Journal of Robotics and Control (JRC)
Volume 5, Issue 4, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i4.22005

Advanced Threat Detection Using Soft and Hard Voting Techniques in Ensemble Learning

Hanan Ghali Jabbar [1*]
[1] Department of Postgraduate Studies, University of Technology, Baghdad, Iraq
Email: [1] Hanan.G.Jabar@uotechnology.edu.iq
*Corresponding Author

Abstract—**This study addresses the challenge of detecting network intrusions by exploring the efficacy of ensemble learning methods over traditional machine learning models. The problem of network security is exacerbated by sophisticated cyber-attack techniques that standard single model approaches often fail to counter effectively. Our solution employs a robust ensemble methodology to improve detection rates. The research contribution lies in the comparative analysis of individual machine learning models—K-Nearest Neighbors (KNN), Decision Trees (DT), and Gradient Boosting (GB)—against ensemble methods employing soft and hard voting classifiers. This study is one of the first to quantify the performance gains of ensemble methods in the context of network intrusion detection. Our methodological approach involves applying these models to the WSNBFSF dataset, which consists of traffic types including normal operations and various attacks. Performance metrics such as accuracy, precision, recall, and F1-score are calculated to assess the effectiveness of each model. The ensemble methods combine the strengths of individual models using voting systems, which are tested against the same metrics. Results indicate that while individual models like DT and GB achieved near-perfect accuracy scores (99.95% and 99.9%, respectively), the ensemble models performed even better. The soft voting classifier achieved an accuracy of 99.967%, and the hard voting classifier reached 100%, demonstrating their superior capability in network traffic classification and intrusion detection. In conclusion, the integration of ensemble methods significantly enhances the detection accuracy and reliability of network intrusion systems. Future research should explore additional ensemble techniques and consider scalability and class imbalance issues to further refine the efficacy of intrusion detection systems.**

Keywords—Network Intrusion Detection; Machine Learning; Ensemble Learning; K-Nearest Neighbors; Decision Tree; Gradient Boosting; Voting Classifiers; Cybersecurity.

I. INTRODUCTION

The widespread adoption of advanced technologies such as big data, the Internet of Things (IoT), and cloud computing has resulted in a phenomenon known as data deluge, where there is a massive simultaneous generation of data by humans and IoT devices [1]. This has significantly impacted both society and the business world in various ways. McKinsey, as referenced in [2], highlights that the competitive advantage in today's global market is driven by effectively utilizing big data and cutting-edge technologies for productivity and efficiency gains. Consequently, these infrastructures have garnered attention not only from governments and industries but also from malicious actors attempting to gain unauthorized access to valuable and sensitive data.

However, it's important to note that the data generated in many big data and real-world applications often exhibit asymmetric and symmetric distributions. For instance, there are symmetric relationships among social network data and asymmetric probability distributions between mali- cious and regular network traffic. Despite the presence of missing information in these applications, they still contain valuable hidden patterns and knowledge. Therefore, there is a need for efficient methods to filter and extract these valuable patterns [3].

Additionally, the increasing reliance on the internet and its services has exposed computer systems and networks to persistent cybersecurity risks [4]. Various types of cyber-attacks have evolved significantly over time, posing serious threats to governments, businesses, and individuals alike [5]. Despite efforts by security experts to implement defense mechanisms, hackers continuously find ways to carry out sophisticated and automated cyber-attacks, resulting in substantial damages.

Intrusion Detection Systems (IDSs) have gained popularity in recent decades due to their ability to detect intrusions in real-time [4][6].

For example, in [7], the authors discuss the importance of security properties in monitoring cloud computing platforms and propose a novel three-level cloud- based IDS that uses rules to represent monitoring properties effectively. This approach leverages virtualization architecture to enhance security significantly and support automatic reconfigurations of applications.

According to [8], Intrusion Detection involves monitoring events in a computer system or network and analyzing them for signs of intrusion, where an intrusion is an attempt to bypass security mechanisms and compromise Confidentiality, Integrity, and Availability (CIA). IDSs, as defined in [9][10], are hardware or software programs that monitor malicious activities within computer systems and networks based on various parameters such as network packets, system logs, and rootkit analysis. Detailed roles and functionalities of standard IDSs can be found in [11].

Intrusion Detection Systems (IDS) have evolved from simple mechanisms designed in the early 1980s to sophisticated networks capable of detecting and mitigating emerging threats through advanced heuristic and behavioral techniques [40]. Today, IDS methodologies primarily include misuse (signature-based) and anomaly-based detection systems. Misuse detection systems are adept at recognizing

known threats but fail against novel, unseen attacks [Reference to studies on misuse detection systems]. Anomaly-based systems can identify zero-day exploits by detecting deviations from normal activity; however, they suffer from high false positive rates, leading to potential oversight of legitimate threats [41]. However, in recent years, a hybrid-based technique has emerged as a more robust and effective approach by combining the strengths of both methods [12].

In a Misused Intrusion Detection System (MIDS), specific signatures of known attacks are stored and matched with real-time network events to detect intrusions or intrusive activities. Meanwhile, MIDS has a good track record for known malicious activities but it has a dip in foreseeing new attacks as well as processing a huge amounts of signature profiles [13][14][15]. In addition, as case shown in 17 where a signature-base system attained high accurate detection of SQL injects in databases, but the system is limited to the SQL type of intrusion.

As opposed to behavioral anomaly-based IDS, anomaly-based IDSs follow a pattern of behaviors, and it is when these behaviors change in an unusual way that it becomes a possible attack. With such detection, IDSs are able to detect new attacks like zero-day exploits [16]. On the other hand, some of the issues that they confront are not specific to this war, including a high rate of false alerts that can result from the rapid improvement of cyber-attack techniques [17]. As a consequence, [18] describes an optimized anomaly-based ensemble classifier, which efficacy and precision is reportedly the best.

By all means, IDSs are developed intensively while the rest problems, e.g. missing the attack, inconsistent accuracy, sending wrong alarms, etc. stay open [19].

Machine learning algorithms have recently appeared as the most frequently used tools to solve these problems, with the supervised learning and unsupervised learning being most typical approaches [10]. A supervised learning model relies on labeled training data to achieve high accuracy for recognized attacks but it is often hard to scale it out, while an unsupervised one can work with unlabeled data and therefore is easily scalable and affordable [20].

Recent advancements have integrated machine learning models to enhance IDS capabilities. For instance, a lightweight IDS tailored for the IoT context uses a Support Vector Machine (SVM) classifier to detect DDoS attacks with minimal computational resources [38]. Similarly, another study compares various ML algorithms—Logistic Regression, SVM, Decision Tree, Random Forest, and Artificial Neural Network—in detecting attacks and anomalies in IoT infrastructure, achieving high accuracy and demonstrating the effectiveness of ensemble approaches [39].

Despite the advancements, single ML models in IDS face several challenges that compromise their effectiveness in dynamic cybersecurity environments. Firstly, these models are prone to overfitting, performing well on training datasets but failing to generalize to new, unseen data. This issue is critical in cybersecurity, where new types of attacks constantly emerge. Secondly, single models often struggle with handling diverse data and attack types, typically being optimized for specific scenarios and therefore not performing well across varied environments or against different kinds of threats. Additionally, once trained, these models can be inflexible, requiring comprehensive retraining to adapt to new threats, a process that is both time-consuming and resource-intensive. Moreover, even the best-performing single models have limitations in terms of accuracy, recall, precision, and other metrics, particularly in complex, noisy, or imbalanced datasets typical of cybersecurity settings. Finally, individual models may harbor unique biases or assumptions that can lead to errors, affecting the reliability and robustness of the IDS. These limitations underscore the need for ensemble learning approaches that combine multiple models to mitigate these issues, enhancing both the adaptability and accuracy of intrusion detection systems.

Building on the existing foundation and addressing the gaps identified, this study proposes a focused approach to enhance the effectiveness and efficiency of Intrusion Detection Systems (IDS) through the integration of ensemble learning techniques. The specific research objectives are to:

1. Develop and test a robust ensemble learning framework that combines multiple machine learning models to improve the detection accuracy and generalization across diverse cybersecurity scenarios.
2. Investigate the performance of this ensemble approach in comparison to traditional single-model IDS methods, particularly in its ability to reduce overfitting and adapt to new, emerging cyber threats without extensive retraining.
3. Conduct a comparative analysis with previous research to establish the benchmarking performance of the proposed ensemble framework and underscore its contributions to the field.

The novelty of this research lies in its strategic focus on ensemble learning as a solution to the inherent limitations of single-model IDS, such as vulnerability to overfitting and poor adaptability to new attack vectors. By leveraging a variety of machine learning algorithms—such as K-Nearest Neighbors (KNN), Decision Tree (DT), and Gradient Boosting (GB)—this study aims to create a more dynamic and flexible IDS that can respond more effectively to the evolving landscape of cyber threats. Ensemble methods, by integrating diverse models, offer a promising approach to overcome the performance bottlenecks of traditional IDS solutions, particularly in handling complex, noisy, and imbalanced data sets typical in cybersecurity environments.

The contributions are twofold:

1. Enhanced Detection Capability: The research empirically validates that ensemble learning markedly elevates the detection rates of intrusion detection systems (IDS). By integrating multiple machine learning models, this approach is demonstrated to effectively identify both known threats and novel cyber anomalies that elude traditional single-model systems. This enhancement not only strengthens security protocols but also extends the functional capabilities of existing IDS frameworks.

2. Adaptability and Scalability: Additionally, this study showcases the superior adaptability and scalability of ensemble-based IDS solutions. These systems adeptly adjust to evolving cyber threat landscapes with minimal need for manual reconfiguration or extensive retraining. This flexibility ensures that security measures remain robust and effective, even as new types of threats emerge, thereby providing a sustainable, scalable solution for maintaining high security levels across diverse operational environments.

In conclusion, this research is expected to make substantial contributions to the field of cybersecurity by demonstrating the practical benefits and strategic importance of employing ensemble learning techniques in intrusion detection. The anticipated results are likely to influence both future research directions and the development of more effective, resilient cybersecurity technologies.

II. RELATED WORK

Rose et al. [21] constructed an approach that depends mainly on the network profiling and a machine learning system as a cyber safeguarding technique for IoT devices. The anomaly-based intrusion system is built that perpetually monitors networked device activites to discern tempering and suspicious operations. Their methodology, with its reliability of 98.35%, achieved an accuracy rate equivalent to the number of false-positive alarms. It was accomplished by eventually integrating the methodology onto the Cyber-Trust platform.

In the study by Ali et al. [22], a comprehensive machine learning approach was employed to identify IoT devices, utilizing NFStream to extract 85 attributes from network traffic. The study narrowed down these attributes to 20 using the information gain method and tested six machine learning models, with random forest and naïve Bayes classifiers achieving up to 99% accuracy in identifying IoT devices.

El-Sayed et al. [23] analyzed seven supervised learning algorithms to determine the most efficient one for IoT security, grouping them into CNN-based classifiers (two-layer CNN, four- layer CNN, VGG16) and traditional classifiers (logistic regression, support vector machine, and K-nearest neighbors). They found that the SVM algorithm performed best, reaching 94% accuracy on MobileNetv2 features, due to its efficient and resource-light training process.

Le K-H et al. [24] introduced IMIDS, a smart intrusion detection system for IoT, centered around a lightweight convolutional neural network that efficiently classifies various cyber threats.

The system's performance significantly improved with additional training data from an attack data generator, achieving an average F-measure of 97.22%, indicating its potential as an effective IDS for IoT environments.

Joo et al. [25] proposed an IoT intrusion detection system based on deep learning, using a CNN to achieve 86.2% accuracy. They enhanced this with a hybrid method, incorporating machine learning classifiers instead of the CNN's fully connected layers, which increased accuracy to around 87%. The integration of the Xception model with a bidirectional GRU further improved performance, yielding a 95.6% accuracy rate.

Bendiab et al. [26] designed a new concept which is driven by deep learning techniques and visual representing that is used for the analysis of malware traffic. The main target is zero-day malware which works in different ways. This method has been tested on a dataset containing 1,000 pcap files, each comprising 500 malware and 500 benign files. It exhibited good detection with 94.50 percent malware detection rate with the ResNet50 model.

A study on machine leaning techniques [27] that result from MQTT-based attacks was evaluated by six machine learning methods in respect to both unidirectional and bidirectional flow features at different abstraction levels, thus preventing the readers from understanding clearly the key take away of the study. The findings of the study, based on MQTT data simulation, gave evidence of mapping capabilities between the flow based features and MQTT type attacks, as well as the model accuracy that was at quite a high level; 99.04

Sapre et al. [24] explored the head to head comparison of KDDCup99 and NSLKDD datasets to gauge the performance of machine learning classifiers. They concluded that the NSL-KDD dataset is superior, as classifiers trained on it were less prone to redundancy and exhibited more accurate performance, despite a 20.18% lower accuracy compared to those trained on the KDDCup99 dataset.

Liu et al. [28] investigated the impact of various attacks on IoT sensors and networks using the NSL-KDD dataset, evaluating eleven machine learning techniques. Their findings indicated that tree-based and ensemble methods outperformed others, with XGBoost leading in accuracy (97%), MCC (90.5%), and AUC (99.6%). The study also highlighted the effectiveness of the expectation-maximization technique in detecting attacks, surpassing the naïve Bayes classifier by 22% in accuracy.

Amouri et al. [29] utilized a two-stage process to differentiate benign and malicious nodes in a network. Initially, data were gathered by dedicated sniffers (DSs), followed by the generation and dispatch of the Comprehensive Cyber-Intelligence (CCI) to the super node (SN). The SN then applied linear regression on the CCIs from various DSs to identify malicious nodes. They tested this approach using random waypoint (RWP) and Gauss Markov (GM) mobility models, focusing on black hole and DDoS attacks. The system demonstrated high detection rates, particularly in high-velocity conditions, achieving over 98% effectiveness.

Fenanir et al. [30] developed a lightweight IDS employing a filter-based feature selection method to minimize computational demands. Their investigation included various machine learning algo- rithms, with the decision tree (DT) method emerging as the most effective, demonstrating high accuracy across different datasets, reaching up to 98%.

Islam et al. [31] explored various IoT threats and evaluated both shallow and deep learning IDSs, using

decision tree (DT), random forest (RF), SVM, DNN, DBN, LSTM, stacked LSTM, and Bi-LSTM models. They assessed these models using multiple datasets, including NSL-KDD and IoTID20, finding that deep learning-based IDSs outperformed their shallow counterparts, with notable accuracies up to 98.79%.

Ahmad et al. [32] proposed a feature clustering approach using the UNSW-NB15 dataset, ad- dressing common data analysis challenges like overfitting and unbalanced datasets. They applied machine learning methods, including random forest (RF), support vector machine (SVM), and artificial neural networks, achieving significant accuracies in binary and multiclass classifications, with RF showing particularly high accuracy.

Saba et al. [33] introduced a two-stage hybrid method for enhancing IDS accuracy. Initially, a genetic algorithm (GA) selected pertinent features, followed by the application of SVM, ensemble classifiers, and decision trees, achieving a remarkable 99.8% accuracy on the NSL-KDD dataset.

Smys et al. [34] proposed a hybrid CNN model for an IoT IDS, capable of detecting various types of attacks. Their approach demonstrated a high sensitivity to threats, with an accuracy rate of 98.6%.

Papafotikas et al. [35] developed a digital system that uses an ML-based clustering method, specifically K-means, to detect suspicious activities in IoT devices, showcasing effective identifica- tion capabilities.

Lastly, in [36], an IDS utilizing a combined machine learning model was introduced, integrating three different datasets to create a novel architecture, resulting in a promising accuracy of 95.18%.

Table I presents a summary of previous works about IDSs.

TABLE I. SUMMARY OF RELATED WORKS

Ref.	Study Description	Key Findings
[21]	Rose et al. developed a network profiling and machine learning framework to safe-guard IoT devices, employing an anomaly-based intrusion detection system tested on the Cyber-Trust platform.	High efficacy in detecting tampering and suspiciousactivities with 98.35% accuracy and false-positive rate.
[22]	Ali et al. used NFStream to extract 85 attributes from network traffic, narrowingthem down to 20 using the information gain method and testing six machine learning models.	Random forest and naïve Bayes classifiers achieved up to 99% accuracy in identifying IoT devices.
[23]	El-Sayed et al. analyzed seven supervised learning algorithms to find the most efficient for IoT security, comparing CNN-based and traditional classifiers.	SVM algorithm performed best with 94% accuracy using MobileNetv2 features.
[24]	Le K-H et al. introduced IMIDS, a smart intrusion detection system for IoT, based on a lightweight convolutional neural net-work.	Improved performance with additional data, achieving a 97.22% F-measure.
[25]	Joo et al. proposed a deep learning-based IoT intrusion detection system, enhancingit with a hybrid method incorporating machine learning classifiers.	Achieved 95.6% accuracy with the integration ofthe Xception model and abidirectional GRU.
[26]	Bendiab et al. developed a method for IoT security using deep learning and visual representation to analyze malware traffic, targeting zero-day malware.	Demonstrated a 94.50% malware detection rate using the ResNet50 model.
[27]	A study on MQTT-based attacks evaluated six machine learning techniques usingpacket-based and flow-based features.	High model accuracy of 99.04% in distinguishing MQTT based attacks.
[37]	Sapre et al. compared the KDDCup99and NSLKDD datasets to assess the performance of machine learning classifiers.	NSLKDD dataset proved superior, showing more accurate performancewith less redundancy.
[28]	Liu et al. investigated various attacks on IoT sensors and networks using the NSLKDD dataset, evaluating eleven machine learning techniques.	XGBoost was the most effective with 97% accuracy, 90.5% MCC, and 99.6%AUC.
[29]	Amouri et al. applied a two-stage process to differentiate benign and malicious nodes using linear regression on data collected by dedicated sniffers.	Demonstrated high detection rates, particularly in high velocity conditions, with over 98% effectiveness.
[30]	Fenanir et al. developed a lightweight IDS using a filter-based feature selection method and evaluated various machinelearning algorithms.	Decision tree method showed high accuracy across different datasets, reaching up to 98%.
[31]	Islam et al. evaluated both shallow and deep learning IDSs using various models and datasets.	Deep learning-based IDSs outperformed shallowones, with accuracies up to 98.79%.
[32]	Ahmad et al. proposed a feature clustering approach to address data analysis challenges using the UNSW-NB15 dataset.	Random forest achieved high accuracy in binary and multiclass classifications.
[33]	Saba et al. used a two-stage hybrid method for enhancing IDS accuracy, including a genetic algorithm for feature selection.	Attained 99.8% accuracyon the NSLKDD dataset.
[34]	Smys et al. suggested a hybrid CNN model for an IoT intrusion detection system toidentify various types of attacks.	The model was highly sensitive to threats, with a 98.6% accuracy rate.
[35]	Papafotikas et al. developed a digital system using K-means clustering to detectsuspicious activities in IoT devices.	Showcased effective identification capabilities.
[36]	An IDS utilizing a fused machine learning model was proposed, integrating three different datasets under a novel built architecture.	Promising accuracy of95.18%.

III. PROPOSED METHODOLOGY

The adopted methodology, Fig. 1 shows, is a sequential process combining Ensemble Learning to spot any irregularities and attacks in the dataset. Data preprocessing is the starting point in the process. Data should be transformed into the required format for analysis. Following this, the method branches into three different machine learning models: KNN, DT, and GB, each one is a method useful for making forecasts.

These predictions are then consolidated through two distinct voting mechanisms: soft voting and hard voting. Soft voting is about the probability of each class to give a final decision and hard voting is, the most popular prediction out of the models. Later, the proposed forecasts are evaluated using several metrics to figure out how accurate the ensemble technique is.

The technique that has been used here, blends the strengths of different machine learning models, to increase the accuracy of malicious versus benign node identification predictions in a network. This approach seeks to exploit the individualistic features every model has, combining them to make the best decision with regard to their integrity. Evaluation metrics help us to measure the efficiency of this method in one way or another.

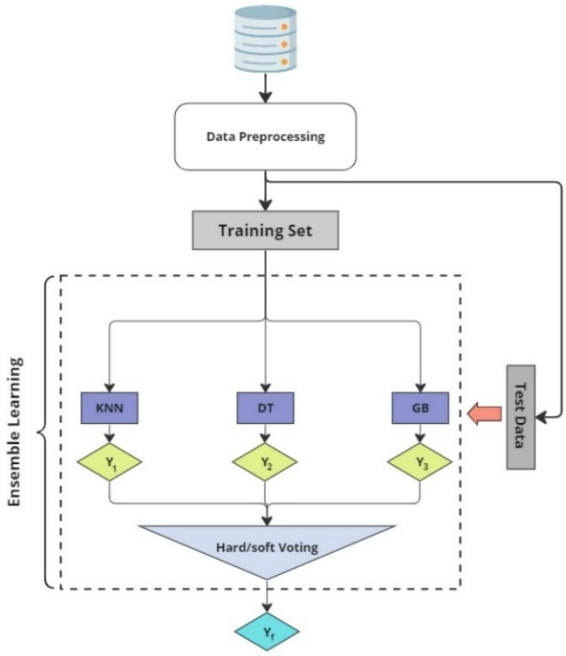

Fig. 1. Proposed Scheme

A. Dataset Overview

The analysis of DDoS attack patterns in WSNs is based on the WSNBFSF data which is a collection of network traffic captured specifically for the purpose of the experiment. It incorporates precise representations of Blackhole, Flooding, as well as Selective Forwarding attacks.

This database of datasets was created with a specified wireless network environment in a con- trolled manner. These kinds of attacks, which could happen in any IT system, are also of great relevance to the area of network security because of their ability to interfere with operations.

While meticulously preprocessing the dataset, raw network logs were reduced to structured format, which is good enough for analytical use. This consequently leads to a dataset that has 16 features and commit to 312,106 records.

The WSNBFSF dataset down comes to four types of network traffic comprising of Normal traffic and Blackhole, Flooding, and Selective Forwarding attacks data instances. This kind of a categorization moreover facilitates the more focused analysis of assault vectors, bringing to the field of cybersecurity in wireless sensor networks useful results.

B. Data Preprocessing

Our investigation commenced with the WSNBFSF dataset, encompassing data indicative of normal network behavior and that characteristic of three types of attacks: Blackhole (BH), Flooding (FF), and Selective Forwarding (SF) within wireless sensor networks. The first step in our process involved ensuring secure data transfer, followed by employing a data manipulation library for deeper processing to unravel underlying structures and patterns.

An initial visual inspection of class distribution, depicted in Fig. 2, highlighted the dataset's composition, revealing a significant class imbalance. The rationale behind addressing class imbalance is to prevent models from favoring the majority class and overlooking the minority class, which could result in biased and unreliable predictions. To address this, we applied class resampling techniques to ensure an equitable class distribution. This downsampling process, illustrated in Fig. 3, involved reducing the size of larger classes to match that of the smallest class, thus achieving uniformity across all classes. Downsampling was chosen to maintain the integrity of the original dataset without introducing synthetic variability.

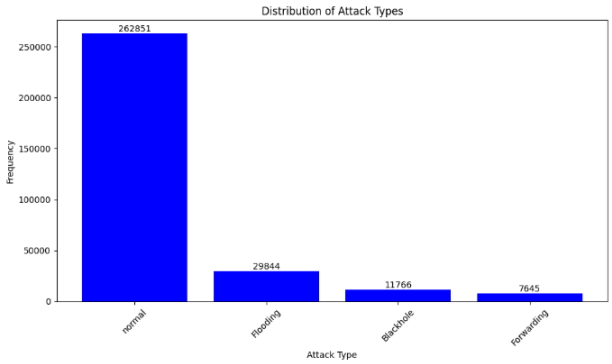

Fig. 2. Distribution of classes before downsampling

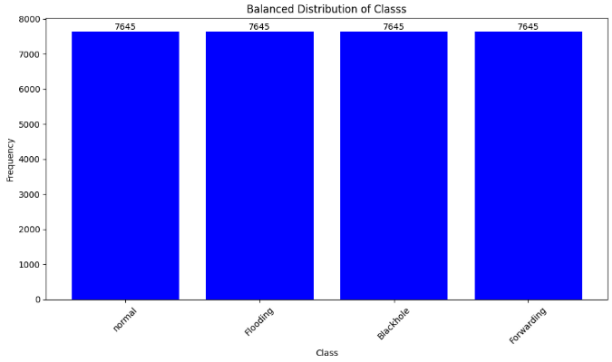

Fig. 3. Distribution of classes after downsampling

The resampling algorithm employed non-replacement sampling to create a balanced dataset, thereby mitigating the potential bias introduced by class imbalance. The feature segregation process subsequently identified and isolated the various features and classes within the dataset. To further standardize the dataset and ensure that each feature contributes equally to analytical models, we applied feature scaling using the standardization equation:

$$z = \frac{(x - \mu)}{\sigma}$$

where z is the standardized score, x is the original value, μ is the mean of the feature values, and σ is the standard deviation of the feature values. This is critical for algorithms that are sensitive to the scale of the data and facilitates faster convergence for optimization algorithms.

Finally, we partitioned the data into two subsets: one for training the models and another reserved for evaluation tasks. The split was conducted with an 80-20% distribution, allocating 80% of the data for training and 20% for testing purposes. The choice of an 80-20 split is a well-established practice, balancing the need for a substantial training dataset with the necessity of a robust test set for model validation. This standard partitioning approach, prevalent in machine learning processes, facilitates a robust evaluation of the models' performance.

C. Machine Learning Models

Our selection of the KNN, Decision Tree, and Gradient Boosting models was a deliberate decision based on the unique attributes each model brings to the table, particularly for the complexities of intrusion detection in IoT environments. KNN was specifically chosen for its operational simplicity and its effective utilization of proximity in feature space as a key to classification, a quality that serves the IoT security domain well due to the varied nature of the device profiles and the subtleties of their interactions [42]. This model's reliance on the nearest neighbor rule makes it particularly responsive to local patterns within the data, which is essential when the goal is to detect anomalous behaviors that may indicate a security breach.

Decision Trees were selected for their ability to break down the decision process into a series of straightforward questions and answers—a method that not only performs well with categorical and continuous input but also provides clear explanations for each decision, a crucial factor when actions may need to be justified or reviewed in a security context. Their structured approach to decision-making, coupled with the interpretability of the resulting tree, offers an invaluable asset in establishing transparent and trustable security protocols within IoT systems [43].

Gradient Boosting stands out due to its systematic approach to learning from mistakes. It doesn't merely learn; it evolves by intensifying its focus on instances that previous iterations have failed to classify correctly. This adaptability makes Gradient Boosting especially potent for intrusion detection, where it is essential to catch and adapt to new and sophisticated attack vectors. By combining multiple weak learners and directing efforts towards improving their mistakes, Gradient Boosting converges to a strong predictive model capable of handling the diverse and dynamic nature of IoT security threats [44].

The intrinsic features of these models, including KNN's agility, Decision Tree's clarity, and Gradient Boosting's progressive learning, make them not just individually capable but also collectively powerful when used in concert within an ensemble method. These models cover a broad spectrum of machine learning approaches—from instance-based learning to ensemble methods, ensuring a robust defense against the varied and evolving threats characteristic of IoT networks.

D. Ensemble Learning

Ensemble learning represents a sophisticated paradigm in machine learning that involves combining multiple models to achieve superior predictive performance compared to individual models operating in isolation [45][46]. This approach is particularly effective in addressing complex problems like intrusion detection in IoT environments, where the diversity of attack vectors and the subtleties of normal versus malicious activities can be challenging for any single model to handle accurately.

In this study, we implemented both soft and hard voting techniques as part of our ensemble strategy. Soft voting is used to calculate the mean probability of class predictions from various models, selecting the class with the highest mean probability as the final prediction. This method is particularly beneficial when different models deliver probabilities with varying levels of confidence, allowing for a more nuanced decision-making process that leverages the strengths of each model's predictive confidence [47].

On the other hand, hard voting employs a majority rule approach where the final class prediction is the one that receives the most votes from the individual models within the ensemble. This method assumes equal weight for each model's prediction, providing a straightforward and robust decision-making mechanism that reduces the risk of error from any single model's misjudgment. It is particularly effective in situations where a clear consensus among different models can indicate a higher confidence in the prediction outcome [47].

These ensemble techniques were not selected arbitrarily. They were chosen based on their ability to integrate and harmonize the diverse strengths of the individual models used in this research—KNN, Decision Tree, and Gradient Boosting. Each model contributes its unique perspective to the ensemble, thereby enhancing the overall decision-making process. KNN contributes localized sensitivity to the proximity of data points, Decision Trees provide interpretability and clear decision pathways, and Gradient Boosting offers increasing accuracy over iterations by focusing on previously misclassified instances.

Together, these ensemble learning methods enhance the overall robustness, accuracy, and reliability of the intrusion detection system [48]. By effectively pooling the predictions of individual models, the ensemble approach mitigates individual model biases or weaknesses, leading to improved generalization across diverse IoT network scenarios. This methodological choice is crucial for developing a scalable and effective security solution capable of adapting to the evolving landscape of cyber threats in IoT environments [49].

E. Evaluation Measures

The evaluation of machine learning models is crucial to understanding their performance and appropriateness for specific tasks. This section elaborates on the evaluation metrics briefly mentioned in the methods section, including their rationale, equations, and limitations.

1) Confusion Matrix

The confusion matrix is a fundamental tool used to visualize the performance of a classification model. It provides a matrix format that shows the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, offering a clear picture of both the successes and errors of the model [50].

2) Accuracy

Accuracy measures the proportion of total correct predictions (both positive and negative) made by the model out of all predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3) Precision

Precision assesses the accuracy of positive predictions. It is particularly useful in scenarios where the cost of a false positive is high.

$$Precision = \frac{TP}{TP + FP}$$

4) Recall

Recall (or sensitivity) measures the ability of a model to identify all relevant instances (true positives) within a dataset. This metric is crucial in situations where missing a positive instance (false negative) carries a significant penalty, such as in medical diagnosis.

$$Recall = \frac{TP}{TP + FN}$$

5) F1-Score

The F1-score is the harmonic mean of precision and recall, providing a balance between these two metrics. It is particularly useful when the class distribution is uneven and errors in one class are more significant than errors in other classes.

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The accuracy is more informative when the classes are balanced, whereas precision, recall, and the F1-score provide more insights when dealing with imbalanced datasets. Each metric provides a different perspective on model performance, addressing various aspects of prediction accuracy and error consequences.

While these metrics provide valuable insights, they also have limitations:

- Accuracy can be misleading in the presence of imbalanced classes.
- Precision and Recall may contradict each other, and improving one can often result in lowering the other, known as the precision-recall trade-off.
- The F1-score, although it balances precision and recall, might not always reflect the practical usefulness of a model, especially when a balance is not preferable.

In summary, the methodology we have embraced is a meticulous and structured sequence of steps aimed at optimizing the detection and classification of network intrusions. The cornerstone of this process is a thorough data preprocessing phase, essential for preparing the dataset for subsequent analysis [51]. Following this foundational stage, we employed a trio of machine learning models [52]—KNN, DT, and GB—each chosen for their specific attributes that are advantageous in the realm of network security. These models were then integrated into a robust ensemble learning [53] framework, utilizing both soft and hard voting techniques to enhance predictive accuracy.

The ensemble approach, a testament to the power of collective intelligence, capitalizes on the complementary strengths of the individual models [54]. It offers a sophisticated decision-making mechanism that is both diverse in perspective and unified in goal [55]. Evaluation metrics, including accuracy, precision, recall, and the F1-score, serve as the yardsticks for assessing the effectiveness of our models, ensuring that our results are not only statistically significant but also relevant and applicable in practical scenarios [56].

As we transition into the results section, we carry forward the methodological rigor and analytical insights that have characterized the preceding stages of this study. The forthcoming section will not only present the outcomes of our modeling efforts but will also contextualize these findings within the broader landscape of network intrusion detection research. It will highlight the precision of our approach in identifying threats and validate the efficacy of our models through a comparative analysis with existing methodologies, ultimately reinforcing the contribution of this research to the field of cybersecurity.

IV. Experiment Results

A. Machine Learning Models Results

In the conducted experiments, a comparative analysis of three machine learning models was performed: KNN, DT, and GB. Each model was tasked with classifying network traffic into four categories: Blackhole, Flooding, Forwarding, and Normal.

To address model generalization, cross-validation was employed during the training process of each machine learning model [57]. Cross-validation is a robust statistical technique that helps ensure the model's ability to perform well on unseen data. It involves partitioning the data into subsets, training the model on some subsets while validating on others, and rotating this process to cover the entire dataset. This method helps mitigate overfitting and provides a more reliable estimate of the model's performance in different network environments [58].

Regarding computational resources, the training and deployment of these models were conducted using Google Colab, which provides a robust cloud-based environment with substantial computational resources. Specifically, the system's specifications included 51.0 GB of system RAM,

15.0 GB of GPU RAM, and 201.2 GB of disk space, which were adequate for handling the computational demands. For instance, the KNN model achieved a mean accuracy of approximately 86.52% with a training time of 12.28 seconds, highlighting efficiency in both training speed and model accuracy.

The DT classifier demonstrated near-perfect mean accuracy, achieving approximately 99.95% with a swift training time of 7.35 seconds. This high level of accuracy, combined with low computational time, illustrates the model's efficiency and potential for real-world application.

GB, despite being a more complex ensemble technique, was trained in 184.47 seconds, indicating a longer training time which is expected due to the model's intricacy. However, it yielded a high mean accuracy of approximately 99.83%, justifying the additional computational cost. The high accuracy rates across these models, especially for DT and GB, suggest that the trade-off between computational resources and model performance is favorable.

These results affirm that the developed models are not only accurate but also computationally viable, with the potential to be scaled and deployed in real-world network environments where performance and computational efficiency are critical.

Table II presents the performance results of single models.

TABLE II. SUMMARY OF MACHINE LEARNING MODELS PERFORMANCE

Model	Accuracy	Precision	Recall	F1-Score
KNN	86.52%	Blackhole: 77% Flooding: 95% Forwarding: 90% Normal: 91%	Blackhole: 90% Flooding: 94% Forwarding: 97% Normal: 69%	Blackhole: 83% Flooding: 94% Forwarding: 93% Normal: 79%
DT	99.95%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%
GB	99.82%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%

The KNN classifier achieved an overall accuracy of 86.52%, demonstrating substantial precision and recall in its predictions. It showed commendable proficiency, particularly in identifying Flooding and Forwarding attacks with precision rates of 95% and 90%, respectively, and almost equally high recall rates. However, it was less adept at correctly classifying Normal traffic, achieving a lower precision of 91% and a notably modest recall of 69%, resulting in a lower f1-score for this class compared to the others.

The provided confusion matrix in Fig. 4 for the KNN reveals insights into the model's performance, which necessitates a more in-depth error analysis. From the matrix, we can observe a commendable performance in accurately classifying most instances of Flooding (label 2) and Forwarding attacks (label 3), as seen in the high values on the matrix diagonal for these classes. However, the substantial number of misclassifications of Normal traffic (label 0) as Flooding (label 2) and Forwarding (label 3) attacks, evident in the non-diagonal elements, warrants a closer examination.

This misclassification pattern suggests potential biases or weaknesses in the model when dealing with Normal traffic, which is of particular concern as it is imperative for a network security system to distinguish between benign and malicious traffic reliably. It is critical to conduct a comprehensive error analysis to delve into the root causes of these misclassifications. Factors such as feature selection, the representation of traffic patterns, and the model's sensitivity to the overlap between Normal traffic characteristics and those of attack traffic should be scrutinized.

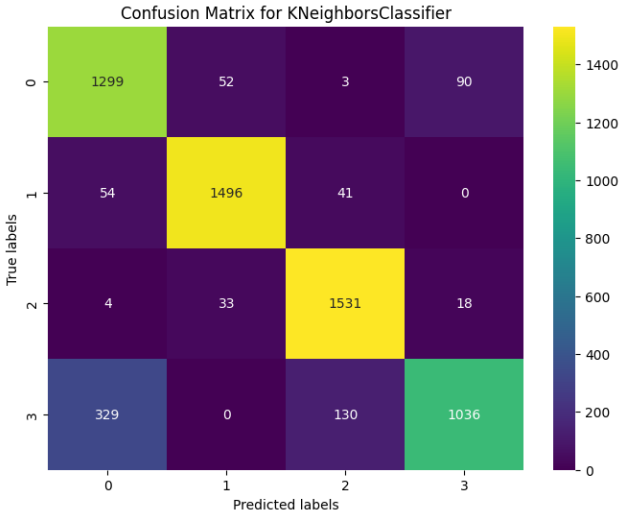

Fig. 4. Confusion matrix of KNN

The DT classifier presented a near-perfect performance, boasting an impressive ac- curacy of 99.95%. It exhibited remarkable precision and recall scores, reaching 100% in most categories. The f1-scores mirrored these results, indicating the model's exceptional ability to discern between different types of network traffic with almost flawless accuracy. Only a solitary misclassification was noted among the Blackhole category, and an equally minimal misstep was observed within the Flooding class.

The confusion matrix in Fig. 5 for the DecisionTreeClassifier provides a quantitative depiction of the model's classification performance. The matrix indicates a high number of correct predictions with 1441 instances of Normal traffic (label 0) accurately identified, and similar high values for attack types 1, 2, and 3 with 1590, 1586, and 1493 correct classifications, respectively. The relatively low misclassification numbers—only one instance of Normal traffic misclassified as attack type 1, and two instances of attack type 3 misclassified as Normal traffic—suggest that the DecisionTreeClassifier exhibits a strong discriminative ability.

However, the true merit of a classifier is not only in the high numbers of correct classifications but also in understanding and mitigating the instances it misclassifies. For instance, the three instances where the model has predicted Normal traffic instead of an attack (types 0

misclassified as 3 and vice versa) may indicate specific scenarios where the model's decision boundary is not adequately defined. A nuanced analysis of these cases could yield insights into the feature space where overlaps occur, leading to potential improvements in the model's classification rules.

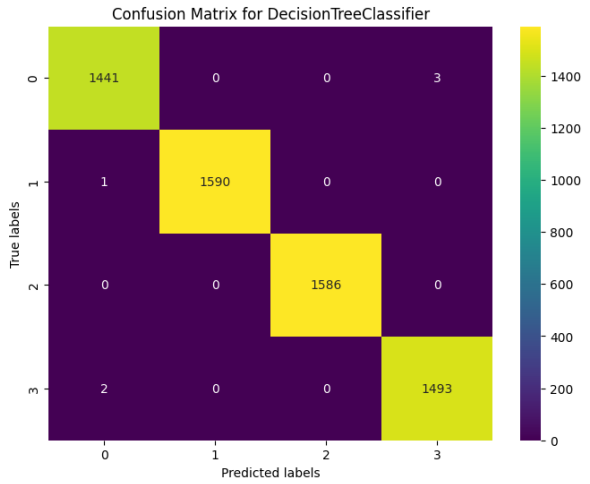

Fig. 5. Confusion matrix of DT

Gradient Boosting, a potent ensemble technique, displayed a similar prowess with an accuracy of 99.82%. It also achieved perfect precision and recall scores of 100% across the majority of the classes, with only a slight deviation in the Normal category, where it misclassified three instances.

The confusion matrix in Fig. 6 for the GB depicts a robust classification capability with high correct prediction rates across all classes—1444 for Normal traffic, 1588 for attack type 1, 1586 for attack type 2, and 1492 for attack type 3. These results underscore the model's strengths in discriminating between different traffic types. However, there are some instances of misclassification, such as Normal traffic being incorrectly labeled as attack type 1 or 3, and similarly, some instances of attack types being mistaken for Normal traffic.

While these misclassification numbers are low, they are crucial for an error analysis to ascertain the robustness and fairness of the model across diverse scenarios. The analysis should probe into these errors to discover if they are attributable to particular attributes, noise, or potential overfitting of the model to the training data. For instance, the misclassification of Normal traffic as attack types could suggest that the model may be overly sensitive to certain patterns that are not necessarily indicative of an attack.

The superiority of DT and GB was shown further with their confusion matrices which indicated a fine number of true positives and were neglected of false negatives and false positives in most classes. The identical nature by which the diagonal rows of these matrices depicted demonstrated that the stated models were applicable to the task at hand of discriminating traffic, it was almost certain that there was a negligible error in the distinction of traffic types. Hence, they will deliver exceptional performace in predictions and modeling.

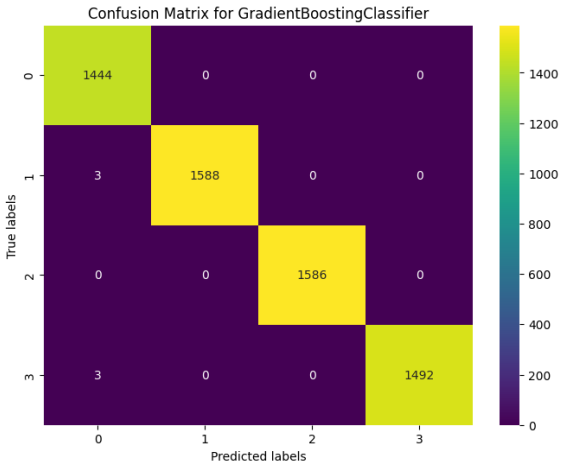

Fig. 6. Confusion matrix of GB

B. Ensemble Learning Models Results

The confusion matrix in Fig. 7 for the soft VotingClassifier reveals a highly effective consensus approach in making predictions, with the majority voting of multiple classifiers resulting in a strong performance across all classes. The matrix displays a notable accuracy in classification, with 1442 correct predictions for Normal traffic, and 1591, 1586, and 1495 for attack types 1, 2, and 3 respectively, indicating the classifier's proficiency. The negligible number of misclassifications suggests that the ensemble approach has succeeded in creating a robust model.

Nevertheless, the two instances where Normal traffic was classified as an attack type and the zero instances of attacks misclassified as Normal traffic present a critical opportunity for model improvement. Examining these misclassifications could uncover if the ensemble's decision boundaries between Normal and abnormal traffic patterns need refinement. Delving into the specifics of these instances would help to further tune the model, enhancing its predictive precision and thereby its practical applicability in real-world network security systems.

In assessing model generalization, the ensemble techniques, specifically the soft and hard voting classifiers, were subjected to rigorous cross-validation to ensure their robust performance on unseen data.

The soft voting classifier, an ensemble that combines model predictions with weighted probabilities, achieved a commendable mean accuracy of approximately 99.97% with a training time of roughly 177.6 seconds. Similarly, the hard voting classifier, which uses a majority voting scheme, also displayed outstanding performance with a mean accuracy of approximately 100% and a training time of approximately 178.1 seconds. These high accuracy levels, coupled with the thoroughness of cross-validation, indicate a strong potential for these models to generalize well to new and diverse network environments, which is critical for reliable network intrusion detection. The ensemble methods effectively aggregate the predictive strength of individual models, enhancing the overall predictive stability and reducing the likelihood of overfitting, which is a common pitfall that can impair generalization to new data.

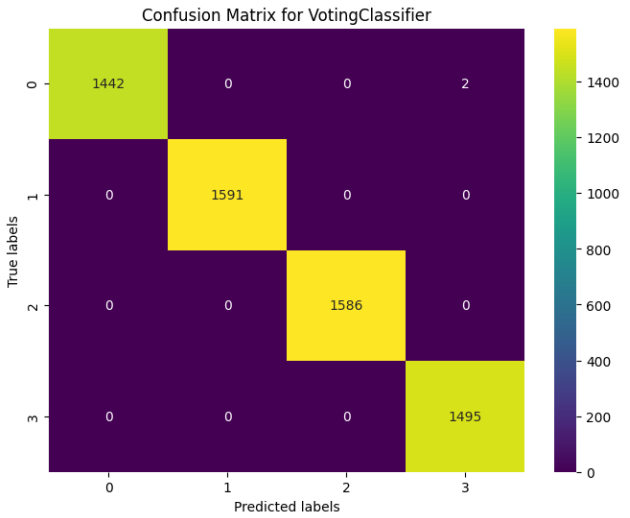

Fig. 7. Confusion matrix of soft voting

As illustrated in Table III, the soft voting classifier got a great accuracy rate up to 99.967%, with zero faultiness in both precision and recalls for all categories proved absolute consent for Blackhole, Flooding, Forwarding, and Normal traffic. The f1-score, a balanced criterion for measuring precision and recall, returned the value of 1 for all the voting types which proved the efficiency of soft voting method for recognizing the attack types with the value of 1.

The confusion matrix in Fig. 8 for the hard VotingClassifier underscores the model's high accuracy in classifying network traffic, with a substantial number of true positives for each category and minimal misclassification. However, the occurrence of misclassifications—though few—highlights the necessity for a detailed error analysis to fully comprehend the model's performance nuances. For instance, the complete absence of misclassified instances for certain attack types is encouraging, yet the few misclassified cases between Normal traffic and attacks are crucial for evaluating the model's reliability and robustness.

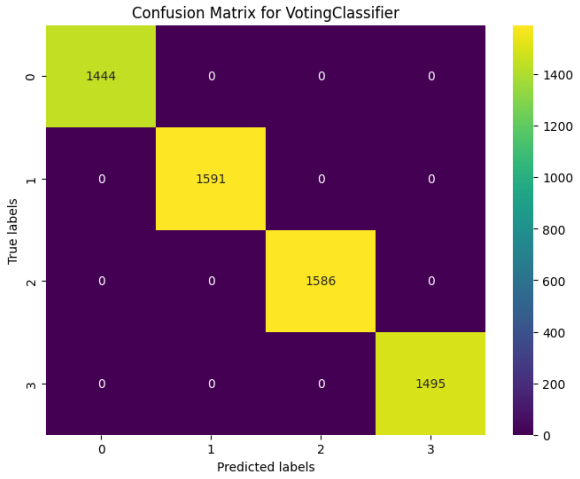

Fig. 8. Confusion matrix of hard voting

In our experiments, Hard voting classifier demonstrated its superiority by acquiring an almost perfect accuracy score of 100%. This accordingly means that each prediction made by the classifier conformed with the exact labels to the letter. High performance was the fruit of an accuracy that coalesced with a recall of 100% for all classes, making the f1 score as flawless as it could. Its outstanding concurrence with others in the ensemble shows that the hard voting classifier has accomplished a high level of accuracy beyond the individual performance of each classifier in this domain which eventually means that collective decision making can notably enhance the accuracy of predictive models in the context of network intrusion detection.

TABLE III. SUMMARY OF ENSEMBLE LEARNING MODELS PERFORMANCE

Model	Accuracy	Precision	Recall	F1-Score
Voting Classifier (Soft)	99.967%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%
Voting Classifier (Hard)	100%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%	100% Blackhole: 100% Flooding: 100% Forwarding: 100% Normal: 100%

C. Comparison Results

The predictive model's performance for network intrusion detection was evaluated by comparing the differences between standalone machine learning models and the application of ensemble method. The best accuracy was shown by the KNN model of 86.52%, which has precisions and recalls equal to 77% and 90% respectively for the Blackhole class as depicted in Table IV. Identification of the two types of attacks was a success, with a precision, in general, above 90% and a recall as high as 97% at times. However, it had a reasonable drop in efficiency when it came to norm traffic and had precision at 91% but recall at a depressing 69%.

Replacing the first DT/GB classifiers by the Decision Tree classifier and then by Gradient Boosting we observed that the DT achieved an impressive 99.95% accuracy, and the GB was behind with only 99.82% accuracy. This approach showed perfect precision and recall in all cases of any traffic category, what was confirmed by its F1-scores that demonstrated more or less perfect classifying capacity.

The ensembles methods in combination with soft and hard voting decreased the complexity involved, thus boosting the performance of the model. The soft voting classifier brought on an outstanding performance which was encircled by the near-perfect accuracy of 99.967% and preser- vation of 100% precision and recall across all classes. The hard voting classifier, although, became a symbol of perfect classification with 100% accuracy without any exceptions, referring to every case without a mistake.

The comparison is the analytical comparison which demonstrates the resilience of ensemble learning techniques, specifically, voting classifier, in the sense that it assembles the strengths of respective models for greater precision and

trustworthiness in network intrusion detection. These practical results suggest that thoughts should be given to the implementation of ensemble tech- niques within the cybersecurity frameworks in order to make the detection and classification of suspicious activities more efficient.

TABLE IV. COMPARISON OF MACHINE LEARNING AND ENSEMBLE LEARNING MODELS PERFORMANCE

Model	Accuracy	Precision	Recall	F1-Score
KNN	86.52%	88%	87%	87%
DT	99.95%	100%	100%	100%
GB	99.82%	100%	100%	100%
Voting Classifier (Soft)	99.967%	100%	100%	100%
Voting Classifier (Hard)	100%	100%	100%	100%

Table V provides a comparative landscape of the performance of various machine learning models against the hard voting classifier developed in our study. The comparison underscores the efficacy of the ensemble technique adopted in the hard voting classifier, which outperforms several notable models and techniques across a range of datasets.

The MobileNetV3 model, trained on data from the IEEE DataPort website, achieved an accuracy of 98.35%, which while commendable, is surpassed by our hard voting classifier's perfect accuracy. Similarly, the combination of Random Forest and Naïve Bayes models applied to the UNSW IoT Traces and Your Things dataset secured an accuracy of 99%, still marginally less than our model. The SVM, known for its effectiveness in high-dimensional spaces, fell short at 94% accuracy when applied to packet data, highlighting the complexity and challenge present in network intrusion detection tasks.

Further, the Xception-BiGRU architecture, despite its advanced deep learning capabilities, achieved an accuracy of 95.6% on a Kaggle dataset, indicating the difficulty of achieving high accuracy in diverse datasets. The XGBoost model, applied to the well-known NSL-KDD dataset, achieved an accuracy of 97%, which suggests its proficiency in feature engineering and classification tasks, yet it does not reach the benchmark set by our model.

Our hard voting classifier, evaluated on the WSNBFSF dataset, attained a 100% accuracy rate, which represents a significant advancement in the field of intrusion detection systems. This suggests that the ensemble of models within the hard voting classifier is well-suited for the intricacies and variability inherent in network security datasets. Perfect accuracy indicates that the classifier has effectively leveraged the strengths of multiple models to achieve a level of predictive performance that is unmatched by the individual models compared here.

The results in Table V not only validate the superiority of our hard voting classifier in accuracy but also hint at its robustness and adaptability across different types of network intrusion scenarios. This comprehensive performance metric illustrates the potential for the hard voting classifier to serve as a benchmark for future intrusion detection system developments and evaluations.

TABLE V. COMPARISON OF OUR HARD VOTING MODELS PERFORMANCE WITH EXISTING APPROACHES

Ref.	Model	Dataset	Accuracy
[21]	MobileNetV3	Data from IEEE DataPort website	98.35%
[22]	Random Forest and Naïve Bayes	UNSW IoT Traces, and Your Things dataset	99%
[23]	SVM	packet data	94%
[25]	Xception-BiGRU	Dataset from Kaggle	95.6%
[28]	XGBoost	NSL-KDD	97%
Our Model	**Voting Classifier (Hard)**	**WSNBFSF**	**100%**

The findings from the experiments with the ensemble learning models, particularly the hard voting classifier, have several practical implications for the design and implementation of network intrusion detection systems (NIDS). The near-perfect performance of the ensemble models indicates that combining multiple algorithms can lead to more accurate detection of various types of network intrusions, which is crucial in the rapidly evolving landscape of cybersecurity threats.

In practical terms, the integration of such ensemble methods can enhance the ability of NIDS to differentiate between benign and malicious traffic with greater precision, reducing the number of false positives and negatives. This accuracy is particularly valuable in large-scale networks where the volume of traffic can make manual review of alerts impractical. The high accuracy of the ensemble models can also increase trust in automated security measures, which is essential for their adoption in critical infrastructure.

Moreover, the application of ensemble learning models can potentially improve the speed and efficiency of threat detection. By leveraging the strengths of individual classifiers and minimizing their weaknesses, these models can operate effectively even when faced with large and complex datasets. This capability is essential for real-time intrusion detection where the speed of response can mitigate the impact of attacks.

V. CONCLUSION

The comprehensive analysis of machine learning and ensemble learning models presented in this study offers an in-depth evaluation of their performance in network intrusion detection. Our investigation began with the application of individual machine learning models—KNN, DT, and GB—each demonstrating competencies in classifying network traffic and identifying threats with varying degrees of accuracy. The KNN model, while being the least precise, was notably effective in detecting Flooding and Forwarding attacks, underscoring the value of simplicity and local inference in model construction. DT and GB, on the other hand, stood out with their exceptional accuracy, reflecting the robustness of tree-based algorithms in managing the intricate patterns often present within cybersecurity data.

Our exploration extended into ensemble learning techniques, where soft and hard voting classifiers were examined. The results highlighted the advantage of combining multiple models, with both voting techniques achieving nearly flawless accuracy—soft voting with 99.967% and hard voting with 100%. This amalgamation not only harnessed the strengths of each contributing model but also diminished their individual weaknesses, resulting in a predictive system that was both reliable and robust.

The ensemble models were rigorously evaluated against previous approaches, as indicated in Table V, where they demonstrated superior accuracy. Notably, the hard voting classifier's perfect accuracy benchmarked a significant advancement over other cited methods, illustrating the potential of collective decision-making in enhancing the precision of network intrusion detection systems.

The practical implications of these findings are significant for the field of cybersecurity. The ability to detect and classify network intrusions accurately is paramount, and this research underscores the promise of blending individual and ensemble learning approaches. However, there is an understanding that model performance in controlled experiments must translate into real-world effectiveness. Therefore, we underscore the necessity for continuous validation of these models against emerging threats in dynamic network environments.

Future research should not only continue to refine the accuracy and efficiency of these models but also address critical areas such as scalability, resource efficiency, and model interpretability. Scalability and efficiency are crucial for deploying these systems within large-scale and diverse network infrastructures. Meanwhile, interpretability remains a cornerstone for trust and accountability in automated decision-making systems, particularly in a domain as sensitive as network security [59].

In light of the evolving nature of cyber threats, future studies must also focus on the adaptability of intrusion detection systems. The development of models that can evolve with the threat landscape and detect new types of attacks in real-time will be a critical area of advancement. Further exploration into sophisticated deep learning architectures [60] and novel ensemble techniques that can operate within the constraints of real-time detection and with limited computational resources will be essential.

This study lays the groundwork for a robust and sophisticated approach to network intrusion detection, with the aspiration that future advancements will build upon this foundation to achieve even greater levels of security and reliability in the cyber domain.

REFERENCES

[1] J. H. Park, "Advances in Future Internet and the Industrial Internet of Things," *Symmetry*, vol. 11, no. 2, p. 244, 2019, doi: 10.3390/sym11020244.

[2] C. Tankard, "Big data security," *Network Security*, vol. 2012, no. 7, pp. 5–8, 2012, doi: 10.1016/S1353-4858(12)70063-6.

[3] M. A. Khan, M. R. Karim, and Y. Kim, "A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network," *Symmetry*, vol. 11, no. 4, p. 583, 2019, doi: 10.3390/sym11040583.

[4] A. Meryem and B. E. L. Ouahidi, "Hybrid intrusion detection system using machine learning," *Network Security*, vol. 2020, no. 5, pp. 8–19, 2020, doi: 10.1016/S1353-4858(20)30056-8.

[5] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *J. Big Data*, vol. 7, no. 1, pp. 41–29, 2020, doi: 10.1186/s40537-020-00318-5.

[6] Q.-V. Dang, "Studying Machine Learning Techniques for Intrusion Detection Systems. Future Data and Security Engineering," *Springer*, 2019, doi: 10.1007/978-3-030-35653-8_28.

[7] A. Muñoz, A. Maña, and J. González, "Dynamic Security Properties Monitoring Architecture for Cloud Computing," *Security Engineering for Cloud Computing: Approaches and Tools*, 2013, doi: 10.4018/978-1-4666-2125-1.ch001.

[8] T. Rupa Devi and S. Badugu, "A Review on Network Intrusion Detection System Using Machine Learning," *Advances in Decision Sciences, Image Processing, Security and Computer Vision*, 2019, doi: 10.1007/978-3-030-24318-0_69.

[9] K. S. Bhosale, M. Nenova, and G. Iliev, "Intrusion Detection in Communication Networks Using Different Classifiers," *Techno-Societal 2018*, 2019, doi: 10.1007/978-3-030-16962-6_3.

[10] H. Liu and B. Lang, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, 2019, doi: 10.3390/app9204396.

[11] A. S. Ashoor and S. Gore, "Importance of intrusion detection system (IDS)," *International Journal of Scientific and Engineering Research*, vol. 2, no. 1, pp. 1-4, 2011.

[12] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artif. Intell. Rev*., vol. 51, no. 3, pp. 403–443, 2019, doi: 10.1007/s10462-017-9567-1.

[13] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One Class Support Vector Machine," *Electronics*, vol. 9, no. 1, p. 173, 2020, doi: 10.3390/electronics9010173.

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

[15] R. Lyu, M. He, Y. Zhang, L. Jin, and X. Wang, "Network Intrusion Detection Based on an Efficient Neural Architecture Search," *Symmetry*, vol. 13, no. 8, p. 1453, 2021, doi: 10.3390/sym13081453.

[16] J. Song, H. Takakura, Y. Okabe, and K. Nakao, "Toward a more practical unsupervised anomaly detection system," *Inform. Sci.*, vol. 231, pp. 4–14, 2013, doi: 10.1016/j.ins.2011.08.011.

[17] I. Ullah and Q. H. Mahmoud, "A filter-based feature selection model for anomaly-based intrusion detection systems," *2017 IEEE International Conference on Big Data (Big Data)*, pp. 2151-2159, 2017, doi: 10.1109/BigData.2017.8258163.

[18] Q. R. S. Fitni and K. Ramli, "Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems," *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 118-124, 2020, doi: 10.1109/IAICT50021.2020.9172014.

[19] T. Vaiyapuri and A. Binbusayyis, "Application of deep autoencoder as an one-class classifier for unsupervised network intrusion detection: a comparative evaluation," *PeerJ Comput. Sci.*, vol. 6, p. e327, 2020, doi: 10.7717/peerj-cs.327.

[20] S. K. Wagh and S. R. Kolhe, "Effective semi-supervised approach towards intrusion detection system using machine learning techniques," *International Journal of Electronic Security and Digital Forensics*, vol. 7, no. 3, pp. 290-304, 2015.

[21] J. R. Rose, M. Swann, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion Detection using Network Traffic Profiling and Machine Learning for IoT," *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, pp. 409-415, 2021, doi: 10.1109/NetSoft51509.2021.9492685.

[22] Z. Ali, F. Hussain, S. Ghazanfar, M. Husnain, S. Zahid, and G. A. Shah, "A Generic Machine Learning Approach for IoT Device Identification," *2021 International Conference on Cyber Warfare and*

Security (ICCWS), pp. 118-123, 2021, doi: 10.1109/ICCWS53234.2021.9702983.

[23] R. El-Sayed, A. El-Ghamry, T. Gaber, and A. E. Hassanien, "Zero-Day Malware Classification Using Deep Features with Support Vector Machines," *2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 311-317, 2021, doi: 10.1109/ICICIS52592.2021.9694256.

[24] K.-H. Le, M.-H. Nguyen, T.-D. Tran, and N.-D. Tran, "IMIDS: An Intelligent Intrusion Detection System against Cyber Threats in IoT," *Electronics*, vol. 11, no. 4, p. 524, 2022, doi: 10.3390/electronics11040524.

[25] H. Joo, H. Choi, C. Yun, and M. Cheon, "Efficient network traffic classification and visualizing abnormal part via Hybrid Deep Learning Approach: Xception+ bidirectional gru," *Global Journal of Computer Science and Technology*, vol. 21, pp. 1-10, 2021.

[26] G. Bendiab, S. Shiaeles, A. Alruban and N. Kolokotronis, "IoT Malware Network Traffic Classification using Visual Representation and Deep Learning," *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pp. 444-449, 2020, doi: 10.1109/NetSoft48620.2020.9165381.

[27] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, X. Bellekens, "Machine learning based IoT intrusion detection system: An MQTT case study (MQTT-IoT-IDS2020 dataset)," in *International Networking Conference*, pp. 73-84, 2020, doi: 10.1007/978-3-030-64758-2_6.

[28] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *Proceedings of the 2nd ACM workshop on wireless security and machine learning*, pp. 25-30, 2020.

[29] A. Amouri, V. T. Alaparthy, and S. D. Morgera, "A Machine Learning Based Intrusion Detection System for Mobile Internet of Things," *Sensors*, vol. 20, no. 2, p. 461, 2020, doi: 10.3390/s20020461.

[30] S. Fenanir, F. Semchedine, and A. Baadache, "A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things," *Revue d'Intelligence Artificielle*, vol. 33, no. 3, 2019.

[31] N. Islam *et al*., "Towards Machine Learning Based Intrusion Detection in IoT Networks," *Computers, Materials & Continua*, vol. 69, no. 2, 2021.

[32] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *J. Wireless Com. Network.*, vol. 2021, no. 1, pp. 10–23, 2021, doi: 10.1186/s13638-021-01893-8.

[33] T. Saba, T. Sadad, A. Rehman, Z. Mehmood, and Q. Javaid, "Intrusion Detection System Through Advance Machine Learning for the Internet of Things Networks," *IT Prof.*, vol. 23, no. 2, pp. 58–64, 2021, doi: 10.1109/MITP.2020.2992710.

[34] S. Smys, A. Basar, and H. Wang, "Hybrid intrusion detection system for internet of things (IoT)," *Journal of ISMAC*, vol. 2, no. 4, pp. 190-199, 2020.

[35] S. Papafotikas and A. Kakarountas, "A Machine-Learning Clustering Approach for Intrusion Detection to IoT Devices," *2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1-6, 2019, doi: 10.1109/SEEDA-CECNSM.2019.8908520.

[36] M. S. Farooq, S. Abbas, K. Sultan, M. A. Atta-ur-Rahman, M. A. Khan, and A. Mosavi, "A Fused Machine Learning Approach for Intrusion Detection System," *Tech Science Press*, vol. 74, no. 2, pp. 2607 – 2623, 2023.

[37] S. Sapre, P. Ahmadi, and K. Islam, "A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms," *arXiv preprint arXiv:1912.13204*, 2019.

[38] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a Lightweight Intrusion Detection System for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019, doi: 10.1109/ACCESS.2019.2907965.

[39] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things*, vol. 7, p. 100059, 2019, doi: 10.1016/j.iot.2019.100059.

[40] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *Int. J. Inf. Secur.*, vol. 22, no. 5, pp. 1125–1162, 2023, doi: 10.1007/s10207-023-00682-2.

[41] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecur.*, vol. 4, no. 1, pp. 1–27, 2021, doi: 10.1186/s42400-021-00077-7.

[42] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimed. Tools Appl.*, vol. 82, no. 15, pp. 23615–23633, 2023, doi: 10.1007/s11042-023-14795-2.

[43] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "An improved anomaly detection model for IoT security using decision tree and gradient boosting," *J. Supercomput.*, vol. 79, no. 3, pp. 3392–3411, 2023, doi: 10.1007/s11227-022-04783-y.

[44] M. Douiba, S. Benkirane, A. Guezzaz, and M. Azrour, "Anomaly detection model based on gradient boosting and decision tree for IoT environments security," *J. Reliable Intell. Environ.*, vol. 9, no. 4, pp. 421–432, 2023, doi: 10.1007/s40860-022-00184-3.

[45] D. Mishra, B. Naik, P. B. Dash, and J. Nayak, "SEM: Stacking Ensemble Meta-Learning for IOT Security Framework," *Arab. J. Sci. Eng.*, vol. 46, no. 4, pp. 3531–3548, 2021, doi: 10.1007/s13369-020-05187-x.

[46] A. Odeh and A. Abu Taleb, "Ensemble-Based Deep Learning Models for Enhancing IoT Intrusion Detection," *Appl. Sci.*, vol. 13, no. 21, p. 11985, 2023, doi: 10.3390/app132111985.

[47] R. Majeed, N. A. Abdullah, M. Faheem Mushtaq, M. Umer, and M. Nappi, "Intelligent Cyber-Security System for IoT-Aided Drones Using Voting Classifier," *Electronics*, vol. 10, no. 23, p. 2926, 2021, doi: 10.3390/electronics10232926.

[48] A. Aldhaheri, F. Alwahedi, M. A. Ferrag, and A. Battah, "Deep learning for cyber threat detection in IoT networks: A review," *Internet of Things and Cyber-Physical Systems*, vol. 4, pp. 110-128, 2024.

[49] A. M. Banaamah and I. Ahmad, "Intrusion detection in iot using deep learning," *Sensors*, vol. 22, no. 21, p. 8417, 2022.

[50] I. Düntsch and G. Gediga, "Confusion matrices and rough set data analysis," in *Journal of Physics: Conference Series*, vol. 1229, no. 1, p. 012055, 2019.

[51] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, 2022.

[52] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, pp. 160–21, 2021, doi: 10.1007/s42979-021-00592-x.

[53] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[54] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: A review," *Engineering Applications of Artificial Intelligence*, vol. 115, p. 105151, 2022.

[55] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Front. Comput. Sci.*, vol. 14, no. 2, pp. 241–258, 2020, doi: 10.1007/s11704-019-8208-z.

[56] G. Naidu, T. Zuva, and E. M. Sibanda, "A Review of Evaluation Metrics in Machine Learning Algorithms," in *Computer Science On-line Conference*, pp. 15-25, 2023.

[57] P. Barbiero, G. Squillero, and A. Tonda, "Modeling generalization in machine learning: A methodological and computational study," *arXiv preprint arXiv:2006.15680*. 2020.

[58] T. Leinonen, D. Wong, A. Wahab, R. Nadarajah, M. Kaisti, and A. Airola, "Empirical investigation of multi-source cross-validation in clinical machine learning," *arXiv preprint arXiv:2403.15012*, 2024.

[59] D. Minh, H. X. Wang, Y. F. Li, and T. N. Nguyen, "Explainable artificial intelligence: a comprehensive review," *Artif. Intell. Rev.*, vol. 55, no. 5, pp. 3503–3568, 2022, doi: 10.1007/s10462-021-10088-y.

[60] S. Dong, P. Wang, and K. Abbas, "A survey on deep learning and its applications," *Computer Science Review*, vol. 40, p. 100379, 2021, doi: 10.1016/j.cosrev.2021.100379.