iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 Issue: IV Month of publication: April 2024

DOI: https://doi.org/10.22214/ijraset.2024.60590

www.ijraset.com

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

DDoS Attack Detection and Classification using Machine Learning Algorithms

Prof. Shwethashree GC[1], Tanusha R[2], Harshitha PB[3], Dhyan Ganesh S[4], NS Dheeraj Gowda[5]
[1]Professor, [2, 3, 4, 5]Student, Department of Computer Science and Engineering, JSSSTU, Mysore, Karnataka, India

Abstract: In the dynamic realm of networking, Software Defined Networking (SDN) emerges as a transformative force, offering centralized control and programmable capabilities that empower network administrators to efficiently manage and secure network infrastructures. However, amidst the ever-present threat of distributed denial-of-service (DDoS) attacks, the need for robust detection mechanisms is imperative. This research proposes a machine learning-based approach to enhance DDoS attack detection and classification within SDN environments. Leveraging the SelectKBest algorithm for feature selection and employing various classifiers such as Decision tree, KMeans++, XGBoost, etc.. with a focus on Random Forest as the most effective, the project aims to bolster detection accuracy and efficiency. Through comprehensive experimentation and comparative analysis, the efficacy of the proposed methodology in identifying DDoS threats is demonstrated, contributing to the ongoing efforts in fortifying cybersecurity defenses against sophisticated adversarial tactics.

Keywords: Cyber Attack, Attack Detection Mechanisms, DDoS Attacks, Machine Learning, Random Forest Algorithm, SelectKBest Algorithm.

I. INTRODUCTION

The Software Defined Network(SDN) is a new network architecture which is the decoupling of forwarding plane from the control plane. [4] SDN differs from traditional networks as it separates Control Plane and Data Plane. Therefore SDN can improve network management, scalability, and dynamics. Thus SDN can dynamically modify the program to increase network security. In traditional network architecture, the network devices such as router and switch are managed and controlled by the network administrator according to the devices vendor company. Open Networking Foundation (ONF) develops SDN architecture, where the network administrators perform and manage network service from the centralized SDN controller. There are three layers found in SDN architecture:

1) *Infrastructure layer:* This layer is also called a data plane. In this layer, the forwarding devices such as switches and routers can forward and drop incoming packets according to the flow table which is configured by the control plane via southbound interface such as openflow protocol.
2) *Control layer*: The middle control layer is called control plane. The main function of the control plane is to install the flow rules to the forwarding devices whether the traffic is forwarded or dropped.
3) *Application Layer:* The upper application layer is also called management plane which gives applications and services over control and infrastructure layer through Representational State Transfer (REST) APIs.

With separate control and data fields, SDN can implement a centralized control function where all packets are received by the switch from the host, then the controller determines the packet delivery path. [4] The controller is the heart of the SDN where most of the functions of the SDN depend on the controller. Because it is centralized, SDN becomes the target of attack. A DDoS attack can take advantage of this functionality by attacking the Server via the registered Host. SDN network security must be considered especially for IoT, Server, or Cloud. A DDoS attack can be launched in a variety of ways, and its main effect is to decrease the availability of services, which can lead to losses in money and a variety of other issues.

The project at hand is dedicated to implementing a machine learning-based DDoS attack detection and classification system within an SDN environment. Key objectives include refining preprocessing techniques, training machine learning models to distinguish between normal and malicious network activities, and implementing a user-friendly interface for network administrators to monitor and respond to potential DDoS attacks in SDN environments. Through this research endeavor, the project aims to advance DDoS detection techniques in SDN environments, ultimately fortifying network security and ensuring the uninterrupted delivery of network services.

II. RELATED WORK

Mainly there are two previous DDoS attack detection methods:

A. Entropy-based DDoS Attack Detection

Researcher [1] developed an entropy-based algorithm capable of detecting DDoS attacks and identifying attacking paths. By analyzing entropy variations in destination IP addresses and flow initiation rates, the method swiftly detects attack traffic and triggers mitigation processes for network protection. [2] Another study introduced StateSec, a stateful SDN approach leveraging in-switch processing to detect and mitigate DDoS attacks. StateSec monitors packet matching features and employs an entropy-based algorithm for efficient and precise detection, outperforming traditional methods like sflow.

B. Machine Learning-based DDoS Attack Detection

[3]A researcher enhanced the Support Vector Machine (SVM) algorithm with Advanced Support Vector Machine (ASVM) techniques. This multi-class classification method categorizes DDoS attacks into three classes, utilizing volumetric and asymmetric features. Experimental results demonstrate a detection accuracy of approximately 97% with reduced training and testing times.

The research paper, [4] authored by Yudha Purwanto and his team, proposes a method for detecting Distributed Denial-of-Service (DDoS) attacks in SDN using Machine Learning with Ensemble Algorithm. Recognizing the vulnerability of SDN to DDoS attacks due to its centralized control, the study focuses on enhancing security through high detection accuracy and efficiency. The research consists of two main methodologies: clustering and classification, and detection validation. By employing Ensemble Algorithm techniques such as K-means++ and Random Forest, the study achieves significant improvements in accuracy and efficiency in detecting DDoS attacks within SDN environments. Experimental validation conducted on the InSDN dataset using the Mininet emulator demonstrates the effectiveness of the proposed approach, showcasing 99% accuracy, precision, recall, and F1-Measure, along with low processing time.

In one of the research [5] project, conducted by Dr. J. Thangakumar, along with Dr. M. Sambath and S. Santhosh, focuses on enhancing the detection of Distributed Denial-of-Service (DDoS) attacks using machine learning techniques. The study employs the CICDDoS2019 dataset and various machine learning algorithms, including Random Forest, XGBoost, and a modified version of XGBoost, to develop a robust detection model. By comparing the performance of these algorithms, the research aims to identify the most effective approach for accurately detecting DDoS attacks. Through extensive experimentation and analysis, the study demonstrates that the modified XGBoost classifier achieves the highest accuracy rate of 97% thereby enhancing cybersecurity.

Aye Thandar Kyaw [8] and his colleagues proposed a DDOS attack detection system for SDN networks using machine learning algorithms, specifically comparing the performance of linear and polynomial Support Vector Machine (SVM) classifiers. The proposed system utilizes flow data collection, feature extraction, and attack classification, employing the polynomial SVM algorithm to differentiate between normal and attack traffic. Through experimental evaluation, the paper demonstrates that the polynomial SVM classifier achieves higher accuracy and lower false alarm rates compared to the linear SVM classifier, providing an effective approach for mitigating DDOS attacks in SDN networks.

Nisharani Meti [9] showed the result by comparing machine learning algorithms: Naïve Bayes, Support Vector Machine (SVM) and Neural Network (NN) classifier to detect the legitimate and illegitimate connection. This paper showed the implementation of the proposed mechanism by using Mininet and Ryu SDN controller on different topologies. According to the experimental result, the author proved that SVM was a better classifier compared to the other two machine learning algorithms.

The results of recent study, however, are constrained. First off, because specific classification models are statically chosen, they have a poor detection rate. Second, some of the earlier methods had high computing costs during the model-training phase. Thirdly, because it can be difficult to cope with big amounts of data, many earlier methods relied on relatively limited data sets.

III. ALGORITHMS USED

A. Stratified K Fold

This is a data splitting method for cross-validation. It shuffles the data while maintaining the original class distribution, then splits it into folds (groups) for training and testing a model. Hence it ensures each fold maintains the same class distribution as the entire dataset.

It includes benefits as it provides a more reliable estimate of model performance on imbalanced datasets and also reduces the bias towards the majority class. It helps identify models that might struggle with the minority class. It provides a robust estimation of model performance compared to a single train-test split.

Usually K-Fold Cross-Validation splits the data into k groups (folds) of (almost) equal size. In each iteration (fold), it uses one fold for testing (validation set) and the remaining k-1 folds for training. Finally it evaluates the model's performance on the testing set and repeats the process for all k folds.

B. Random Forest Classifier

Using decision trees, the Random Forest is a supervised machine learning algorithm that can be applied to regression, classification, and other tasks. Random forests are especially useful for managing complicated and sizable datasets, managing high-dimensional feature spaces, and offering insights into the significance of individual features. This algorithm is widely used in many different domains due to its capacity to minimize overfitting and maintain high predictive accuracy. A randomly chosen portion of the training set is used by the Random Forest Classifier to generate a set of decision trees. It starts with a set of decision trees (DT) drawn at random from the training set. To determine the final prediction, it tallies the votes from each decision tree.

There are two stages to Random Forest(RF): the creation of the random forest and the creation of the random forest classifier. Initially, the Random Forest creation pseudocode is displayed.

- At random, choose "k" features (where k<< m) from the total "m" features.
- Determine the node "d" between "k" features by utilizing the optimal separator point. Partition nodes into offspring nodes by utilizing the optimal separation.
- Continue steps 1 through 3 until the number of nodes equal to "I" is reached.
- To construct number of trees "n," repeat steps 1 through 4 a number of times. This will result in the creation of multiple trees.
- In the subsequent stage, once the Random Forest classifier is established, it proceeds to make predictions. Below, you'll find the pseudocode outlining the steps performed by the Random Forest algorithm.
- Creates a decision tree at random using the test feature and each tree's rules to predict the outcome. The predicted outcome (target) is then stored.
- Total the votes for every projected goal.
- Take into account that the Random Forest algorithm's final prediction is the target that received the most votes.

When combined, Stratified K-Fold cross-validation with Random Forest, the following steps are typically followed:

1) Splitting the Data: The dataset is divided into K folds, maintaining the original class distribution. Each fold serves as a validation set, and the remaining folds are used for training the model.
2) Training the Model: For each fold, a Random Forest classifier is trained on the training data (K-1 folds).
3) Model Evaluation: The trained classifier is then evaluated on the validation set (the fold that was held out), and performance metrics such as accuracy, precision, recall, and F1-score are computed.
4) Performance Aggregation: Performance metrics from each fold are aggregated to obtain an overall estimate of the model's performance. This could involve calculating the average or weighted average of the metrics across all folds.
5) Final Model Training: After cross-validation, the model can be retrained on the entire dataset (if desired) to make predictions on new, unseen data.

Combining Stratified K-Fold cross-validation with Random Forest helps in obtaining more reliable estimates of the model's performance by reducing the impact of variability due to different train-test splits. It also allows for better utilization of the available data by using each data point for both training and validation across different folds. This approach is particularly beneficial when dealing with datasets with limited samples or imbalanced class distributions.

C. SelectKBest feature

One of the most popular techniques for selecting features is SelectKBest. It is a kind of filter-based feature selection technique used in machine learning. Filter-based feature selection techniques select features without relying on a particular machine learning algorithm. Rather, the features are ranked and scored using statistical methods.

SelectKBest scores and ranks the features according to how they relate to the output variable using statistical tests such as the mutual information score, ANOVA F-test, and chi-squared test. The K features that have the best scores are then chosen to be a part of the final feature subset.

Working with huge datasets requires the ability to quickly reduce the feature set to a manageable quantity, which is something that SelectKBest excels at doing. The parameters of SelectKBest are k and the score function. The feature importance is assessed via the usage of the score function.

D. Chi-square Testing

A statistical test called the chi-square test is performed to ascertain whether two categorical variables significantly correlate with one another. Since it is non-parametric, it does not make any assumptions on the data's distribution. The comparison of observed and expected frequencies inside a contingency table is the basis of the test. The chi-square test examines the relationship between the parts to assist with feature selection issues. It establishes if the correlation between two sampled categorical variables would represent the true correlation between them in the population.

It is a member of the continuous probability distribution family. The sum of the squares of the k independent standard random variables, as provided by is the definition of the Chi-Squared distribution.

χ^2=∑(O-E)^2/E

Where:

χ^2 is the chi-square test statistic.

O is the observed frequency of a category.

E is the expected frequency of a category under the null hypothesis of independence.

Chi-square test is utilized for categorical highlights in a dataset. We calculate Chi-square between each highlight and the target and select the required number of highlights with best Chi-square scores. Highlights that appear critical conditions with the target variable are considered imperative for forecast and can be chosen for assist investigation.

IV. PROPOSED METHODOLOGY

A. Dataset Selection and Preprocessing

1) Dataset Selection:

- *NSL-KDD :* The NSL-KDD dataset serves as a refined version of the KDD'99 dataset, addressing its inherent issues by eliminating redundant records, ensuring unbiased model training and evaluation. It achieves this by selecting records from each difficulty level group in inverse proportion to their percentage in the original KDD'99 dataset. Notably, the dataset comprises 41 features detailing the characteristics of the cyber network, providing a comprehensive foundation for research and analysis in cybersecurity.
- *CICDDoS:* The CICDDoS2019 dataset accurately depicts both conventional network traffic and various Botnet-based network attacks. The dataset was utilized in this investigation by researchers at the Canadian Institute for Cybersecurity (CIC), ensuring its reliability and relevance. It comprises 88 features extracted using CICFlowMeter and includes 11 types of DoS attacks generated in controlled settings. Furthermore, the CIC researchers devised a technique to identify the 22 key components of each DDoS attack within the dataset.

2) Data Preprocessing

The dataset undergoes thorough preprocessing to ensure data quality and compatibility with machine learning algorithms. This includes handling the missing values, encoding categorical variables, and scaling the numerical features. Preprocessing steps are essential to prepare the dataset for effective model training and evaluation.

B. Initial Model Training and Evaluation

1) *Algorithm Selection:* Multiple Machine Learning algorithms, including Logistic Regression, Decision Tree, Gradient Boosting, Random Forest, XGBoost, Deep Learning, Gaussian Naive Bayes, Stratified K-Fold and K-Means++ are trained on the preprocessed dataset. These diverse set of algorithms allows for comprehensive exploration of different modeling approaches.
2) *Model Evaluation:* Each trained model is evaluated using standard performance metrics such as accuracy, precision, recall, and F1-score. Evaluation results provide insights into the strengths and weaknesses of each algorithm, guiding the selection of the most effective models for further refinement.

C. Feature Selection

1) *Select K Best :* Select K Best is a feature selection technique that ranks features based on their individual scores using statistical tests. It evaluates each feature independently in relation to the target variable (DDoS attack or normal traffic). Features with the highest scores are considered the most informative and are selected for inclusion in the final feature set.

2) *Chi-Square Test :* The chi-square test assesses the independence between each feature and the target variable. It determines whether there is a significant association between the presence of a feature and the occurrence of a DDoS attack. Features with lower p-values and higher chi-square scores are considered more relevant for classification, as they exhibit stronger relationships with the target variable. Selected features are listed below:

TABLE I
FEATURES EXTRACTED FOR THE PROPOSED SYSTEM

No.	Feature	No.	Feature
1	Duration	6	Hot
2	Service	7	Count
3	Src Bytes	8	Service Count
4	Dst Bytes	9	Dst Host Count
5	Wrong Fragment	10	Dst Host Service Count

D. Model Refinement and Integration

1) *Model Adjustment:* The selected machine learning algorithms are fine-tuned and optimized based on the chosen features and evaluation results. Upon training on KDD dataset. Random Forest is chosen for its exceptional performance, with an accuracy of 99.5% during initial evaluation. While it gave an accuracy of 99.2% on CICDDoS19 dataset.
2) *Integration with Flask:* Random forest algorithm is integrated into the Flask backend of the application. Flask provides a flexible framework for deploying machine learning models as part of web applications, ensuring seamless interaction with users.

E. Front End Development

1) *User Interface Design:* The front-end interface is designed to provide users with an intuitive experience for interacting with the DDoS detection system. Input forms are customized to facilitate feature selection and visualization of model outputs, enhancing user engagement and usability.
2) *Input Processing:* In the front-end development, the Flask backend will process the user inputs received through interface. Based on chi square test, 10 features are selected as inputs for the machine learning model, streamlining the input process and enhancing usability.

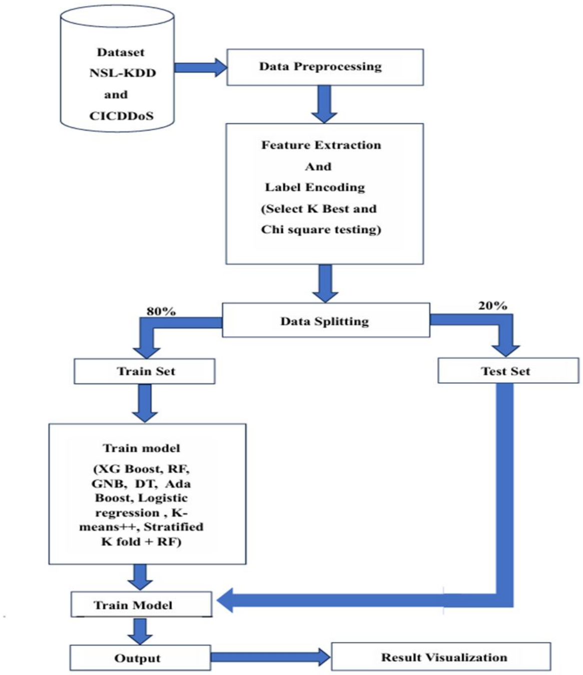

Fig.1. Dataflow diagram

V. PERFORMANCE EVALUATION

The efficiency of machine learning classifications is measured with accuracy, precision, recall and F1 score. There are 4 essential phrases needed in assessment metrics.

- True Positives (TP): Both, the predicted and actual values are Positive.
- True Negatives (TN*):* Both, the predicted and the actual values are Negative in this case.
- False Positives (FP): The actual value is Negative, but the predicted value is Positive.
- False Negatives (FN): The actual value is Positive, but the predicted value is Negative.

A. Accuracy

Accuracy is the ratio of the total number of correct predictions to that of all predictions. Accuracy gives the proposition of correct predictions out of total number of predictions.

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

B. Precision

Precision is the ratio between the total number of True Positives and that of predicted positives. Precision gives the proposition of anticipated trues are simply true.

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

C. Recall

Recall or True Positive Rate (TPR) is the ratio between the total number of True Positives and that of all relevant samples. Recall gives the proposition of truly trues are anticipated as true.

$$RECALL/TPR/SENSITIVITY = \frac{TP}{TP + FN}$$

D. F1 score

F1 score is the harmonic imply of precision and recall both.

$$F1\ Score = \frac{2 * PRECISION * RECALL}{PRECISION + RECALL}$$

VI. RESULTS AND DISCUSSIONS

The trained Random Forest and Stratified K-Fold algorithms exhibited exceptional performance in classifying attacks, boasting accuracy scores of 99.95% on the KDD dataset and 99.92% on the CICDDoS dataset. These high accuracy rates underscore the effectiveness of the models in accurately categorizing cyber threats.

Factors contributing to this success include meticulous data curation, robust algorithmic frameworks, and rigorous evaluation methodologies. Effective feature engineering and parameter optimization further enhanced the models' ability to discern subtle attack patterns.

While accuracy is a critical metric, further evaluation using complementary metrics such as precision, recall, and F1-score would provide a more comprehensive assessment of the models' capabilities.

Hence , the obtained accuracy scores demonstrate the promising potential of the trained models for practical deployment in cybersecurity, though further validation in real-world scenarios is warranted.

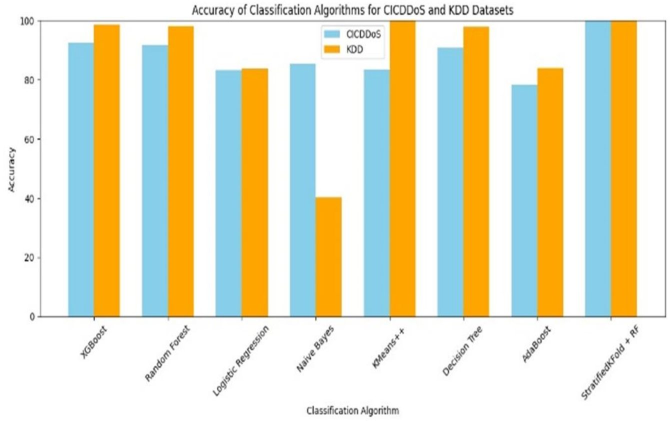

Fig.2. Graph representing comparison of accuracies of different classification algorithms

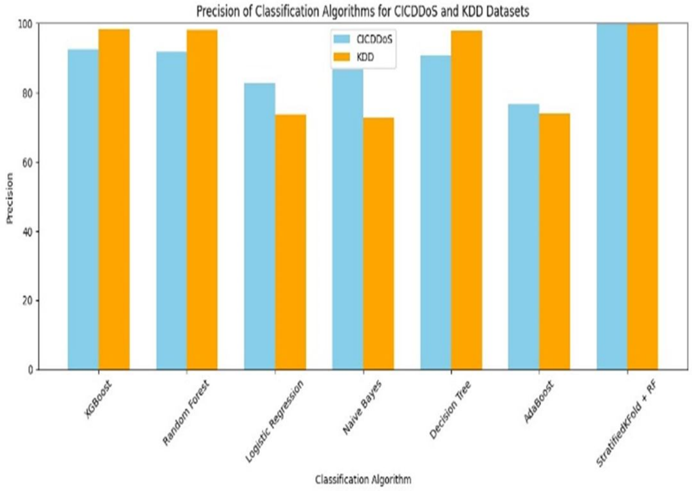

Fig.3. Graph representing the comparison of precision values of different classification algorithms

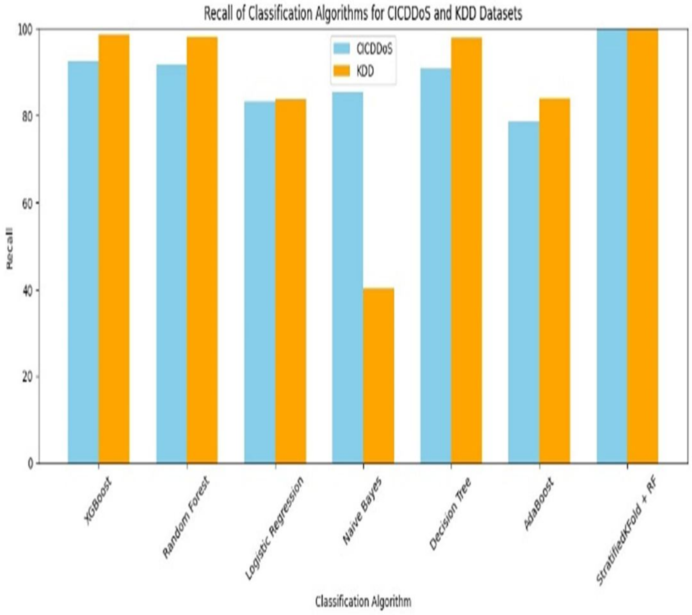

Fig.4. Graph representing the comparison of recall score of different classification algorithms

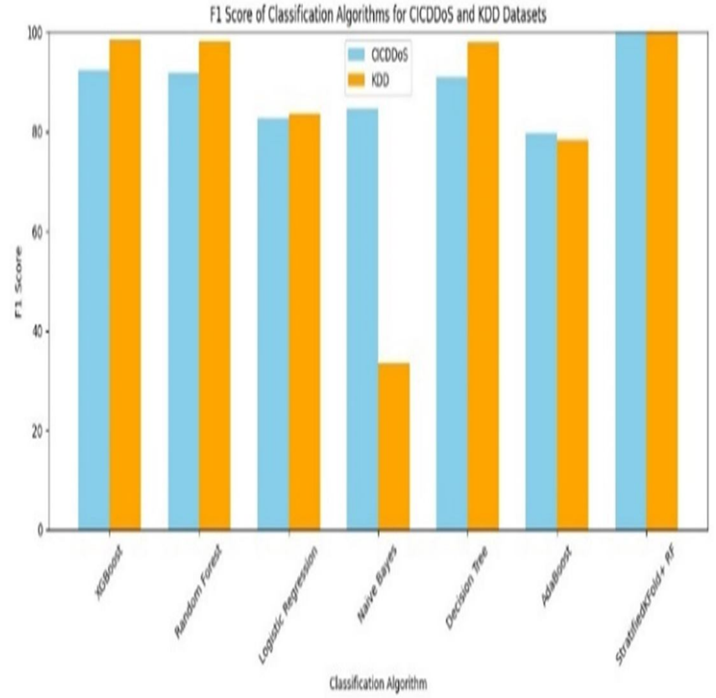

Fig.5. Graph representing the comparison of f1 scores of different classification algorithms

VII. CONCLUSIONS

In our study, we delve into the critical domain of DDoS attacks, recognized as among the most severe threats to network infrastructure. By exploring testing, analysis, and machine learning model development, we aim to enhance DDoS attack detection capabilities.

Utilizing the KDD and CICDDoS datasets, we apply a consistent methodology, leveraging Random Forest for classification and Stratified K-Fold for data splitting. Our results showcase exceptional accuracy rates of 99.95% on the KDD Dataset and 99.92% on the CICDDoS dataset. This consistency underscores the robustness and effectiveness of our approach across diverse datasets, offering promising prospects for real-world cybersecurity applications. Overall, our study contributes significantly to advancing DDoS attack detection methodologies, paving the way for improved cyber threat mitigation strategies and enhancing network infrastructure resilience against evolving threats.

VIII. ACKNOWLEDGMENT

We would like to thank few people who helped us in carrying out this project. We sincerely thank Dr. C Nataraju , Principal, JSSSTU, Mysuru and Dr. Srinath S, HOD, Department of computer science and engineering, JSSSTU, Mysuru who encouraged us at this venture.

It is our duty to thank our project supervisor Prof. Shwethashree G C for her encouragement and effective guidance throughout the project. We also thank our panel members who have corrected us in every step during Continuous Internal Evaluation.

REFERENCES

[1] M.Kia, "Early detection and mitigation of DDoS attacks in software defined networks", M.Sc. Thesis. Ryerson University, Toronto, Ontario, Canada, 2015.

[2] J.Boite, P.A.Nardin, F.Rebecchi, M.Bouet, V.Conan, "StateSec: Stateful monitoring for DDoS protection in software defined networks",IEEE Conference on Network Softwarization, Bologna,Italy, 2017.

[3] M. Myint Oo, K. Sinchai, and K. ossaporn, " Advanced support vector machine based detection for distributed denial of service attack on software defined network," Journal of Computer Networks and Communications, Volume 2019.

[4] Diash Firdaus , Rendu Munadi , Yudha Purwanto , "DDOS Attack Detection in Software Defined Network using Ensemble K-means ++ and Random Forest" , 2020 3rd International Seminar on Research of Information and Technology and Intelligent systems (ISRITI) .

[5] S.Santhosh, Dr. M.Sambath, Dr. J. Thangakumar, "Detection Of DDOS Attack using Machine Learning Models ", 2023 International Conference on Networking ans Communications (ICNWC).

[6] Rashmikiran Pandey, Mrinal Pandey, Alexey Nazarov , “Enhanced DDoS Detection using Machine Learning” , 2023 6th International Conference on Information Systems and Computer Networks (ISCON) GLA University, Mathura, India. Mar 3-4, 2023.

[7] C.M. Nalayini , Jeevaa Katiravan , “A New IDS for Detecting DDoS Attacks in Wireless Networks using Spotted Hyena Optimization and Fuzzy Temporal CNN” , Journal of Internet Technology Vol. 24 No. 1, January 2023.

[8] Aye Thandar Kyaw , May Zin Oo , Chit Su Khin , “Machine-Learning Based DDOS Attack Classifier in Software Defined Network” , 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).

[9] N. Meti, D.G Narayan,V.P Baligar,"Detection of distributed denial of service attacks using machine learning algorithms in SDN",IEEE 2017.