Iraqi Journal for Computer Science and Mathematics

Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
e-ISSN: 2788-7421 p-ISSN: 2958-0544

Intelligent Detection of Distributed Denial of Service Attacks: A Supervised Machine Learning and Ensemble Approach

Mustafa S. Ibrahim Alsumaidaie [1], Khattab M. Ali Alheeti [1], Abdul Kareem Alaloosy [1]

[1]Department of Computer Science, University of Anbar Ramadi, IRAQ

*Corresponding Author: Mustafa S. Ibrahim Alsumaidaie

DOI: https://doi.org/ 10.52866/ijcsm.2023.02.03.002
Received March 2023; Accepted May 2023 ; Available online June 2023

ABSTRACT: The rapid growth of The Internet of Things (IoT), intelligent devices, and 5G networks has increased the prevalence and complexity of Distributed Denial of Service (DDoS) attacks, posing significant challenges to cybersecurity. The objective of this research is to develop an effective method to detect and prevent DDoS attacks, thereby safeguarding communication systems from such threats. The proposed "Intelligent Distributed Denial of Service Attacks Detection (IDDOSAD) Approach" utilizes supervised machine learning algorithms, including Random Forests, Decision Trees, K-Nearest Neighbor, XGBoost, and Support Vector Machine, along with ensemble learning to enhance detection accuracy. The model development process consists of data collection, pre-processing, splitting into training and testing sets, selecting prediction models, and evaluating their performance. Evaluated on a dataset of 11,423 instances, the proposed approach demonstrated promising results, with accuracy ranging from 92% to 100% for the time series dataset. In conclusion, the proposed approach effectively detects and mitigates DDoS attacks, offering a reliable solution to protect communication systems against this growing cybersecurity threat.

Keywords: Decision tree; Random Forest; DDoS attack; Ensemble Learning; Machine learning; K-Nearest Neighbor; Cyber Security; XGBoost.

1. INTRODUCTION

Over two decades ago, the inception of the World Wide Web (WWW) took place, unleashing a global force that has tremendously impacted people's lives through the abundance of diverse web applications that present billions of web pages on a daily basis [1]. Web applications have evolved significantly over time, with various forms including online retail transactions, webmail, Amazon, online auctions, and instant messaging services. These have made them asynchronous, interactive, and dynamic. Given its critical international significance, it has become imperative to ensure that web applications are secure, precise, and of superior quality [2]. The DOS and DDOS attacks represent one of the most pivotal predicaments of the contemporary digital realm. This significant threat, which has been present since the internet was first created, is a serious problem that has not yet been fully addressed by the current TCP/IP protocol. The primary objective of DDOS attacks is to render the system inoperable. Despite the substantial advancements in information security technologies, these assaults persist in challenging the efficacy of extant defense systems [3]. The Internet of Things (IoT) represents the combination of electronic devices equipped with sensors, software, processing capacity, networking capabilities, and other technologies. These devices can operate, connect, and communicate independently with other devices and systems through the Internet, thereby enhancing and enabling services across various domains. In smart cities, industrial automation, healthcare, and agriculture, the IoT promises to augment comprehensive sensing and efficient resource utilization. However, the rapid proliferation and vastness of the network, along with the criticality of data generated by these devices, imparts significant privacy and security concerns for several IoT applications.

Furthermore, the limitations of IoT devices in terms of energy, computational power, and memory worsen security and design challenges. New, efficient protocols and security approaches that meet the requirements of current and future devices are essential to ensure secure data transmission among a large number of devices. In this context, DDoS attacks pose a significant security risk in IoT systems [4]. According to the 2018-2023 Cisco Annual Internet Report, a DDoS attack occurs when multiple systems overload the bandwidth or resources of a targeted system, typically a web server or servers. Such an attack often happens when multiple compromised systems flood the targeted system with traffic. DDoS attacks are the primary threat perceived by most service providers. Infrastructure outages also remain a

problem, with over half of the operators facing this issue. Amplification attackers, who have tools to carry out DDoS attacks, exploit vulnerabilities in the network and computing resources. Security vendors are working diligently to make these attacks financially unfeasible for cybercriminals.

The magnitude and frequency of DDoS attacks are escalating:

• Peak attack volume increased by 63% Y/Y.

• There was a 776% increase in attacks between 100 Gbps and 400 Gbps year-over-year. This significant increase in attack incidents emphasises the growing threat environment that organisations must deal with in the digital age.

• Global frequency of DDoS attacks rose by 39% Y/Y.

• 23% of the attacks were larger than 1 Gbps.

• The average size of DDoS attacks is 1 Gbps, which is sufficient to completely disable most organizations.

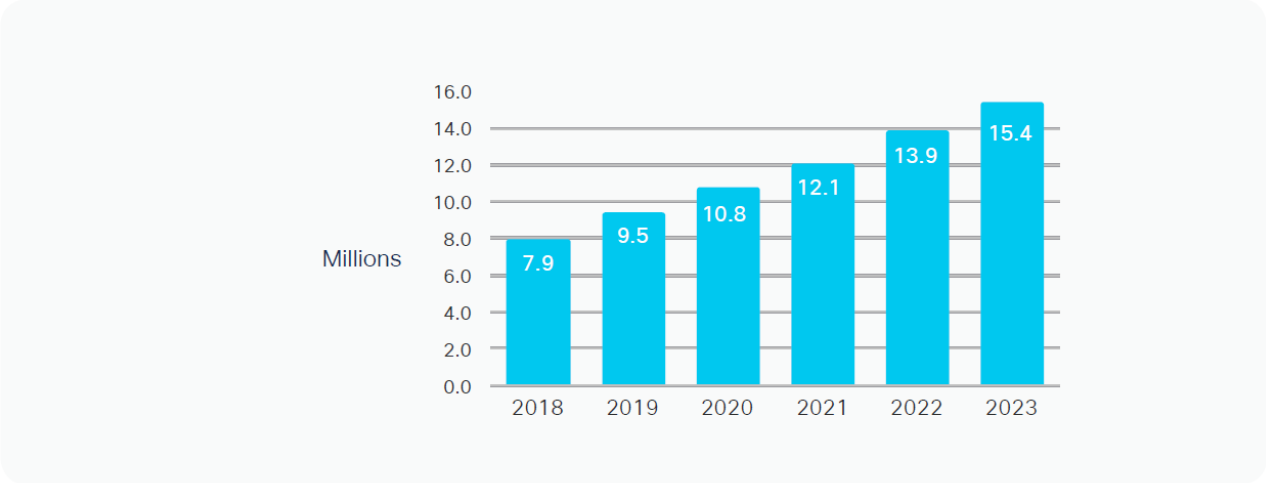

FIGURE 1. - Global DDoS Attack Trends: Predicted Doubling to 15.4 million Attacks by 2023 [4].

Figure 1 illustrates the alarming growth of DDoS attacks on a global scale. Based on projections, it is estimated that the number of DDoS attacks will double to a staggering 15.4 million by the year 2023. This sharp increase in attacks poses a significant concern for cybersecurity professionals and organizations worldwide. It highlights the urgent need for robust defensive measures to safeguard against these malicious activities. Understanding the escalating trend of DDoS attacks is crucial for developing effective countermeasures and fortifying network infrastructures to protect against this mounting threat[5].

Machine learning techniques have been widely used for detecting DDoS attacks due to their ability to process large amounts of data and identify patterns that may be indicative of an attack [6]. Decision tree and Random Forest are two popular machine learning algorithms used in DDoS attack detection [7]. Decision tree is a simple and interpretable algorithm that builds a tree-like model of decisions based on a set of rules, while Random Forest is an ensemble learning method that combines multiple decision trees to improve performance [8].

In recent years, DDoS attacks have become a significant threat to the security of networks and systems. Various machine learning algorithms have been applied to detect these attacks, as seen in the literature review. However, the novelty and relevance of our research lie in the following aspects:

1. First use of time-series dataset: Unlike previous studies, our research is the first to utilize a time-series dataset for detecting DDoS attacks. This approach allows us to capture the temporal patterns of network traffic, which can lead to better detection and prediction of DDoS attacks.
2. Data labelling and balancing: In our study, Special attention is paid to data labelling and balancing, ensuring that the dataset used for training and evaluation is well-represented and unbiased. This aspect is crucial in improving the performance of the machine learning algorithms, as it reduces the chances of overfitting and ensures model on unseen data.
3. Multiple ML techniques and ensemble voting: This study not only explores various machine learning algorithms but also employs ensemble voting techniques to improve the accuracy and robustness of the detection system. By combining the strengths of different algorithms and the ensemble voting method, a better overall performance in detecting DDoS attacks is obtained.

These a fore mentioned aspects contribute to a more accurate and reliable DDoS attack detection system, which can be crucial in maintaining the security and stability of networks and systems in today's increasingly connected world.

The remainder of this manuscript is organized as follows: Section 2 presents a literature review of relevant studies on the detection of DDoS attacks using machine learning methods. Section 3 provides an overview of the dataset used in this study and outlines the approach employed. Section 4 exhibits the outcomes and scrutinizes the performance of the Decision tree and Random Forest algorithms. Finally, Section 5 concludes the paper.

2. LITERATURE REVIEW

DDoS attacks have become increasingly common in recent years, and various techniques have been proposed to detect and mitigate them. Machine learning algorithms have shown promise in detecting DDoS attacks due to their ability to process large amounts of data and identify patterns that may be indicative of an attack. This section presents previous studies related to the detection of DDoS attacks using machine learning techniques in the years from 2018 to 2022, which are explained as follows:

A.A. Abdulrahman and M. K. Ibrahem [9] Suggested a novel architecture of a host-based intrusion detection system (IDS) for detecting DDoS attacks. This IDS employs dynamic and periodic evaluation of intruder groups in relation to their neighbours, thereby enabling intelligent detection of intrusions. The efficacy of the proposed system was evaluated using the CICIDS 2017 dataset, which includes both benign and DDoS attack network flows that meet certifiable criteria and are openly accessible. A comprehensive range of machine learning algorithms and network traffic features was examined to identify the most effective features for detecting the targeted attack classes. The results of the study indicate that the Random Forest (RF) and C5.0 classifiers perform better than the others, achieving an average accuracy of 86.80% and 86.45%, respectively. In addition, these classifiers are highly accurate, with an approximate 99% precision rate.

N. Bindraa and M. Sooda [10] proposed a methodology to determine the optimal supervised machine learning algorithm for detecting DDoS attacks and to evaluate the accuracy of these systems when trained on real-life datasets. the classification accuracy results demonstrated that the logistic regression algorithm achieved an accuracy of 0.82, the K nearest neighbour algorithm achieved an accuracy of 0.94, and the random forest algorithm achieved an accuracy of 0.96.

T. A. Tuan et al. [11] proposed an empirical analysis of machine learning techniques for detecting Botnet DDoS attacks. The evaluation was performed on two well-known and publicly available datasets for Botnet DDoS attack detection, namely, UNBS-NB 15 and KDD99. The commonly used machine learning techniques, including Support Vector Machine, Artificial Neural Network, Naive Bayes, Decision Tree, and Unsupervised Learning (USML), were investigated for their accuracy, false alarm rate (FAR), sensitivity, specificity, false positive rate (FPR), area under the curve, and Matthew's correlation coefficient (MCC) on the datasets. The empirical findings indicated that the performance of the KDD99 dataset was superior to that of the UNBS-NB 15 dataset. This validation of machine learning techniques in detecting Botnet DDoS attacks held significant implications for computer security and other related fields.

G. Usha et al. [12] presented a sophisticated system aimed at efficient detection of Distributed Denial of Service (DDoS) attacks within a network. Previous studies have extensively employed machine learning algorithms for intrusion detection and classification of attack types in contrast to other techniques and Intrusion Detection Systems (IDS). The proposed system employed a variety of machine learning algorithms, including Extreme Gradient Boosting, K-Nearest Neighbor, Stochastic Gradient Descent, and Naive Bayes, along with a deep learning architecture in the form of a Convolutional Neural Network (CNN) for attack identification and classification. The findings indicated that the Extreme Gradient Boosting (XGBoost) algorithm achieved the highest level of accuracy among the methods considered.

A. Najar and S. M. Naik [13] various machine learning techniques have been presented to identify and classify DDoS attack packets. Among these techniques, Support Vector Machine, Random Forest, multi-layer perceptron (MLP), and K-Nearest Neighbour (KNN) were applied and showed promising results. In particular, Random Forest showed an impressive accuracy of 99.13% on both the training and validation datasets and 97% on the complete tested dataset. On the other hand, MLP demonstrated an accuracy of 97.96% on the training data and 98.53% on the validation data, however, its accuracy dropped significantly to 74% on the full tested dataset **[13]**. Table 1. represents a summary of related work:

Table 1. - Related Work.

Ref.	Algorithms	Accuracy

[8]	Random Forest	96.13%
	SVM	82.35%
	KNN	94.36%
[9]	Decision Tree	77%
	Random Forest	86%
[10]		
[12]	KNN	87%
	XGBoost	89%
[12]	SVM	84.32%
	Random Forest	86.80%
	SVM	79.88%

Based on a review of previous studies, several research gaps have been identified that offer potential areas for further investigation. One such gap is the inconsistency in datasets utilized across the studies. Different studies employ different datasets, such as CICIDS 2017, UNBS-NB 15, KDD99, and various real-life datasets. This inconsistency could have an impact on the performance of machine learning models, given that each dataset possesses unique characteristics. It would be beneficial to standardize the dataset for DDoS attack detection research to enhance the comparability of results across studies. Moreover, a cross-study analysis that compares the performances of different models on the same datasets could provide a more accurate picture of each model's strengths and weaknesses.

A second significant gap lies in the performance metrics used in these studies. The metrics deployed vary, with some studies focusing on accuracy, precision, and the false alarm rate while overlooking other important metrics. A more comprehensive suite of metrics, including recall, F1-score, and the area under the ROC curve (AUROC), could offer deeper insights. These additional metrics allow for an understanding of the trade-off between precision and recall and a more holistic assessment of overall model performance.

Furthermore, an additional important factor not explicitly addressed in these studies is the potential role of time-series data in detecting DDoS attacks. Network data is intrinsically sequential and temporal. Leveraging the time-series nature of this data could potentially improve the detection capabilities of these systems. This could involve using time-series analysis methods. Thus, an examination of the effectiveness of time-series data and corresponding analytical models could offer another promising avenue for research in the field of DDoS attack detection.

3. METHODS

The process of building a machine learning model encompasses several stages. The initial stage involves collecting relevant data to the problem at hand. Once the data has been gathered, the next stage involves pre-processing the data and identifying relevant features. Subsequently, a machine learning algorithm is applied to construct a model that addresses the problem. The final stage involves evaluating the performance of the model, including its accuracy and efficiency in making predictions.

This section describes the proposed system. The system will be based on supervised machine learning algorithms, specifically the Random Forest Classifier, Decision Tree Classifier, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and XGBoost.

3.1. THE GENERAL STRUCTURE OF THE SYSTEM

In this section, the structure of the proposed system is presented, and the steps for its implementation are explained in Figure 2.

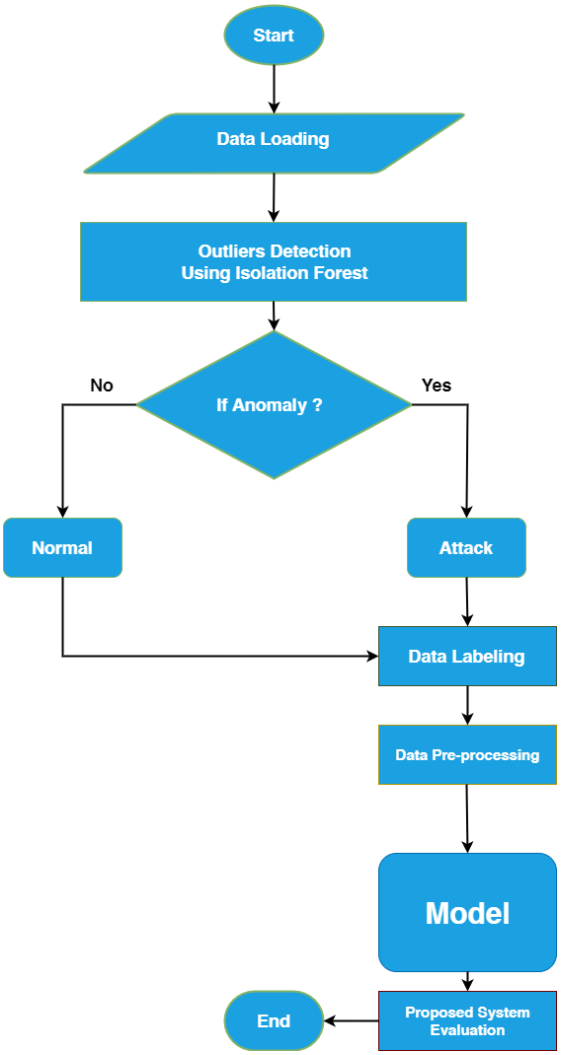

FIGURE 2. - The Flow Chart of the System.

Figure 2 in provides a comprehensive visual representation of the various stages involved in our system. Let's delve a bit deeper into each of these phases to provide a more thorough understanding:

1. *Data Loading*: This is the initial phase where the dataset is loaded into the system. It is a crucial step as it prepares the raw data for the subsequent stages of the process.

2. *Outlier Detection Using Isolation Forest*: After the data is loaded, the next phase involves outlier detection. For this, An Isolation Forest, which is an unsupervised learning algorithm for anomaly detection, is utilized. It works on the principle of isolating anomalies, as opposed to the standard practice of profiling normal points. Outliers or anomalies in a dataset represent instances that deviate significantly from other instances, making them susceptible to being classified as outliers by the algorithm[14].

3. *Data Pre-processing and Labelling*: Upon the completion of outlier detection, the dataset undergoes preprocessing and labeling. In this phase, Data is prepared for modelling by applying several pre-processing techniques, such as normalization and transformation, to remove noise and inconsistencies. Additionally, labelling is performed, a process that involves assigning a meaningful and informative label to each data point. Labelling is an essential step as it facilitates the supervised learning process, enabling the model to learn from the labelled data.

4. *Model Application*: Following the preprocessing and labeling of the data, Next, the model application phase is approached. Here, the prepared dataset is fed into our selected machine learning models. These models are trained on the data, learning to recognize patterns and make predictions or decisions without being explicitly programmed to perform the task.

5. *Classification of Time-Series Data*: The final phase of our system involves the classification of time-series data. Using the trained models Each time-series instance is classified into either 'normal' or 'malicious'. This classification

aids in distinguishing between regular and potentially harmful instances in the dataset, allowing for effective and timely detection of anomalous activities.

3.2. TIME SERIES DATASET DESCRIPTION

The dataset in question, CICDoS2019, is a time series dataset initially crafted by the esteemed Canadian Institute for Cybersecurity (CIC) [15]. The CICDoS2019 dataset was created by S. Ratan Kumar and a team of researchers from the Department of Computer Science and Systems Engineering at Andhra University, Visakhapatnam, India. The team aimed to develop a multicore parallel processing technique for preparing time series data to detect DDoS flooding attacks early. By utilizing parallel processing, the generation of time series data was proven to be time-efficient and aided in the rapid identification of attacks. To test their approach, the team employed the CICDDoS2019 benchmark dataset and created four distinct time series for the detection of TCP-based flooding attacks, including TCP-SYN, TCP-SYN-ACK, TCP-ACK, and TCP-RST. Results showed that their proposed method can provide a 2.3 times increase in processing speed for attack traffic when compared to sequential processing. The CICDoS2019 dataset contains various contemporary reflective DDoS attacks, such as “PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, and SNMP”. The training session began on the 12th of January at 10:30 and ended at 17:15 on the same day, while the test session started at 09:40 on the 11th of March and concluded at 17:35 on the same day. The dataset comprises of 5893 records and 7 features, as displayed in Figure 3.

	File number	time_period	#SYN Packets	#SYN-ACK Packets	#ACK Packets	#RST Packets	#TCP Packets
0	1	1	5	4	38	4	47
1	1	2	1	2	21	4	24
2	1	3	1	0	41	2	42
3	1	4	13	10	91	0	114
4	1	5	88	90	2068	24	2246
...	...	...	...	...	...	...	...
5526	818	5527	36516	4	5817	3763	46100
5527	818	5528	39516	33	1817	1763	43129
5528	818	5529	31516	5	2897	2763	37181
5529	818	5530	37516	12	3817	4763	46108
5530	818	5531	37516	34	3223	3456	44229

FIGURE 3. - Screenshot of a time series dataset.

The features mentioned in dataset are:

1. Pcap file number

2. Time period serial number

3. Number of SYN packets

4. Number of SYN-ACK packets

5. Number of ACK packers

6. Number of RESET packets

7. Number of TCP packets

The time series dataset used for this study can be found at https://ieee-dataport.org/documents/time-series-dataset-ddos-attack-detection

3.3. DATA LABELING

Anomalous behaviour, if known prior to modelling, can be resolved by supervised learning algorithms for the detection of anomalies. However, without initial feedback, it is challenging to discover such spots. Therefore, an isolation forest is employed to find anomalies, enabling the use of supervised learning techniques. The Isolation Forest algorithm is used to label the dataset into 'attack' and 'normal' packets in this study [14]. To apply the Isolation Forest algorithm to our dataset, Data is initially pre-processed by normalizing the features and removing any irrelevant or redundant features. The Isolation Forest model is then trained using a predefined number of trees (e.g., 100) and maximum tree height. After training the model, anomaly scores for each data point in the dataset are computed.

3.4. DATASET PRE-PROCESSING

One of the essential steps in constructing a machine learning model is to apply pre-processing to the dataset. Pre-processing involves preparing the data, which is crucial to reduce the amount of information and prepare it for analysis. Also, Pre-processing is necessary to prevent overfitting and to build a prediction model with accurate results.

Normalization is a popular scaling method used in the pre-processing phase, which ensures that all features are within the same range. Scaling approaches such as standard scalar and minmax scalar can be used to normalize the features. In our study, A suitable threshold is selected to separate 'attack' packets from 'normal' packets based on the distribution of the anomaly scores and the specific requirements of the application, such as desired false positive and false negative rates. Data points with anomaly scores above the threshold are labelled as 'attack' packets, while those below the threshold are labelled as 'normal' packets.

This study divides the dataset into two parts. The first is the learning phase, representing 80% of the total dataset, and the second section is the test phase, representing 20% of the datasets.

Another important aspect of data pre-processing is to handle imbalanced datasets. In many cases, one class may have many samples (majority class) compared to other classes (minority class), which can affect the efficiency of the ML system and lead to inaccurate results. To address this issue, it is necessary to balance the classes of the dataset by applying appropriate techniques such as over-sampling or under-sampling.

The Synthetic Minority Over-sampling Technique (SMOTE) algorithm is used to balance the dataset in this study [16]. SMOTE is a popular over-sampling technique that generates synthetic samples of the minority class to balance the distribution of the dataset. This technique helps improve the accuracy of the ML system by providing sufficient samples for the minority class. The SMOTE algorithm has been implemented in a Python environment to pre-process and balance the dataset before feeding it into the classification system.

3.5. PREDICTION ALGORITHMS

In this study, four supervised algorithms were used. These algorithms are explained in the following subsection:

1. Random Forest (RF)

The Random Forest (RF) is among the most prevalent machine learning-based classification models, comprising a multitude of decision trees (DT). DTs make determinations contingent upon specific parameters, featuring parent and child nodes, with the uppermost node being the root-node and terminal nodes functioning as leaf-nodes. Leaf-nodes encapsulate decisions predicated on various conditions. However, DTs are highly susceptible to data fluctuations; any data alteration without modifying conditions might result in erroneous decisions. The RF model overcomes this issue by incorporating numerous independent DTs, each deliberating with randomly generated datasets. These random datasets are derived by adjusting data occurrences while preserving the original data length, ensuring all trees receive equally sized data[17]. To enhance output accuracy, it is crucial to maintain a low correlation between trees; high correlation fosters the propagation of incorrect decisions, thereby diminishing algorithm accuracy. The ultimate decision emerges from the majority vote among all trees [18].

2. K-Nearest Neighbour (KNN)

The K-Nearest Neighbours (KNN) algorithm, a machine learning technique, operates on the principle that proximate objects exhibit similarity. To ensure optimal efficiency, selecting the appropriate value for K is essential. Decreasing K value augments the model's stability, while increasing it initially enhances stability until it surpasses a specific threshold, after which the error rate escalates. Consequently, discerning optimal K value for a given problem is crucial, achieved by testing various K values and selecting the one yielding superior stability and minimal errors. Upon determining K value, data must be loaded, and the algorithm initiated. Each data set necessitates fulfilling the specified query to obtain a result. Subsequently, the mean of the resulting data set serves as the optimal output [18].

3. Decision Tree

The Decision Tree, a frequently employed supervised machine learning algorithm, assumes an inverted tree structure to depict a specific decision problem. Decision trees encompass two categories: Classification Trees (CT) and Regression Trees (RT). Presently, these algorithms are designated as CART, an acronym for Classification and Regression Trees. CART, a term coined by Leo Breiman, refers to decision tree algorithms employed in constructing models for addressing classification and regression challenges [19].

4. XGBoost (Extreme Gradient Boosting)

Extreme Gradient Boosting (XGBoost) represents an ensemble learning technique founded upon gradient boosting, frequently employed by researchers for predicting Distributed Denial of Service (DDoS) attacks. XGBoost combines a linear model with a boosting tree model, leveraging not only the first derivative but also the second derivative of the loss function for second-order differentiation. This strategy speeds up convergence to global optimality, improving the model's overall solution efficiency. As a powerful machine learning algorithm, XGBoost belongs to the family of boosting techniques and extends the Gradient Boosting method, aiming to enhance both the algorithm's speed and performance [20].

5. Support Vector Machine

Support Vector Machine (SVM) stands as a crucial approach for detecting Distributed Denial of Service (DDoS) attacks. As a classic machine learning technique, SVM remains relevant for addressing large-scale data classification challenges, particularly within multidomain applications in big data environments. Nevertheless, SVM exhibits mathematical complexity and significant computational demands. As a supervised machine learning algorithm, SVM tackles regression and classification tasks [21].

3.6. ENSEMBLE LEARNING TECHNIQUES

Ensemble learning is a new trending method in data mining and artificial intelligence in which multiple learning algorithms are combined with the intention of synergizing the strengths of each weak algorithm to obtain a better and more efficient classifier. Ensembles are an advanced approach to machine learning that is often used when the capability and skill of the predictions are more important than using a simple and understandable model [22]. In the following subsections, This paper discusses the ensemble learning technique, called Voting that is used in this study [23].

3.6.1. Voting

The Voting Classifier represents an ensemble-based classification method, adept at unifying theoretically similar and disparate classifiers to generate a singular classifier predicated on majority characteristics derived from voting. By combining individual classifier outputs, it conducts a new vote to pinpoint the accurate prediction. Various techniques exist for acknowledging a classifier's voting process, including Hard Voting and Soft Voting, which will be clarified in the subsequent section [23].

3.6.1.1. Hard Voting

This process is also known as majority voting. In this process, the mode value is calculated and taken into consideration. If a certain mode value is highest in terms of count, then it is considered to be the voted winner. The class label y is predicted using the following formula [23] :

$$\hat{y} = \text{mode}\ \{C1(x), C2(x), C3(x), C3(x), ..., Cn(x)\} \quad (1)$$

After experimenting with Support Vector Machine (SVM) and K-Nearest Neighbour (KNN) algorithms for DDoS attack detection, it was apparent that the obtained results were not satisfactory. As a result, I opted for an ensemble approach to develop an intelligent detection system for a Distributed Denial-of-Service (DDoS) attack based on time series, and an Ensemble Learning technique. This technique involves combining multiple machine-learning models to improve the overall accuracy of the detection system. Two ensembles were constructed, namely Ensemble1 and Ensemble2, using different machine-learning algorithms. Ensemble1 consisted of three machine learning algorithms, namely Support Vector Machines (SVM), k-Nearest Neighbours (KNN), and Random Forest (RF), represented by model1, model2, and model3, respectively. The voting mechanism used in Ensemble 1 was 'hard', where the final prediction is determined by the majority vote of the constituent models. Ensemble2 consisted of three machine learning algorithms, namely Extreme Gradient Boosting (XGB), Decision Trees (DT), and k-Nearest Neighbours (KNN), represented by model4, model5, and model6, respectively. The voting mechanism used in Ensemble 2 was 'hard’as well. The individual models used in each ensemble were trained using training day data. By combining the models

using Ensemble Learning, the overall accuracy of the detection system was improved, and the likelihood of false positives or false negatives was reduced.

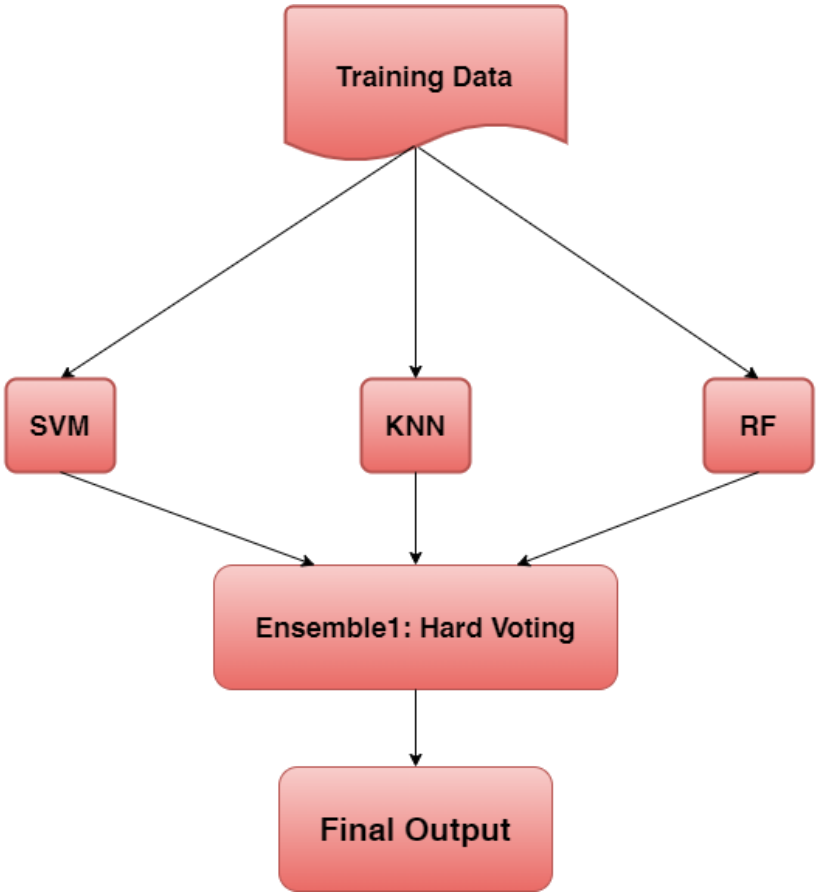

FIGURE 4. - Ensemble1 flowchart

The flowchart in Figure 4 is visually represents the machine learning process described, showing the flow of data through the system, from the training data to the individual models, through the ensembles, and finally to the output.

4. EVALUATION MEASURES AND RESULTS

The Intelligent Distributed Denial of Service Attacks Detection (IDDOSAD) approach was evaluated using a dataset of 11,423 instances. Various evaluation metrics are used to evaluate the accuracy of the prediction model. The effectiveness of supervised machine learning in detecting DDoS attacks was evaluated in this work using accuracy, F1-score, precision, recall, and the Training time, as shown by the following equations [24]:

- *Accuracy*: This is the simplest metric. It is the ratio of the correct predictions (both positive and negative) to all predictions.

  Accuracy = (True Positives + True Negatives) / (Total Number of Predictions) (2)

- *F1-score (or F1-Measure)*: This is the harmonic mean of Precision and Recall and tries to find the balance between these two metrics. It ranges from 0 to 1. F1-Score is better metric when there are imbalanced classes.

  F1-Score = 2 * (Precision * Recall) / (Precision + Recall) (3)

- *Precision*: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It is also called Positive Predictive Value. It ranges from 0 to 1. High Precision indicates an example labelled as positive is indeed positive (small number of False Positives).

$$\text{Precision} = \text{True Positives} / (\text{True Positives} + \text{False Positives}) \quad (4)$$

- *Recall (or Sensitivity or True Positive Rate)*: Recall is the ratio of correctly predicted positive observations to all observations in actual class. It ranges from 0 to 1. High Recall indicates the class is correctly recognized (small number of False Negatives) [25].

$$\text{Recall} = \text{True Positives} / (\text{True Positives} + \text{False Negatives}) \quad (5)$$

The following results were obtained as shown in Table 2:

- Random Forest: This model achieved the highest performance with 100% accuracy, F1-score, precision, and recall. The training time was 3.54 seconds, demonstrating a relatively fast learning process.
- Decision Tree: The Decision Tree model also achieved exceptional performance, with 100% accuracy, F1-score, precision, and recall. The training time for this model was only 0.06 seconds, making it the fastest among all the tested models.
- Ensemble 2: The second ensemble model exhibited 98% accuracy, F1-score, precision, and recall. The training time for this model was 0.84 seconds, showcasing the efficiency of combining multiple algorithms.
- XGBoost: With an accuracy of 94%, F1-score of 94%, precision of 95%, and recall of 94%, the XGBoost model displayed strong performance, although not as high as the top-performing models. The training time for this model was 0.75 seconds.
- Ensemble 1: The first ensemble model achieved 92% accuracy, F1-score, and recall, with a slightly higher precision of 93%. However, the training time for this model was the longest among all the models, at 6.77 seconds.

In conclusion, both the Random Forest and Decision Tree models demonstrated excellent performance in detecting and mitigating DDoS attacks, offering a reliable solution to protect communication systems against this growing cybersecurity threat. The ensemble models also showed promising results, indicating the potential for further improvement by combining different algorithms as shown in Table 2.

Table 2. - Comparison of the results.

Model	Accuracy (%)	F1-score (%)	Precision (%)	Recall (%)	Training time (sec)
Random Forest	100%	100%	100%	100%	3.54
Decision Tree	100%	100%	100%	100%	0.06
Ensemble 2	98%	98%	98%	98%	0.84
XGBoost	94%	94%	95%	94%	0.75
Ensemble 1	92%	92%	93%	92%	6.77

After calculating the execution time, DT takes 0.06 seconds to execute, which is the lowest execution time (Training time) compared to other algorithms, as shown in Figure (5).

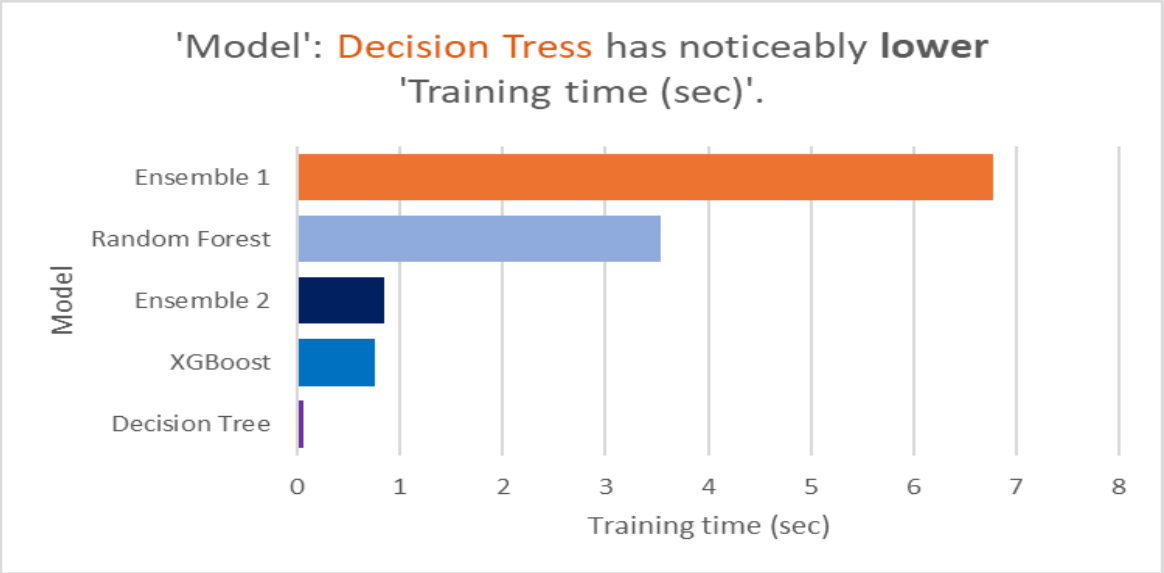

FIGURE 5. - Comparing the Training time for Each ML algorithm

Based on the comparison results, it is found that the best algorithm for classifying DDOS attacks in the (IDDOSAD) Approach is DT.

5. CONCLUSION

This study aimed to develop an effective method for detecting and preventing Distributed Denial of Service (DDoS) attacks in the increasingly complex world of IoT, smart devices, and 5G networks. The proposed Intelligent Distributed Denial of Service Attacks Detection (IDDOSAD) Approach employed supervised machine learning algorithms, including Random Forest, Decision Tree, K-Nearest Neighbor, XGBoost, and Support Vector Machine, along with ensemble learning to enhance detection accuracy.

The results demonstrated that the IDDOSAD approach effectively detects and mitigates DDoS attacks, with accuracy ranging from 92% to 100% for the time series dataset. Notably, the Decision Tree model exhibited the best performance among the tested algorithms, with 100% accuracy, F1-score, precision, and recall, while also having the lowest training time of 0.06 seconds. The ensemble models showcased the potential for further improvement by combining different algorithms, achieving promising results.

The performance of the Decision Tree model, in particular, demonstrates the potential for real-time deployment in cybersecurity systems. Future research can focus on further optimizing the ensemble learning approach and exploring the integration of unsupervised learning techniques to enhance the adaptability and resilience of the system to new and evolving threats. Additionally, the IDDOSAD approach can be tested on larger and more diverse datasets to evaluate its scalability and effectiveness across various network environments.

Despite the promising results and valuable contributions of this research, there are several limitations to acknowledge. Firstly, the detection accuracy depends heavily on the quality and completeness of the training data. If the dataset lacks variety or represents only a limited number of DDoS attack types, the model's detection ability may be compromised. Secondly, the effectiveness of our approach could potentially decline if novel, unobserved types of DDoS attacks emerge. This is a common limitation in machine learning-based detection systems, which typically perform better with known attack patterns. Future research should aim to improve the model's ability to generalize to unseen DDoS attacks, optimize computational efficiency, and finely balance detection accuracy with potential false alarms.

Funding

None

ACKNOWLEDGEMENT

All thanks to Allah Almighty, who allowed me to successfully accomplish this effort. I express my heartfelt appreciation and thanks to my supervisor, Dr. Khattab M. Ali Alheeti, for his support and for proposing the topic of this research. I appreciate him for his intellectual direction, critical remarks, and excellent recommendations during the duration of the study.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Y.-F. Li, P. K. Das, and D. L. Dowe, "Two decades of Web application testing—A survey of recent advances," *Information Systems*, vol. 43, pp. 20-54, 2014.

[2] Q. Li *et al.*, "A survey on text classification: From shallow to deep learning," *arXiv preprint arXiv:2008.00364*, 2020.

[3] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future generation computer systems*, vol. 28, no. 5, pp. 755-768, 2012.

[4] R. Hummel, C. Hildebrand, H. Modi, and G. Sockrider, "Netscout threat intelligence report," *Netscout Systems, Inc., Tech. Rep*, 2020.

[5] R. Qamar, "Gradient Techniques to Predict Distributed Denial-Of-Service Attack," *Iraqi Journal For Computer Science and Mathematics*, vol. 3, no. 2, pp. 55-71, 2022.

[6] A. A. Nafea, N. Omar, and M. M. Al-Ani, "Adverse Drug Reaction Detection Using Latent Semantic Analysis," *J. Comput. Sci*, vol. 17, pp. 960-970, 2021.

[7] M. A. Kadhim and A. M. Radhi, "Heart disease classification using optimized Machine learning algorithms," *Iraqi Journal For Computer Science and Mathematics*, vol. 4, no. 2, pp. 31-42, 2023.

[8] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi, and Y. Gulzar, "Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight," *Symmetry*, vol. 13, no. 2, p. 227, 2021.

[9] A. A. Abdulrahman and M. K. Ibrahem, "Evaluation of DDoS attacks detection in a new intrusion dataset based on classification algorithms," *Iraqi Journal of Information and Communication Technology*, vol. 1, no. 3, pp. 49-55, 2018.

[10] N. Bindra and M. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Automatic Control and Computer Sciences*, vol. 53, pp. 419-428, 2019.

[11] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evolutionary Intelligence*, vol. 13, pp. 283-294, 2020.

[12] G. Usha, M. Narang, and A. Kumar, "Detection and classification of distributed DoS attacks using machine learning," in *Computer Networks and Inventive Communication Technologies: Proceedings of Third ICCNCT 2020*, 2021: Springer, pp. 985-1000.

[13] A. A. Najar and S. Manohar Naik, "DDoS attack detection using MLP and Random Forest Algorithms," *International Journal of Information Technology*, vol. 14, no. 5, pp. 2317-2327, 2022.

[14] M. Heigl, K. A. Anand, A. Urmann, D. Fiala, M. Schramm, and R. Hable, "On the improvement of the isolation forest algorithm for outlier detection with streaming data," *Electronics*, vol. 10, no. 13, p. 1534, 2021.

[15] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019: IEEE, pp. 1-8.

[16] A. D. Amirruddin, F. M. Muharam, M. H. Ismail, N. P. Tan, and M. F. Ismail, "Synthetic Minority Over-sampling TEchnique (SMOTE) and Logistic Model Tree (LMT)-Adaptive Boosting algorithms for classifying imbalanced datasets of nutrient and chlorophyll sufficiency levels of oil palm (Elaeis guineensis) using spectroradiometers and unmanned aerial vehicles," *Computers and Electronics in Agriculture*, vol. 193, p. 106646, 2022.

[17] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," in *2019 IEEE world congress on services (SERVICES)*, 2019, vol. 2642: IEEE, pp. 184-189.

[18] S. S. Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, "Machine learning based DDoS detection," in *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2020: IEEE, pp. 234-237.

[19] H. H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74-78, 2018.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794.

[21] S. Suthaharan, "Machine learning models and algorithms for big data classification," *Integr. Ser. Inf. Syst*, vol. 36, pp. 1-12, 2016.

[22] M. Jabbar and R. Aluvalu, "RFAODE: A novel ensemble intrusion detection system," *Procedia computer science*, vol. 115, pp. 226-234, 2017.

[23] C. Hu, "Ensemble feature learning-based event classification for cyber-physical security of the smart grid," Concordia University, 2019.

[24] M. S. I. Alsumaidaie, K. M. A. Alheeti, and A. K. Al-Aloosy, "Intelligent Detection System for a Distributed Denial-of - Service (DDoS) Attack Based on Time Series," in *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, 9-12 Jan. 2023 2023, pp. 445-450, doi: 10.1109/DeSE58274.2023.10100180.

[25] M. M. Al-Ani, N. Omar, and A. A. Nafea, "A Hybrid Method of Long Short-Term Memory and Auto-Encoder Architectures for Sarcasm Detection," *Journal of Computer Science*, vol. 17, no. 11, 2021, doi: 10.3844/jcssp.2021.1093.1098.