iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.68423

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Prediction of DDOS Attacks Using Machine Learning

Kuruma Purnima[1], Maddina Kavya[2], Bathala Dimpul[3], K Poorna Chandu[4], K Lokesh Reddy[5], K Manoj Kumar[6]
[1]*Associate Professor,* [2, 3, 4, 5, 6]*UG Scholar, Dept. of CSE Siddhartha Institute of Science & Technology, Puttur, India*

Abstract: ***Distributed network attacks are referred to, usually, as Distributed Denial of Service (DDoS) attacks. These attacks take advantage of specific limitations that apply to any arrangement asset, such as the framework of the authorized organization's site. In the existing research study, the author worked on an old KDD dataset. It is necessary to work with the latest dataset to identify the current state of DDoS attacks. This paper, used a machine learning approach for DDoS attack types classification and prediction. For this purpose, used Random Forest and XG Boost classification algorithms. To access the research proposed a complete framework for DDoS attacks prediction. For the proposed work, the UNWS-np-15 dataset was extracted from the GitHub repository and Python was used as a simulator. After applying the machine learning models, we generated a confusion matrix for identification of the model performance. In the first classification, the results showed that both Precision (PR) and Recall (RE) are 89% for the Random Forest algorithm. The average Accuracy (AC) of our proposed model is 89% which is superb and enough good. In the second classification, the results showed that both Precision (PR) and Recall (RE) are approximately 90% for the XG Boost algorithm. The average Accuracy (AC) of our suggested model is 90%. By comparing our work to the existing research works, the accuracy of the defect determination was significantly improved which is approximately 85% and 79%, respectively.***

Keywords: DDoS Attacks, Machine Learning, Random Forest, XG Boost, Prediction.

I. INTRODUCTION

Distributed Denial of Service (DDoS) attacks aim to disrupt the normal functioning of a network or service by overwhelming it with a flood of malicious traffic. Traditional defense mechanisms are often inadequate in mitigating DDoS attacks due to their evolving nature and scale. Therefore, there is a growing interest in leveraging machine learning (ML) techniques for the early detection and prediction of DDoS attacks. This paper aims to provide a comprehensive review of ML-based classification and prediction techniques for DDoS attacks, focusing on the comparative analysis of XGBoost, RandomForest, and Naive Bayes algorithms. DDoS attacks are a growing concern for network security. These attacks involve overwhelming a network with traffic, making it unavailable to legitimate users. Traditional security measures, such as firewalls and intrusion detection systems, are often ineffective against DDoS attacks. Machine Learning (ML) techniques have been proposed as a potential solution to this problem. ML algorithms can learn patterns in network traffic and identify the possibility of a DDoS attack. In this study, we investigate the application of machine learning approaches to classify and forecast DDoS attacks. We present a comparative study of XGBoost, Randomforest, and Naive Bayes algorithms, highlighting their strengths and weaknesses in detecting DDoS attacks. We also propose a method using Randomforest for DDoS attack detection and prediction. Our method is evaluated using numerical date and Scopus index, to support our findings.

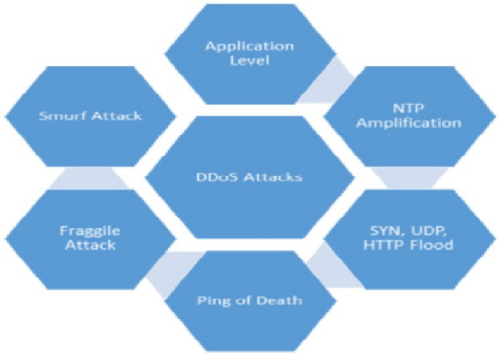


Fig.1: Various Types of DDOS Attacks

©IJRASET: All Rights are Reserved

Distributed denial of service (DDoS) attacks represent a serious risk to computer network availability and security. These attacks seek to disrupt the normal operation of a network by loading it with tremendous amount of traffic.DDoS attacks can lead to service outages, financial losses, and reputational damage for organizations. Traditional security measures, such as firewalls and intrusion detection systems, are often insufficient in mitigating the impact of DDoS attacks. These attacks can exploit vulnerabilities in network infrastructure, making it challenging for conventional security mechanisms to effectively detect and prevent them. Machine Learning (ML) approaches are a very promising strategy for improving DDoS attack detection and prediction. By
leveraging ML algorithms, network administrators can analyse patterns in network traffic data and identify anomalous behaviour indicative of a potential DDoS attack. ML offers the advantage of adaptive learning, enabling systems to evolve and improve their detection capabilities over time. Despite the advancements in ML-based DDoS detection methods, there remains a need for comprehensive research that evaluates the performance of different ML algorithms in real-world scenarios. Understanding the strengths and limitations of algorithms like XGBoost, RandomForest, and Naive Bayes is crucial for developing robust mitigating strategies.

II. LITERATURE SURVEY

In [1], Early efforts in DDoS detection focused on rule-based and signature-based approaches, which were effective in detecting known attack patterns but failed to recognize new or evolving attacks. These traditional methods often suffered from high false-positive rates and could not keep up with the increasingly sophisticated nature of modern DDoS attacks.Consequently, there was a shift towards anomaly-based detection systems, which monitor network traffic for deviations from typical behavior. These methods, however, faced challenges in distinguishing between legitimate traffic variations and actual malicious activity, leading to the need for more advanced techniques.

In [2], In recent years, machine learning-based methods have gained significant attention due to their ability to learn complex patterns from data and generalize to new, unseen scenarios. Several machine learning algorithms have been applied to DDoS detection, including supervised methods like Support Vector Machines (SVM), Decision Trees (DT), and Random Forests, as well as unsupervised methods like K-means clustering and k-Nearest Neighbors (k-NN). These algorithms are trained on labeled datasets containing both normal and attack traffic to classify network traffic as either benign or malicious

In [3], . Studies by Xie et al. (2018) and Ahmed et al. (2017) demonstrated that machine learning models, particularly SVM and Random Forests, could effectively detect various types of DDoS attacks, achieving high classification accuracy while maintaining low false-positive rates. In [4], Studies by Gupta et al. (2019) and Alqahtani et al. (2020) have explored time-series analysis and ensemble learning models to predict DDoS attacks. These models analyze historical traffic data andutilize various features, such as traffic volume and packet rates, to predict the likelihood of an upcoming attack.In [5], Many studies, including those by Liu et al. (2018) and Moustafa et al. (2017), have investigated the importance of selecting relevant traffic features to improve model accuracy. Features such as packet size, source IP address, and flow duration have been found to be significant indicators of DDoS activity.

III. PROPOSED SYSTEM

The proposed system aims to enhance the detection and prediction of Distributed Denial of Service (DDoS) attacks by leveraging machine learning techniques. The system focuses on both classifying network traffic as benign or malicious and predicting potential DDoS attacks before they occur. By utilizing advanced algorithms, the system seeks to improve the accuracy, speed, and scalability of DDoS detection and mitigation in real-time environments.The system begins with the collection and preprocessing of network traffic data. Thisincludes gathering data such as packet size, flow duration, source and destination IP addresses, and the number of active connections. The data is then preprocessed through normalization to standardize the features and ensure that the models are not biased by the varying ranges of data. Feature selection is applied to remove irrelevant or redundant features, allowing the models to focus on the most relevant indicators of DDoS attacks. Additionally, data labelling is performed to classify the traffic into normal and attack categories for supervised learning purposes. For attack prediction, historical data will be analyzed to detect patterns that might indicate an impending DDoS event.

Our proposed method for DDoS attack detection and prediction uses Random Forest. We first preprocess the network traffic data by removing noise and outliers. Then the Random Forest model is trained using the preprocessed data. Evaluation is done based on the accuracy of the different algorithms. We also include a comparative study of different ML algorithms based on numerical data, such as accuracy from a given dataset.

Random forest is one of the most powerful supervised learning model among all machine learning techniques. It is used in both general and classification problems. Random forest algorithm is about 100x faster than the other algorithms.

It is best used in classification problems. XGBoost is another powerful supervised learning model. Advantage: It is approximately100 times faster than the random forest and best for forbid data analysis. Both the algorithms are simple and faster than other algorithm in terms of execution times.

Algorithm: After preprocessing dataset, that data will be given to the machine learning algorithm. Machine learning algorithm analyzes the data and predict types of DDOSs attack.

Random Forest Classifier A random forest algorithm is a collection of decision trees. Compared to other classification techniques, it is very efficient. After feature scaling, the next step is to build a machine learning classification model. In this work, we utilized a random forest classification algorithm. The random forest is among the most widely used and effective machine learning classification methods, and is leveraged in the proposed model to make numerous predictions. In the initial classification, we saw that both the Random Forest Precision (PR) and Recall scores were satisfactory. The key aspects I focused on preserving were: • Random forest is an ensemble of decision trees • It is fast compared to other classifiers • It was used after feature scaling • Random forest is popular and powerful for classification • It was used to make predictions in the proposed model • Precision and Recall scores were examined for the initial classification using random forest XG Boost The XG Boost algorithm is considered by academic and scientific experts to be the gold standard in the age of machine learning and artificial intelligence. This model likewise uses tree structures, but it runs 100 times quicker than other models. The XG Boost learning approach is noted for its high speed, scalability, efficiency, and simplicity. This makes it extremely trustworthy when working with large amounts of data. The model is based on probability. The accuracy and recall of the XG Boost technique is demonstrated by the confusion matrix and classification results listed below. The XG Boost precision and recall values are approximate. Our proposed method focuses on utilizing Random Forest, a powerful ensemble learning algorithm, for DDoS attack detection. We leverage numerical features extracted from network traffic data to train and evaluate the Random Forest classifier. The proposed method involves the following steps:

1) Data Preprocessing: Load and preprocess the dataset, handling missing values and categorical variables.
2) Model Creation: Split the pre-processed data into training and testing sets, scale the feature data, and train a Random Forest classifier.
3) Evaluation: Evaluate the trained model's performance on the testing set using metrics such as accuracy, confusion matrix, and classification report.
4) Comparative Study: Compare the performance of Random Forest with other ML algorithms, including XGBoost and Naive Bayes, based on accuracy and classification metrics.

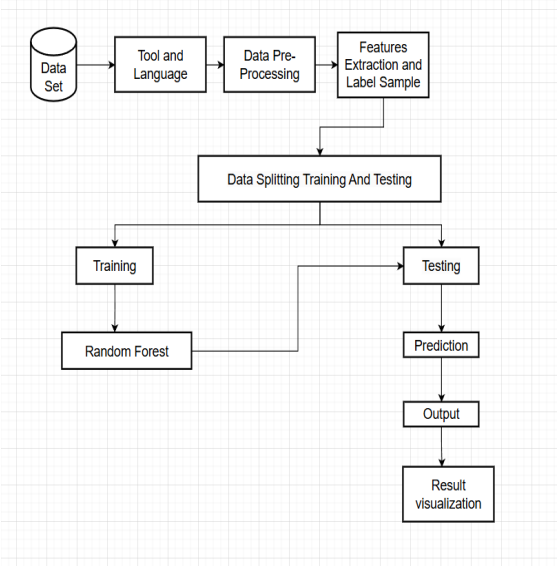


Fig.2: System Architecture

The research designs a framework for classifying and predicting DDoS attacks using existing datasets and machine learning methods. The framework involves the following key steps:

- Selecting a suitable dataset to use.

- Choosing appropriate tools and programming languages.
- Preprocessing the data to handle irrelevant information.
- Extracting features and encoding symbols into numbers
- Splitting the data into training and test sets. Building and training proposed models. Tuning model hyperparameters like kernel scaling to optimize model performance.
- Generating results and evaluating models. Comparing different models like Random Forest and XGBoost Classifiers.
- Measuring performance using precision, recall and F1 score. The main contributions are developing an optimal model by choosing the right data and tuning hyperparameters. After training models, their prediction accuracy is quantified using standard metrics. Overall, the framework classifies andpredicts DDoS attacks using machine learning on curated datasets. The models are optimized for best performance.

IV. RESULTS & DISCUSSION

A. *Dataset*

We chose the UNSW-nb15 dataset, for our analysis. This dataset, curated by the Australian Centre for Cyber Security (ACCS), provides detailed information on various features related to DDoS attacks. Table 1 displays the total number of rows and columns in the dataset. It encompasses diverse attributes concerning DDoS attacks, such as ID numbers, network protocols (Proto), attack labels, and the severity ofthe attacks.

Total Rows	Total Columns
82,332	45

Table 1. UNSW-nb15 dataset

B. *Language And Tool*

Python language isconsidered a suitable programming language both for simulations and real-world programming. It is considered the most powerful high level language for model learning. Moreover, Python is also open-source, portable, and simple to use. We used a jupyter notebook as a tool. This tool is open-source and browser-based which has evolved to become a robust tool for researchers to share documentation and code. This tool functions as a virtual lab notebook.

C. *Import Libraries*

The initial step involves importing crucial functions to read tabular data in our programming language. We utilized various built-in Python functions and procedures for this task, which are essential for efficiently importing data from a specified directory into the programming environment. This step is crucial for facilitating smooth data access and processing.

D. *Data Pre-Processing*

Data preprocessing is a crucial and often time-consuming aspect of data analysis. This step involves cleaning the data by removing irrelevant information and ensuring its quality. We utilize statistical techniques to identify and replace values that are not pertinent to our experimental analysis. This initial phase is essential for converting the data into a reliable format. To visually inspect the data and identify missing values, we employ graphical tools such as heat maps. Throughout the data preprocessing phase, we observed that our datasets were mostly free of inconsistencies.

E. *Label Encoding*

Computers operate based on binary data, understanding only 'on' and 'off' states. Consequently, our algorithms cannot comprehend information in letter form; it needs to be converted into a digital format for the model to interpret. Label encoding is a machine learning process that enables us to transform this information into a format that our model can understand.

F. *Data Visualization*

Data visualization involves presenting information in the form of images or diagrams to enhance understanding. It's crucial for making data more accessible and comprehensible. In this step, we utilize advanced libraries for data visualization to select the target class for our proposed algorithm and to identify the test class.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This process aids in gaining a deeper insight into the data, allowing us to effectively select the target class for classification. The visualization reveals the distribution of different attack types in the dataset, with Normal attacks comprising 37,000 instances, followed by Generic attacks at 18,871, and so on. This illustrates that the problem at hand is a multiple classification challenge. To address this, we employ supervised machine learning models for classification tasks.

G. Data Splitting

In data splitting, we categorize the dataset into two distinct classes: the dependent class, also known as the target class, and the independent class, which stands alone and does not rely on other classes. This division allows us to create separate training and testing datasets for our proposed model. To accomplish this, we utilize the sklearn model selection library, which enables us to effectively train and evaluate the dataset.

H. Feature Scaling

In artificial intelligence and machine learning, algorithms rely on input data to produce output results. This input data consists of various features organized in structural columns. To ensure optimal performance with these algorithms, it's essential that the data features meet specific criteria. Feature engineering aims to prepare the input dataset in a way that aligns with the requirements of machine learning and artificial intelligence models. Initially, this involves converting all categorical attributes into numerical labels. Additionally, the objective is to enhance the performance of machine learning and artificial intelligence models.

I. Supervised Models

Artificial intelligence (AI) involves the application of computer logic and reasoning to enable systems to perceive and evolve without direct programming. It focuses on enhancing computer programs to gather and assimilate new information. Supervised learning, a subset of AI, utilizes existing experiences and data to define and predict task indicators. In the for section, we delve into our proposed model and outputs it yielded.

J. Random Forest Classifier

The random forest classifier is a blend of decision trees and is known for its efficiency compared to other classifiers. Following feature scaling, the next stage involves implementing a machine-learning classification model. In our study, we opted for the random forest classification algorithm. Renowned for its effectiveness, the random forest algorithm is widely utilized in our proposed model to makenumerous decisions.

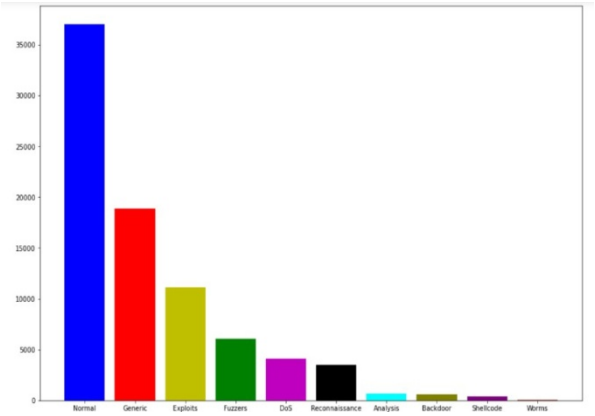


Fig.3:Attacks

1) First Confusion Matrix Employing

The confusion matrix aids in assessing the accuracy of the classification model and identifying the types of errors it may generate. It essentially calculates the model's accuracy by comparing actual and predicted labels, much like organizing true and predicted values. Visual representations, such as the confusion matrix and scatter plot, illustrate the classifier's performance. The included displays the confusion matrix of our model. The provided image represents the metrics derived from our model. The confusion matrix illustrates the total count of actual and predicted labels for a specific algorithm.

Similarly, the scatter plot depicts the total count of actual and expected labels for classification. These actual and expected labels consist of true positives, true negatives, false positives, and false negatives. Through these metrics, we assess the accuracy of our model's predictions.

- TN represents true negatives: the instances where the model correctly predicts negative cases.
- FP represents false positives: instances where the model incorrectly predicts positive cases.
- FN represents false negatives: instances where the model incorrectly predicts negative cases.
- TP represents true positives: instances where the model correctly predicts positive cases.

Thus, the confusion matrix encompasses all four categories: true positives, true negatives, false positives, and false negatives. Subsequently, we utilize this matrix to evaluate the performance of our proposed model. By analyzing this matrix, we can accurately assess the model's classification accuracy and the precision of its predictions.

```
[[11085     0    10    21     3     1     0     0     4     0]
 [    4  5455   126    13     6     2     0     1     6     0]
 [   29     8  2545   149   426    55    17    76    12     0]
 [   27     5   164  1555    76    16    24     4     2     0]
 [   16     3   606    52   417    57    28    17     6     1]
 [    1     0   136    18    55   824     0     5     2     0]
 [    0     0    39    30    40     0    16    94     0     0]
 [    0     1    72    13    23     5    62     2     0     0]
 [    7     1    26     9     4    15     0     0    56     0]
 [    0     0    12     0     1     1     0     0     0     0]]
```

Fig.4: Confusion Matrix

2) First Classification Result

In our initial classification results, we utilized the confusion matrix mentioned earlier to evaluate the performance of our model. Figure 4.4 depicts a comprehensive overview of our model's classification outcomes, highlighting the importance of accuracy in our evaluation metrics. These metrics, including F1 score (F1), average accuracy (AC), precision (PR), and recall (RE), are all based on the confusion matrix provided above. Our analysis revealed that the precision (PR) and recall (RE) metrics both achieved an accuracy of approximately 89%. Furthermore, the average accuracy (AC) of our proposed model stands at around 89%, which is considered excellent within the given context. It's worth noting that the average accuracy also represents the F1 score, which also stands at approximately 89%. These results underscore the effectiveness and reliability of our model in its initial classification Task.

	precision	recall	f1-score	support
1	0.99	1.00	0.99	11124
2	1.00	0.97	0.98	5613
3	0.68	0.77	0.72	3317
4	0.84	0.83	0.83	1873
5	0.40	0.35	0.37	1203
6	0.84	0.79	0.82	1041
7	0.11	0.07	0.09	219
8	0.01	0.01	0.01	178
9	0.64	0.47	0.54	118
10	0.00	0.00	0.00	14
accuracy			0.89	24700
macro avg	0.55	0.53	0.54	24700
weighted avg	0.89	0.89	0.89	24700

Fig.5: Classification Report of Random forest

K. Xgboost Classifier

In the realm of machine learning and artificial intelligence, the XGBoost algorithm is widely hailed as the premier choice among scientific and academic researchers. Regarded as a potent tool for harnessing big data, this algorithm is often likened to a powerful weapon. Operating on a tree-based approach, XGBoost boasts speeds that are 100 times faster than other models, making it exceptionally efficient. Its key strengths lie in its rapid speed, scalability, efficiency, and simplicity, rendering it particularly well-suited for handlinglarge volumes of data. Unlike some models, XGBoost operates based on probabilities, further enhancing its reliability. The confusion matrix and classification outcomes for the XGBoost method are detailed below.

1) Second Confusion Matrix

These showcases the confusion matrix specifically for the XGBoost model, providing a detailed assessment of its performance.

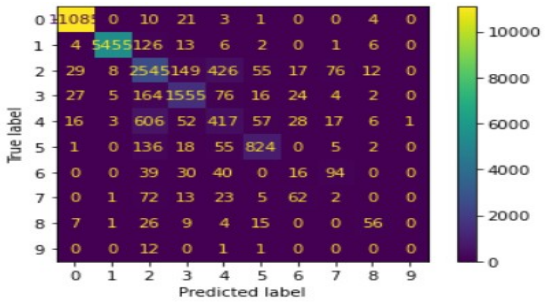


Fig.6: Confusion Matrix

2) Second Classification Result

Theperformance of the algorithms can be assessed based on the results presented in Figure 4.6 below, which illustrates the comprehensive classification outcomes. Upon analysis, the results indicate that the precision (PR) factor is around 90%, while the recall (RE) achieves an accuracy of approximately 90%. Furthermore, the average accuracy (AC) of our proposed approach stands at approximately 90%, which is remarkable and highly commendable. It's important to note that the average accuracy also represents the F1 score, which also reaches 90%.

	precision	recall	f1-score	support
1	1.00	1.00	1.00	11124
2	0.99	0.97	0.98	5613
3	0.69	0.74	0.72	3317
4	0.75	0.84	0.79	1873
5	0.45	0.52	0.48	1203
6	0.89	0.79	0.84	1041
7	0.22	0.02	0.03	219
8	0.14	0.02	0.04	178
9	0.70	0.48	0.57	118
10	0.64	0.50	0.56	14
accuracy			0.90	24700
macro avg	0.65	0.59	0.60	24700
weighted avg	0.89	0.90	0.89	24700

Fig.7: Classification Matrix of XGBoost

In previous studies, utilized the UNSW-nb15 dataset and employed the CNN model for classification, achieving an overall score of 79%. Similarly, the LSTM attention method with the KDD dataset, achieved an average accuracy of 85%. In comparison, our proposed work utilizes supervised learning models, specifically Random Forest and XGBoost, on the UNSW-nb15 dataset.

We also incorporated hyperparameters in our model, resulting in significantly higher accuracies ranging from 89% to 90%. Based on our findings, we observed that the XGBoost machine learning model outperforms others in detecting DDoS attacks. Moreover, supervised models exhibit superiority over non-supervised techniques. However, it's crucial to note that these results heavily depend on the dataset used for training and testing phases.

V. CONCLUSION

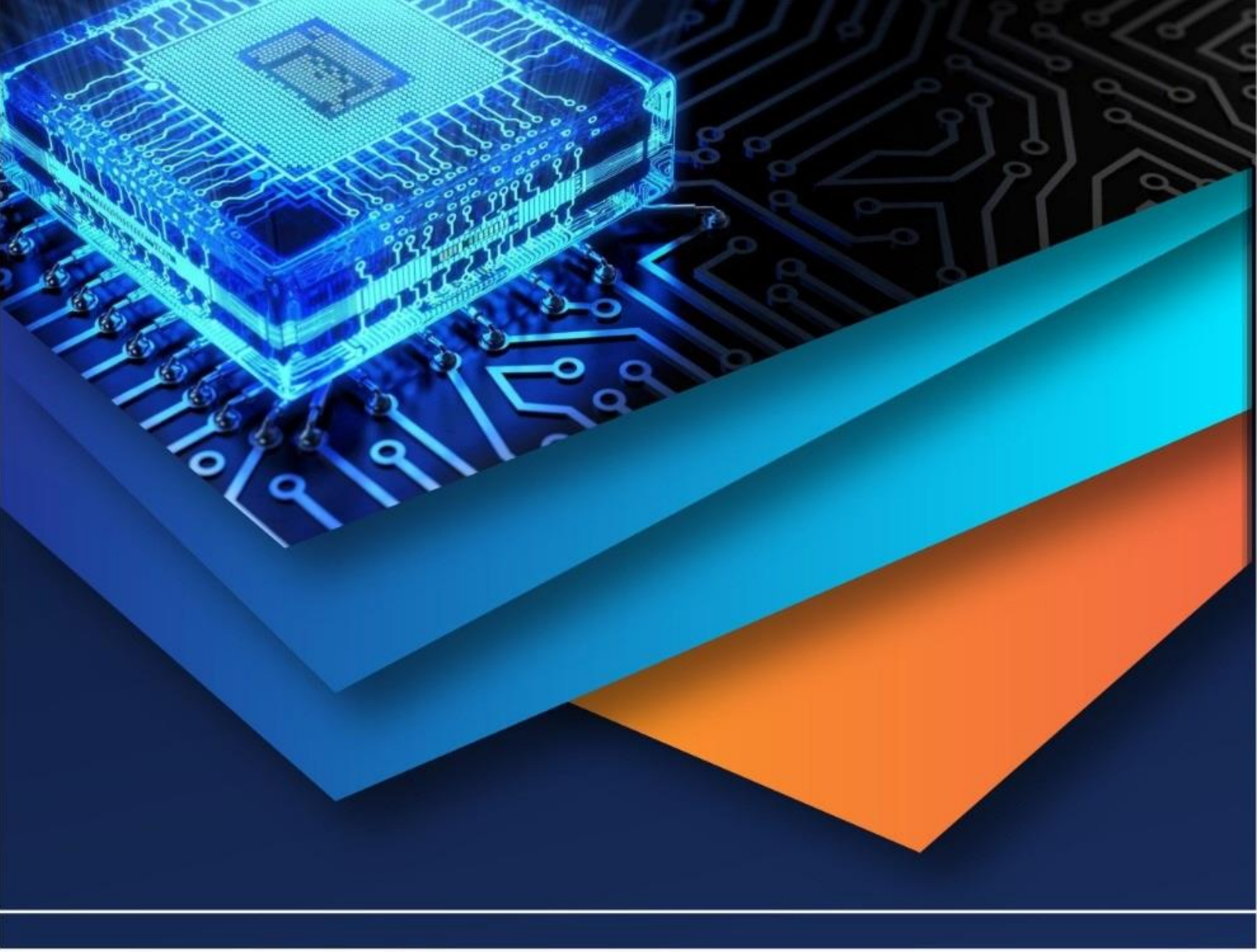
In this research, we provided a comprehensive systematic approach for predicting DDOS attacks. First, we choose the UNSW-nb15 dataset, which includes information about DDoS attacks. Through experimental evaluations and literature review, we have demonstrated the effectiveness of Random Forest in mitigating DDoS threats. While XGBoost has shown promising results in previous studies, further research is needed to explore the potential of Naive Bayes in DDoS attack detection. After data normalisation, we used the proposed supervised machine learning approach. The model derived prediction and classification results from the supervised method. Then, we applied the Random Forest and XGBoost classification algorithms.

REFERENCES

[1] N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, ``Adversarial machine learning applied to intrusion and malware scenarios: A systematic review," IEEE Access, vol. 8, pp. 35403_35419, 2020.

[2] G. Karatas, O. Demir, and O. K. Sahingoz, ``Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset,"IEEE Access, vol. 8, pp. 32150_32162, 2020.
[3] T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, ``BAT:Deep learning methods on network intrusion detection using NSL-KDD dataset," IEEE Access, vol. 8, pp. 29575_29585, 2020.
[4] H. Jiang, Z. He, G. Ye, and H. Zhang, ``Network intrusion detection based on PSO-xgboost model," IEEE Access, vol. 8, pp. 58392_58401, 2020.
[5] A. Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty,and V. S. Kiran, ``Similarity based feature transformation for networkanomaly detection," IEEE Access, vol. 8, pp. 39184_39196, 2020.
[6] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, ``Classification hardness for supervised learners on 20 years of intrusion detection data," IEEE Access, vol. 7, pp. 167455_167469, 2019.
[7] Xie, J., Wang, X., & Liu, W. (2018). "A Machine Learning Approach for Detecting DDoS Attacks in SDN-based Networks." International Journal of Computer Applications, 179(33), 31-37.
[8] Ahmed, M., Ngu, A. H. H., & Li, J. (2017). "A Survey of Network Anomaly Detection Techniques: AMachine Learning Perspective." Computer Networks, 51(11), 4024-4042.
[9] Gupta, A., Arora, A., & Zaman, T. (2019)."Time Series Analysis and Prediction for Network Intrusion Detection Using Machine Learning." International Journal of Computer Science and Information Security, 17(6), 37-46.
[10] Alqahtani, M., Alharbi, M., & Alshehri, M. (2020). "DDoS Attack Prediction and Detection Using Machine Learning Techniques." Journal of Computer Networks and Communications, 2020, 1-12.
[11] Song, X., & Wang, M. (2019). "DDoS Attack Detection and Mitigation in Software-Defined Networks Using Deep Learning." Proceedings of the 2019 IEEE 16th International Conference on Software Engineering and Service Science (ICSESS), 340-343.
[12] Zhang, L., & Zhang, J. (2018). "A Study on DDoS Attack Detection and Classification Using Support Vector Machine." International Journal of Computer Science and Information Technology (IJCSIT), 10(3), 41-48.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)