learners, and $f_k(\mathbf{x}_i)$ represents the output of the k-th weak learner. The ensemble prediction is obtained by summing the outputs of all weak learners, and $\phi(\mathbf{x}_i)$ produces the final predicted value. The parameters of the weak learners are adaptively updated during the iterative training process, allowing the model to learn complex relationships in the data.

4) K-Nearest Neighbors (KNN): The KNN classifier is a type of machine learning algorithm used for classification and regression tasks that does not require any pre-defined parameters [37]. It works by determining the classification of a data point based on the majority class of its KNNs in the feature space. The algorithm often uses Euclidean distance as a metric to measure similarity between data points. The consensus of the classes within the neighborhood of a point forms the decision boundary. The KNN algorithm is versatile and can adapt to different data and decision boundaries. However, choosing the appropriate value for k is essential since a small value of k can increase sensitivity to noise, while a high value of k may smooth out local patterns. Despite its simplicity, KNN has proven effective in various applications, especially when decision boundaries are intricate and non-linear [38]. The KNN classifier predicts the class label of a data point by considering the majority class among its k-nearest neighbors in the feature space [39]. Let x be the data point for which we want to make a prediction, and let $N(x)$ denote the set of its k-nearest neighbors based on a specified distance metric. The predicted class label, $\hat{y}$, is determined by the majority class among these neighbors. It is presented in the equation:

$$\hat{y} = \arg\max_{y} \left(\sum_{i=1}^{k} \mathbb{I}\{y_i = y\} \right) \tag{4}$$

where, y_i represents the class labels of the k-nearest neighbors. The decision is made by selecting the class y that maximizes the count of its occurrences among the neighbors. This formulation captures the essence of the KNN algorithm, emphasizing the reliance on local neighborhood information for classification decisions. The choice of the distance metric and the parameter k significantly influences the algorithm's performance and its adaptability to different datasets.

5) Multi-Layer Perceptron (MLP): The MLP classifier is an artificial neural network comprising several layers of interconnected nodes, each serving as a processing unit [40]. It is a versatile and powerful model that can be used for both classification and regression tasks. In classification, the MLP classifier uses a feed-forward architecture, where information moves from the input layer through hidden layers to the output layer. The hidden layers have nodes that use non-linear activation functions, enabling the model to capture complex relationships in the data. Overall, the MLP classifier is a reliable tool for handling complex datasets [41]. Let $\mathbf{x}$ represent the input vector, and W and b denote the weight matrix and bias vector, respectively, for each layer. The output of the MLP is obtained through a series of transformations and activation functions. The prediction $\hat{y}$ is calculated as:

$$\hat{y} = \sigma(W_{\text{out}} \cdot \sigma(W_{\text{hidden}} \cdot \sigma(W_{\text{input}} \cdot \mathbf{x} + b_{\text{input}}) + b_{\text{hidden}}) + b_{\text{out}} \tag{5}$$

σ denotes the activation function, such as the Rectified Linear Unit (ReLU) or hyperbolic tangent (tanh). During training, the weights and biases of MLPs are optimized using back propagation and gradient descent algorithms. The flexibility of these models enables them to learn complex patterns and relationships present in the data, which makes them an excellent fit for a wide range of machine learning applications.

C. Performance Metrics

This section establishes a set of evaluation metrics to assess the performance of our ensemble model and its components quantitatively. We have chosen metrics that help evaluate accuracy, precision, recall, and F1-score. These metrics provide objective benchmarks for comparing our approach with existing detection methods. Additionally, we discuss our methodology for conducting experiments and collecting results. Selecting and interpreting evaluation metrics is essential to ensure a comprehensive and informative assessment of our research outcomes. We use several standard evaluation metrics for accuracy, precision, recall, and F1-score.

- **Accuracy (ACC):** Accuracy is a metric that measures the ratio of correctly predicted instances to the total instances in the dataset. ACC is calculated as,

$$\text{ACC} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{6}$$

- **Precision (P):** Precision measures the model's ability to identify true positives accurately. P is calculated as,

$$\text{P} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{7}$$

- **Recall (R):** Recall is a metric that indicates the percentage of actual positive cases that the model correctly predicted. R is calculated as,

$$\text{R} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

- **F1-Score (F1):** The F1-score provides a balance between precision and recall. F1 is calculated as,

$$\text{F1} = \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}} \tag{9}$$

These metrics are essential to objectively evaluate the performance of our ensemble model in detecting BASHLITE and Mirai attacks. They enable comparisons with other detection methods and offer a comprehensive assessment of our research outcomes.

IV. Proposed Weighted Voting Approach

A. Proposed Approach Overview

The proposed approach involves a weighted voting strategy using an ensemble of five single classifiers (as described in Section III-B). These classifiers are trained with a prepared dataset and their predictions are assigned weights based on their performance and confidence. Fig. 1 provides a clear overview of our proposed approach.

B. Preprocessing

In this step, we will discuss data collection, and preprocessing procedures, namely feature encoding and scaling. Feature encoding involves transforming categorical variables into a numerical format, which helps machine learning algorithms to understand and represent categorical data accurately. Feature scaling is an essential technique that plays a pivotal role in normalizing the range of numerical features. The process helps to prevent any feature from disproportionately influencing the learning process. Feature scaling is an essential technique in machine learning, and the Standard Scaler is one of the most commonly used methods [42]. Its primary purpose is to normalize the range of numerical features within a dataset. This helps to prevent any feature from having an outsized impact on the learning process. By transforming the data distribution to have a mean of 0 and a standard deviation of 1, the Standard Scaler ensures all features are brought to a common scale. For each feature x_i in the dataset, the Standard Scaler computes the z-score by subtracting the mean (μ) of the feature and dividing it by its standard deviation (σ), as expressed in the equation:

$$z = \frac{(x_i - \mu)}{\sigma} \tag{10}$$

Here, z means the standardised set of the characteristic x_i. This process ensures the stability and effectiveness of the training process in interpreting and utilizing numerical features. By meticulously scaling our features, we can improve the performance and reliability of our model in the later stages of our research.

C. Training Machine Learning Classifiers

The training data included labels that explained whether a particular output had an anticipated associated class. The main goal is to train the learning model to perceive the correct position of the unseen data by matching it to the sample data. However, in many cases, we found that a single learning model could have produced the best results or the minimum errors. Therefore, we adopted an ensemble learning technique that involved constructing multiple assumptions on the training data and incorporating them to recognize the correct position of the sample. This method combined the decisions from several models and enhanced the overall efficiency of the model, resulting in more accurate results. Moreover, this approach led to a stable and more robust model than individual models.

To prepare our ensemble model, we meticulously execute the training process for each machine learning classifier making up our ensemble. The classifiers as described in Section III-B, include the Random Forest, XGBoost, Gradient Boosting, KNN, MLP, and AdaBoost Classifiers. Each classifier's diverse architectures, hyperparameters, and unique capabilities contribute to the comprehensive learning process. Subsequently, these trained classifiers collectively form the basis learned for the ensemble approach, paving the way for the ensemble's ultimate strength through Weighted Voting in detecting malicious activities within IoT environments.

D. Ensemble Weighted Voting

The proposed approach relies on the Weighted Voting strategy, which allows us to use the unique strengths of individual models within the ensemble [43]. It delves into the intricacies of Weighted Voting, explaining how it assigns weights to predictions from base trained models based on their performance and confidence [44]. This approach aims to optimize the accuracy of our ensemble system, enabling it to adapt to varying degrees of model reliability. Leveraging these weighted predictions is essential in improving the detection of BASHLITE and Mirai attacks in IoT networks. Let C_i represent the i-th base classifier, where i ranges from 1 to n (n = 5). Each base classifier provides a prediction denoted as P_i, and these predictions collectively form the set $\{P_1, P_2, ..., P_n\}$. Next, individual weights are assigned to these predictions based on the performance or reliability of each base classifier. Let W_i denote the weight assigned to the prediction of classifier C_i. The assignment of weights can be influenced by various factors, such as the accuracy or precision of each base classifier on a validation set. The weighted sum, denoted as S, is computed by summing the product of each prediction and its corresponding weight:

$$S = \sum_{i=1}^{n} W_i \cdot P_i \tag{11}$$

In ensemble learning, the final prediction output is the sum of the five predictions made by all the base classifiers. Each base classifier's prediction has a weight assigned to it, and the final output is a combination of all these weighted predictions. To ensure that the weights form a proper distribution, they are often normalized. This means that each weight is divided by

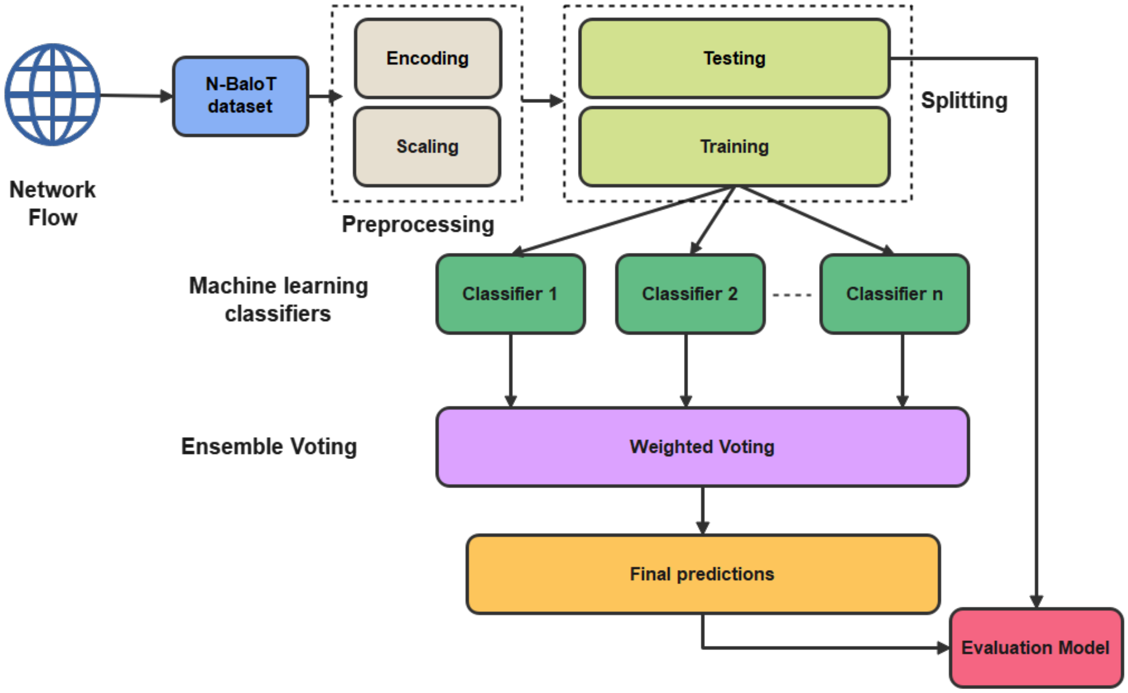

Fig. 1. Flow diagram of the proposed weighted voting attack detection approach.

the sum of all the weights:

$$\text{Normalized Weight, } \bar{W}_i = \frac{W_i}{\sum_{j=1}^{n} W_j} \qquad (12)$$

This normalization guarantees that the weights collectively sum to 1, maintaining the integrity of the weighted voting process. The Weighted Voting mechanism ensures that the final prediction is a well-informed combination of individual base classifier predictions. Each contribution is appropriately weighted to enhance the overall accuracy of the ensemble. The mechanism's effectiveness lies in its ability to assign weights adaptively based on the reliability of each classifier. This helps to improve the predictive accuracy of the ensemble.

V. Evaluation and Results

In this section, we thoroughly evaluate and present the results of our proposed ensemble approach for detecting BASHLITE and Mirai attacks from the N-BaIoT dataset in IoT networks. Our experimental evaluation meticulously analyzes the model's performance using key metrics such as accuracy, precision, recall, and F1-score. We provide insightful analyses of the experimental setup and the obtained results and then conduct a comparative assessment with other existing detection methods. This comprehensive evaluation presents a detailed perspective on the strengths of our ensemble model and potential enhancements in IoT security.

A. Experimental Setup

TABLE I. Parameters and Methods for the Machine Learning Models

Model	**The parameters used for the model experiments.**
Random Forest	- **Number of Trees :** 100. - **Criterion for Splitting (criterion):** 'gini'. Methods:'entropy'.
Gradient Boosting	- **Number of Boosting Stages (n_estimators):** 100. Methods: Tune based on the trade-off between performance and computational cost. - **Loss Function (loss):** 'deviance'. Methods:'exponential'. - **Learning Rate:** 0.1.
K Nearst Neighbors	- **Number of Neighbors (n_neighbors):** 5. Methods: Square root of the number of samples. - **Weight Function (weights):** 'uniform'.
MLP	- **Number of Neurons in Hidden Layers (hidden_layer_sizes):** 100.- **Activation Function (activation):** 'relu'. Methods: 'tanh'.
AdaBoost	- **Number of Weak Learners (n_estimators):** 50. - **Learning Rate (learning_rate):** 1.0.
XGBoost	- **Number of Trees (n_estimators):** 100.

In this section, we explain the experimental settings of our contribution designed for the detection of BASHLITE and Mirai attacks in IoT networks. We provide a detailed account of the implementation steps and propose solutions aimed at validating the effectiveness of our approach in enhancing the security of IoT environments. A critical aspect of the validation process involves the assessment of the N-BaIoT dataset, underscoring its significance in evaluating any proposed solution for IoT security improvement. Table I summarizes parameters and methods employed for various machine learning models, explaining the comprehensive experimental setup to strengthen IoT security. We split the N-BaIoT dataset into two sets-training and testing, with 80% for training and 20% for model evaluation. This helped balance model learning and validation, measuring our ensemble model's real-world performance while minimizing overfitting.

B. Experimental Results

The study employed Python on Google Colab GPU for multiclass classification [45]. To detect BASHLITE and Mirai attacks, we employed five machine learning techniques (Random Forest, XGBoost, Gradient Boosting, KNN, MLP, and AdaBoost). The results were combined using a weighted voting technique. The ensemble approach and the five classifiers were evaluated using performance evaluation measures such as accuracy, precision, recall, and F1-score. The performance of different machine learning classifiers was compared, including the Ensemble Weighted Voting approach and individual classifiers. The results presented in Tables II, and III, show the weighted and macro average performance metrics.

TABLE II. Weighted Averaged for Different Metrics Across Models (in %)

Model	Accuracy	Precision	Recall	F1-Score
Ensemble Weighted Voting	99.9955	99.9955	99.9913	99.9955
XGBoosting	99.9796	99.9796	99.9624	99.9796
KNN	99.9087	99.9087	99.8669	99.9087
Random Forest	99.9909	99.9909	99.9847	99.9909
MLP	99.9781	99.9781	99.9663	99.9781
AdaBoost	69.13	65.14	59.33	65.14

TABLE III. Macro Averaged Metrics for Different Models (in %)

Model	Precision	Recall	F1-Score
Ensemble Weighted Voting	99.99	99.99	99.99
XGBoosting	99.97	99.96	99.96
KNN	99.87	99.86	99.87
Random Forest	99.98	99.98	99.98
MLP	99.97	99.96	99.97
AdaBoost	66.20	66.20	66.20

The Ensemble Weighted Voting model outperformed the individual classifiers across various metrics such as accuracy, precision, recall, and F1-score. The Ensemble Weighted Voting model has shown exceptional accuracy in the Weighted Averaged metrics, achieving a score of 99.9955%. This is higher than all other classifiers, including XGBoosting, KNN, Random Forest, MLP, and AdaBoost. Additionally, the model's precision, recall, and F1-Score are consistently superior, reaching 99.9955%, 99.9913%, and 99.9955%, respectively. Its higher precision and recall values indicate a better ability to correctly identify and classify instances of attacks, leading to a high F1-score. In the macro-averaged metrics, which measure the average performance across different classes, the Ensemble Weighted Voting model outperforms its competitors with precision, recall, and F1-Score values of 99.99%, 99.99%, and 99.99%, respectively. On the other hand, the individual classifiers, including AdaBoost, exhibited lower precision, recall, and F1-score performance. The ensemble's weighted average results highlight its effectiveness in combining diverse models, emphasizing the importance of ensemble learning in achieving improved detection capabilities in IoT security applications. This indicates that the model is robust and reliable, with a superior ability to detect various classes within the IoT security context.

C. Discussion

This section discusses the implications of the study's findings and the recent techniques used to detect and prevent BASHLITE and Mirai cyber-attacks on IoT devices based on evaluating the N-BaIoT dataset. Our proposed solution focused on an ensemble-based weighted voting model, representing a comprehensive IoT security approach. This model combines various classifiers, including Random Forest, XGBoost, Gradient Boosting, KNN, MLP, and AdaBoost, utilizing artificial intelligence and machine learning techniques. The evaluation of the N-BaIoT dataset, a recognized benchmark in the domain, demonstrates the exceptional performance of the weighted voting approach. The model achieves outstanding accuracy, precision, recall, and F1-Score, surpassing state-of-the-art methods. Notably, the accuracy of 99.9955% in detecting and preventing BASHLITE and Mirai cyber-attacks on IoT devices is a remarkable highlight. Comparing our approach with recent works, as summarized in Table IV, affirms the superiority of the proposed model. Abu Al-Haija and Al-Dala'ien [46] used different machine learning techniques AdaBoosted, RUSBoosted, and bagged, achieving a detection accuracy of 99.6% for botnet attacks.

TABLE IV. Comparison Between Most Recent Works and the Proposed Approach with Results Findings

Authors	Dataset	Results in %
Ours	N-BaIoT	99.9955
Abu Al-Haija and Al-Dala'ien [46]	N-BaIoT	99.6
Okur et al. [47]	N-BaIoT	99.92
Sakthipriya et al. [48]	N-BaIoT	95.02
Abbasi et al. [49]	N-BaIoT	above 90
Hezam et al. [50]	N-BaIoT	89.75

Okur et al. [47] reported a 99.92% accuracy detection rate using Random Forest in the N-BaIoT dataset. These results provide a context for understanding the competitive edge of our ensemble-based approach. Furthermore, examining alternative methods, Sakthipriya et al. [48] focused on dimensionality reduction, with auto-encoder outperforming PCA with an accuracy of 95.02%. Abbasi et al. [49] proposed logistic regression for intrusion detection, achieving above 90% classification accuracy, while Hezam et al. [50] explored deep learning algorithms, with RNN achieving the highest accuracy of 89.75%. The comparison presented in Table IV highlights the superior accuracy of our proposed model compared to recent uses the N-BaIoT dataset. Our model's performance was a comprehensive ensemble of classifiers used to prevent IoT security issues with robustness and effectiveness. The evaluation process was meticulous, ensuring the accuracy of the results.

VI. Conclusion and Future Scope

The increasing number of devices in the IoT has also led to an increase in security risks, such as BASHLITE and Mirai attacks. To address these vulnerabilities, we need innovative detection and mitigation strategies. In this study, we present a new solution based on an ensemble-based weighted voting model that uses a variety of classifiers, including Random Forest, XGBoost, Gradient Boosting, KNN, MLP, and AdaBoosting, powered by artificial intelligence and machine learning. We rigorously evaluated the model's effectiveness on the N-BaIoT dataset, a benchmark in the IoT security domain. Our results indicate that the proposed weighted voting approach achieves exceptional accuracy, precision, recall, and F1-Score in accurately classifying various attack instances in the IoT security context, outperforming other state-of-the-art methods. Notably, the model demonstrates an outstanding accuracy rate of 99.9955% in detecting and preventing BASHLITE and Mirai cyber-attacks on IoT devices. The proposed ensemble-based weighted voting model is designed to overcome the challenges posed by BASHLITE and Mirai attacks and provides valuable insights into IoT networks. Combining different machine learning classifiers, the model shows superior performance metrics to individual classifiers, making it adaptable to changing attack patterns. This study aims to protect against current IoT security threats, providing a robust defense model.

In the future, we can explore how attack methods evolve and test more machine learning classifiers to see how well the proposed ensemble-based approach adapts. We can also study how well the model performs in real-world IoT environments, and we can improve its practical usefulness by testing how it handles larger datasets. Continuously improving and expanding the model using different artificial intelligence techniques and cybersecurity advancements ensure its effectiveness in the ever-changing landscape of IoT security.

Acknowledgment

The researcher extends his sincere thanks and appreciation to the administration of King Faisal University in the Kingdom of Saudi Arabia for providing all forms of support to the university's faculty members, especially in the field of scientific research. We extend our sincere thanks to the Deanship of Scientific Research at the university for agreeing to support this scientific research with GRANT number 5,290.

References

[1] D. Singh, "Internet of things," *Factories of the Future: Technological Advancements in the Manufacturing Industry*, pp. 195–227, 2023.

[2] S. Ahmed and M. Khan, "Securing the internet of things (iot): A comprehensive study on the intersection of cybersecurity, privacy, and connectivity in the iot ecosystem," *AI, IoT and the Fourth Industrial Revolution Review*, vol. 13, no. 9, pp. 1–17, 2023.

[3] P. K. Sadhu, V. P. Yanambaka, and A. Abdelgawad, "Internet of things: Security and solutions survey," *Sensors*, vol. 22, no. 19, p. 7433, 2022.

[4] M. H. P. Rizi and S. A. H. Seno, "A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city," *Internet of Things*, vol. 20, p. 100584, 2022.

[5] B. B. Gupta and M. Quamara, "An overview of internet of things (iot): Architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, p. e4946, 2020.

[6] M. D. Iannacone and R. A. Bridges, "Quantifiable & comparable evaluations of cyber defensive capabilities: A survey & novel, unified approach," *Computers & Security*, vol. 96, p. 101907, 2020.

[7] A. E. Omolara, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, H. Arshad *et al.*, "The internet of things security: A survey encompassing unexplored areas and new insights," *Computers & Security*, vol. 112, p. 102494, 2022.

[8] P. Kumari and A. K. Jain, "A comprehensive study of ddos attacks over iot network and their countermeasures," *Computers & Security*, p. 103096, 2023.

[9] M. H. Aysa, A. A. Ibrahim, and A. H. Mohammed, "Iot ddos attack detection using machine learning," pp. 1–7, 2020.

[10] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, and W. Meira, "The evolution of bashlite and mirai iot botnets," pp. 00 813–00 818, 2018.

[11] S. Coltellese, F. Maria Maggi, A. Marrella, L. Massarelli, and L. Querzoni, "Triage of iot attacks through process mining," pp. 326–344, 2019.

[12] G. Bastos, A. Marzano, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, Í. Cunha, D. Guedes, and W. Meira, "Identifying and characterizing bashlite and mirai c&c servers," pp. 1–6, 2019.

[13] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

[14] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, "Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions," *Mobile Networks and Applications*, vol. 28, no. 1, pp. 296–312, 2023.

[15] A. M. Bamhdi, I. Abrar, and F. Masoodi, "An ensemble based approach for effective intrusion detection using majority voting," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 664–671, 2021.

[16] P. Verma, A. R. K. Kowsik, R. Pateriya, N. Bharot, A. Vidyarthi, and D. Gupta, "A stacked ensemble approach to generalize the classifier prediction for the detection of ddos attack in cloud network," *Mobile Networks and Applications*, pp. 1–15, 2023.

[17] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An ensemble learning based intrusion detection model for industrial iot security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273–287, 2023.

[18] M. M. Alani and E. Damiani, "Xrecon: An explainbale iot reconnaissance attack detection system based on ensemble learning," *Sensors*, vol. 23, no. 11, p. 5298, 2023.

[19] K. Keserwani, A. Aggarwal, and A. Chauhan, "Attack detection in industrial iot using novel ensemble techniques," pp. 1–6, 2023.

[20] M. Koppula, L. J. LM *et al.*, "Lnkdsea: Machine learning based iot/iiot attack detection method," pp. 655–662, 2023.

[21] N. Pandey and P. K. Mishra, "Detection of ddos attack in iot traffic using ensemble machine learning techniques," *NHM*, vol. 18, no. 3, pp. 1393–1409, 2023.

[22] Z. Alothman, M. Alkasassbeh, and S. Al-Haj Baddar, "An efficient approach to detect iot botnet attacks using machine learning," *Journal of High Speed Networks*, vol. 26, no. 3, pp. 241–254, 2020.

[23] H. Alkahtani and T. H. Aldhyani, "Botnet attack detection by using cnn-lstm model for internet of things applications," *Security and Communication Networks*, vol. 2021, pp. 1–23, 2021.

[24] E. M. Karanja, S. Masupe, and M. G. Jeffrey, "Analysis of internet of things malware using image texture features and machine learning techniques," *Internet of Things*, vol. 9, p. 100153, 2020.

[25] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 12, 2019.

[26] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "Using medbiot dataset to build effective machine learning-based iot botnet detection systems," pp. 222–243, 2020.

[27] A. Kaushik, L. S. S. Vadlamani, M. M. Hussain, M. Sahay, R. Singh, A. K. Singh, S. Indu, P. Goswami, and N. G. V. Kousik, "Post quantum public and private key cryptography optimized for iot security," *Wireless Personal Communications*, vol. 129, no. 2, pp. 893–909, 2023.

[28] J. Samandari and C. Gritti, "Post-quantum authentication in the mqtt protocol," *Journal of Cybersecurity and Privacy*, vol. 3, no. 3, pp. 416–434, 2023.

[29] V. A. Thakor, M. A. Razzaque, A. D. Darji, and A. R. Patel, "A novel 5-bit s-box design for lightweight cryptography algorithms," *Journal of Information Security and Applications*, vol. 73, p. 103444, 2023.

[30] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[31] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines," *Ore Geology Reviews*, vol. 71, pp. 804–818, 2015.

[32] J. Hatwell, M. M. Gaber, and R. M. A. Azad, "Chirps: Explaining random forest classification," *Artificial Intelligence Review*, vol. 53, pp. 5747–5788, 2020.

[33] O. Alshboul, A. Shehadeh, G. Almasabha, and A. S. Almuflih, "Extreme gradient boosting-based machine learning approach for green building cost prediction," *Sustainability*, vol. 14, no. 11, p. 6651, 2022.

[34] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, and L. Gu, "Feature selection based on artificial bee colony and gradient boosting decision tree," *Applied Soft Computing*, vol. 74, pp. 634–642, 2019.

[35] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using xgboost," *Information*, vol. 9, no. 7, p. 149, 2018.

[36] K. Budholiya, S. K. Shrivastava, and V. Sharma, "An optimized xgboost based diagnostic system for effective prediction of heart disease," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 7, pp. 4514–4523, 2022.

[37] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[38] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," pp. 1255–1260, 2019.

[39] A. X. Wang, S. S. Chukova, and B. P. Nguyen, "Ensemble k-nearest neighbors based on centroid displacement," *Information Sciences*, vol. 629, pp. 313–323, 2023.

[40] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in iot networks based on deep learning," *Sensors*, vol. 21, no. 9, p. 2987, 2021.

[41] J. Naskath, G. Sivakamasundari, and A. A. S. Begum, "A study on different deep learning algorithms used in deep neural nets: Mlp som and dbn," *Wireless Personal Communications*, vol. 128, no. 4, pp. 2913–2936, 2023.

[42] M. M. Ahsan, M. P. Mahmud, P. K. Saha, K. D. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021.

[43] A. Dogan and D. Birant, "A weighted majority voting ensemble approach for classification," pp. 1–6, 2019.

[44] A. Nazir, J. He, N. Zhu, A. Wajahat, X. Ma, F. Ullah, S. Qureshi, and M. S. Pathan, "Advancing iot security: A systematic review of machine learning approaches for the detection of iot botnets," *Journal of King Saud University-Computer and Information Sciences*, p. 101820, 2023.

[45] M. Canesche, L. Bragança, O. P. V. Neto, J. A. Nacif, and R. Ferreira, "Google colab cad4u: Hands-on cloud laboratories for digital design," pp. 1–5, 2021.

[46] Q. Abu 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, p. 18, 2022.

[47] C. Okur, A. Orman, and M. Dener, "Ddos intrusion detection with machine learning models: N-baiot data set," pp. 607–619, 2022.

[48] N. Sakthipriya, V. Govindasamy, and V. Akila, "A comparative analysis of various dimensionality reduction techniques on n-baiot dataset for iot botnet detection," pp. 1–6, 2023.

[49] F. Abbasi, M. Naderan, and S. E. Alavi, "Intrusion detection in iot with logistic regression and artificial neural network: further investigations on n-baiot dataset devices," *Journal of Computing and Security*, vol. 8, no. 2, pp. 27–42, 2021.

[50] A. A. Hezam, S. A. Mostafa, A. A. Ramli, H. Mahdin, and B. A. Khalaf, "Deep learning approach for detecting botnet attacks in iot environment of multiple and heterogeneous sensors," pp. 317–328, 2021.