International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

A Novel Approach for Detection of DoS / DDoS Attack in Network Environment using Ensemble Machine Learning Model

Gottapu Sankara Rao[1], Dr P. Krishna Subbarao[2]
[1]CSE Dept
Research Scholar at JNTUK University
Visakhapatnam, India
shankar.g.p510@gmail.com
[2]CSE Dept
Research Guide , GVP College of Engg(A) , Visakhapatnam
Visakhapatnam, India
krishnasubbarao@gvpce.ac.in

Abstract—One of the most serious threat to **network security** is Denial of service (DOS) attacks. Internet and computer networks are now important parts of our businesses and daily lives. Malicious actions have become more common as our reliance on computers and communication networks has grown. Network threats are a big problem in the way people communicate today. To make sure that the networks work well and that users' information is safe, the network data must be watched and analysed to find malicious activities and attacks. Flooding may be the simplest DDoS assault. Computer networks and services are vulnerable to DoS and DDoS attacks. These assaults flood target systems with malicious traffic, making them unreachable to genuine users. The work aims to enhance the resilience of network infrastructures against these attacks and ensure uninterrupted service delivery. This research develops and evaluates enhanced DoS/DDoS detection methods. DoS attacks usually stop or slow down legal computer or network use. Denial-of-service (DoS) attacks prevent genuine users from accessing and using information systems and resources. The OSI model's layers make up the computer network. Different types of DDoS strikes target different layers. The Network Layer can be broken by using ICMP Floods or Smurf Attacks. The Transport layer can be attacked using UDP Floods, TCP Connection Exhaustion, and SYN Floods. HTTP-encrypted attacks can be used to get through to the application layer. DoS/DDoS attacks are malicious attacks. Protect network data from harm. Computer network services are increasingly threatened by DoS/DDoS attacks. Machine learning may detect prior DoS/DDoS attacks. DoS/DDoS attacks proliferate online and via social media. Network security is IT's top priority. DoS and DDoS assaults include ICMP, UDP, and the more prevalent TCP flood attacks. These strikes must be identified and stopped immediately. In this work, a stacking ensemble method is suggested for detecting DoS/DDoS attacks so that our networked data doesn't get any worse. This paper used a method called "Ensemble of classifiers," in which each class uses a different way to learn. In proposed methodology Experiment#1 , I used the Home Wifi Network Traffic Collected and generated own Dataset named it as MywifiNetwork.csv, whereas in proposed methodology Experiment#2, I used the kaggle repository “NSL-KDD benchmark dataset” to perform experiments in order to find detection accuracy of dos attack detection using python language in jupyter notebook. The system detects attack-type or legitimate-type of network traffic during detection ML classification methods are used to compare how well the suggested system works. The results show that when the ensembled stacking learning model is used, 99% of the time it is able to find the problem. In proposed methodology two Experiments are implemented for comparing detection accuracy with the existing techniques. Compared to other measuring methods, we get a big step forward in finding attacks. So, our model gives a lot of faith in securing these networks. This paper will analyse the behaviour of network traffics.

Keywords- Network, NSL-KDD, Wireshark, TCP Flood, Dataset, DoS Attack, WinPCaP.

I. INTRODUCTION

DoS and DDoS attacks try to flood target systems with malicious traffic, making them inaccessible to legitimate users [31]. DoS attacks use weaknesses in the target system or network architecture to exhaust its resources, while DDoS assaults use numerous compromised systems to coordinate their impact [31]. DDoS attacks might be volumetric, protocol-based, or application layer [1]. DoS/DDoS attacks can destroy e-commerce platforms, financial institutions, government agencies, and internet service providers' finances, reputations, and key services [31]. Thus, better DoS/DDoS detection and mitigation methods are urgently needed [1-35].[36].

Concerns about network security can be put into the three groups and it is shown in beow Figure A). These include illegal denial of service, lack of authenticity, and loss of confidentiality. People have made up a lot of different words

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

for "embezzlement" to describe different kinds of DoS attacks[7]. DDoS is an all-encompassing term that means the attack is coming from many different, unrelated places [23]. The Global Connection has cut down on the time it takes to journey, but it has also made our computer systems more vulnerable to attacks [23]. DoS attacks send a lot of hacker information to the victim's PC. The goal of the attack is to overload and crash the target server [7]. DoS attacks can get rid of rivals in the market. Attackers used DoS strikes on internet companies and asked for money to protect the victims [7]. Server flaws can be used by lone attackers to stop services [16]. A better way to find DoS/DDoS attacks has been shown [29]. Machine learning is a popular way to find DDoS attacks [1, 3, 6, 8, 13, 16, 24, 26, 28, 30, 29]. During detection, the system checks arriving packets to see if they are malicious or just normal traffic [16]. Methods for classifying compare how well a system works. Wireshark and the WinPcap Tool are used in this project to record traffic. Wireshark records the flow of network data. Common DoS/DDoS attacks types are depicted in below Figure 1). DDoS assaults are ICMP (Ping), TCP-SYNC, and UDP floods, HTTP floods [6] [16][36].

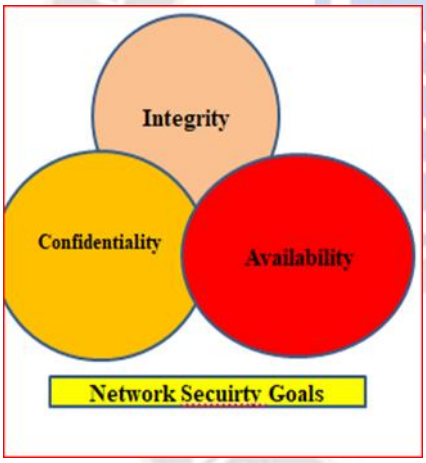

Figure A) Network security Goals/ policies

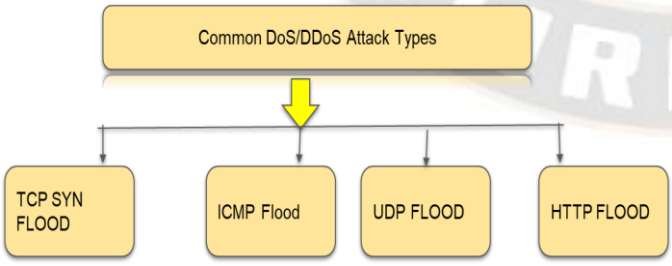

Figure 1. common DoS/DDoS attacks on Availability

A. TCP-SYN Flood Assault:

The attacker sends a lot of SYN packets to the server as part of a DOS attack called a TCP SYN flood attack [36]. But Server never completes the 3 way TCP handshake connections [1][4]. As a result the server will have a lot of "half open" connections and might not be able to service new connections for its trusted clients [1]. The Below Figure 2) and Figure 3) depicts the TCP-SYN Flood attack process.

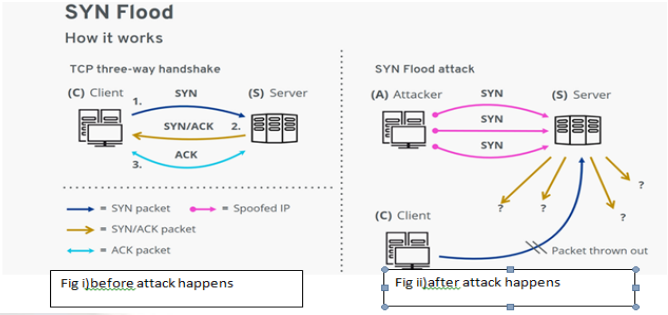

Figure 2. TCP SYN 3-Way Handshake happens

Figure 3. SYN Flood Attack happens process

B. ICMP Flood Assault:

ICMP flood [1][36] is a type of DoS attack that uses ping to send a large number of ICMP packets to a server in an attempt to crash the server and slow it down to the point where it can no longer reply to TCP/IP requests[7].

C. UDP Flood Assault:

UDP flood [1][36] is a denial-of-service (DoS/DDoS) attacks that use a lot of data. A lot of UDP Datagrams are sent to random ports on the target server that are open [4]. Administrators don't always know when ports stay open, which makes the server react. When you respond to each UDP packet with an ICMP "unreachable" message to the spoofed source IP address, you make the problem worse because you flood the network environment of the IP addresses that are being used as a spoofed source. Figure 4) below shows how a UDP flood attack works.

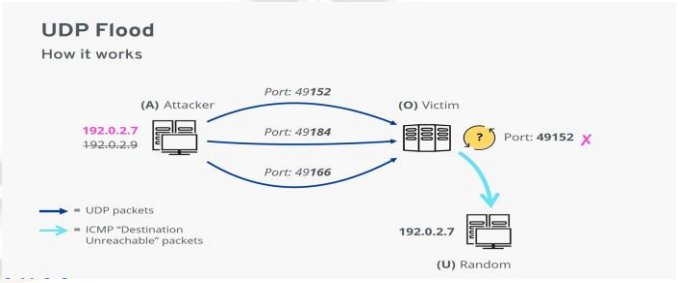

Figure 4. UDP Flood Attack happens process

D. HTTP Flood attack:

GET floods are a type of attack on the application layer that is used to flood it. HTTP request attacks happen when attackers send HTTP GET and POST requests to Web sites in an attempt to flood the server by using a lot of its resources [8].These attacks pose significant challenges to network security, and effective detection and mitigation strategies are essential to safeguard against them.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

II. LITERATURE REVIEW

AHAMED ALJUHANI [1] categorized DDoS assaults using machine and deep learning. Basant Agarwal et al. [2] discovered network anomalies using entropy and SVM. The model of YUANYUAN WEI et al. [3] has greater Precision, Recall, F1-score, and Accuracy (98.34%). Marwane Zekri and his colleagues [4] used C4.5 and a decision tree to detect DDoS. C4.5 was more accurate than other machine learning algorithms. Shreekhand Wankhede et al. utilized ML/NN. NLP and RF classify datasets as benign or dos attacks. In their deep learning-based DDoS detection approach, Xiaoyong et al. [6] reduced DeepDefense's mistake rate by 39.69%. Mohammad Tayyab and others [7] created a machine learning-based IDS for DoS and DDoS attacks. Cooperative ensemble learning and design works too. A GA and ANN helped Mehdi et al. [8] detect a DDoS attack with certainty. A machine learning-based Online DDoS tracking system by Baojun et al. [9] finds ongoing attacks. It uses spark streaming. We examined Naive Bayes, Logistic Regression, and Decision Tree. Ahmad Riza'ain Yuso and friends [10] improved intrusion detection systems by selecting features. Obaid et al. [11] demonstrated how SDN networks can detect and halt DDoS attacks using J48, RF, SVM, and KNN. J48 excelled in training and testing. Pourya et al. [12] found and described DDoS attacks statistically. They found that C2DF is faster and more accurate than past models. OMer ASLAN [13] tried a lot of different classifiers to tell the difference between DDoS activity and normal traffic and found that the way suggested worked best.Suman Nandi et al. [14] found that their way of combining methods was better than other methods at spotting DDoS attacks. This study's visualisations look at trends in multidimensional data, while Chunyuan WU et al.'s [15] study looked at DDoS attacks. It helps find out who's behind DDoS attacks. Francisco Sales de Lima Filho et al.'s [16] online smart detection method can find DoS/DDoS threats. The DR, FAR, and PREC are all made better by the random forest tree method, which sorts network data into different groups. Mateusz Kozlowski et al. [17] attacked machine learning models with very high accuracy in standard tests by using UDP DDoS attacks. Yuan Tao, among others. [18] used a method that doesn't need storage for packet analysis ahead of time or enough computer power in the router to find DDoS flooding attacks in local area networks. In this study by Subhashini Peneti et al. [19], he makes a good intrusion-based system by using feature selection.. Taking away features makes IDS faster and uses less memory. CICID2017 is a reliable and realistic dataset that Osman et al. [20] use to make a new breach detection model. Yalda Khosroshahi et al. [20] use the training dataset to check how well the classifier works. With 0.98 precision, the model can find problems with Android devices. Bao Cui-Mei et al. came up with the idea of a hierarchical SVM-based attack detection system. [21] Using the suggested method, new attacks that haven't been seen before can be found at the first level. Taeshik Shon et al. [22] say that our Enhanced SVM method was made to find and categorise new attacks in network data.DoS/DDoS attacks are a big problem for computer networks, and machine learning has been used a lot to find them. Using standard datasets [23], researchers have made a number of ML-based breach detection systems for DDoS attacks in software-defined networks (SDN). Some studies have focused on using ML in SDN to find and stop low-rate DDoS attacks [24]. Traditional ML-based DDoS detection works better in SDN [25]. Deep learning (DL) has been used to detect DDoS assaults with success [26] [27]. Support vector machine, hidden Markov model, and naïve Bayesian algorithm are ML approaches for DDoS detection [28][29]. AI and ML are commonly employed to prevent DDoS [30] [31]. ML has been used to detect DDoS attacks using network information and application behavior [28]. DDoS attacks prevent real users from utilizing the service and disrupt it. Sending many messages or requests to the central server causes it to overload and shut down [29]. DDoS attack detection uses Linear Regression, Random Forest, Support Vector Machine, Gaussian, and Nave Bayes predictors [30].A modular and adaptable security framework to detect and block LR-DDoS attacks in SDN settings was created [31].[32] Machine learning is crucial for assessing assault difficulty. focused at making smart grid network DDoS detection easier and more accurate.[34] The Random Forest classifier performs best with 99.9998%, followed by the J48 and NB classifiers with 99.9957% and 97.74%, respectively. [35] CNN model, called ResNet, over the changed dataset and looked at how well it could find the most recent DoS and DDoS strikes.

Each detection and mitigation technique has its strengths and limitations. Some of the research gaps identified are Enhanced detection techniques: Existing techniques may struggle to accurately detect sophisticated and stealthy DoS/DDoS attacks. Advancements in machine learning, data mining, and artificial intelligence can be explored to develop more robust detection algorithms that can effectively identify new and emerging attack vectors.

III. DATASET DESCRIPTION

In Proposed Methodology Experiment-1, the Dataset I made myself is called mynetwork.csv. It's what I use to do my project. It is a list of network packet flow with 504576 records. The nine characteristics shown in Table 1 are in this own dataset. I made my own dataset by using the Wireshark, WinPcap Tool interface to capture streaming messages from my home WiFi network, as shown in Figure 5 below. Wireshark is a well-known open-source network analyzer. Using the "pcap" library, Wireshark can record live Network,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

Wi-Fi, Bluetooth, and other data. The dataset contains the source address, destination address, protocol type, packet length, source port, target port, total length, packet metadata, and a mark indicating if the stream is being attacked ("1") or not ("0").The collection has 35409 TCP, 10264 ICMP, 1152 UDP, and 1983 DNS instances. Attack traffic is marked with the number 1, and regular traffic is marked with the number 0. In my Methodology Experiment -2, on the other hand, uses the NSL-KDD Dataset from the Kaggle source. The original KDD Cup 1999 dataset was better than the NSL-KDD dataset. It has information about network activity, including DoS attacks and other types of attacks. It is often used to test systems that look for intrusions. The size of the NSL-KDD dataset is 50MB, and it has 42 characteristics, 494021 cases, and a size of 50MB. NSL-KDD Dataset Features are shown in Table 2). The learned ensemble model was trained using Linear SVC, Naive Bayes, and Random forest classifiers. Network traffic protocol(packet)-wise instances are shown in Figure 9)

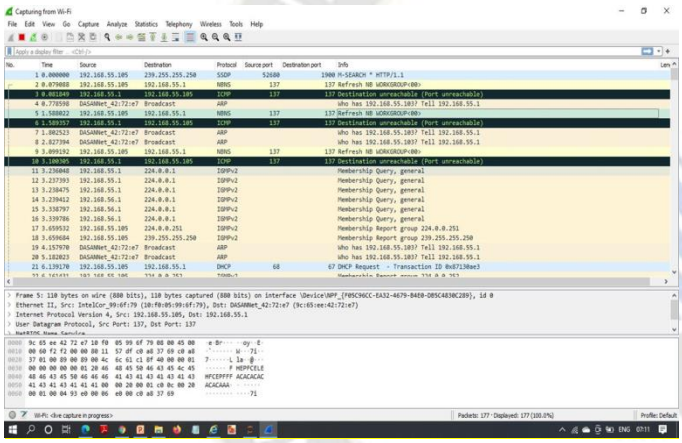

Figure 5. Wireshark Interface and Captured packets stream from wifi network for own dataset creation

Time	Source IP Address	Destination IP address	Protocol	Length	Source port	Destination Port	Total length	Packet information

TABLE I. OWN GENERATED DATASET - FEATURES

IV. METHODOLOGY

Figure 6) shows the work plans for the proposed Methodology Experiment-1 in this paper, and Figure 8) shows the work plans for the suggested Methodology Experiment-2. And here are the steps of this suggested approach-1: The suggested system can tell whether network traffic is of the DOS type or of the benign type. Table 3) compares the accuracy of detection when both methods are used in studies that are done separately from each other.

PROPOSED METHODOLOGY : (ENSEMBLE LEARNING MODEL)

Experiment #1:

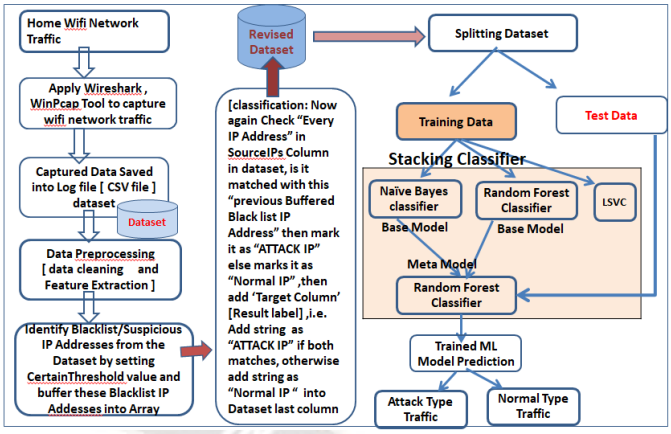

Figure 6. Proposed Methodology Experiment-1 Architecture [ensemble model]

4.1. Proposed Methodology

Experiment # 1: Steps

Ensemble learning joins the models that are learning to make the learned model better at making predictions

4.1.1: Data collection step:

In this step, API Tshark is used with Wire shark and the Win cap Tool on a home WiFi network to record the flow of messages. Traffic is saved in a log file (called a.csv file) after the packet stream is captured. This file was named mynetwork.csv and was used for my experiment. Table 1) shows the information about the Home wifi Network traffic generated dataset characteristics. Python language was used in a Jupyter notebook to do my experiment

4.1.2: Data pre-processing step:

In this step Data Cleaning should be done after above the step 4.1 csv (dataset) file, here null values in dataset are replaced with their mean and then appropriate features like "Time","Protocol", "Length", "Source_port","Destination_port" are extracted.

4.1.3: Identify blacklisted IP addresses:

This phase involves tallying the number of IP packet calls that originated from the same "Source IP Address" and ultimately reached the same "Destination IP Address." If the "source IP request count" is more than the assumed Threshold [2] value [value = 9], then the "source IP address" is classified as a "Black list IP" or "suspicious IP." In that case, we don't consider it to be a "normal IP address." In addition, the IP address will be placed to the Blacklist.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
Article Received: *15 July 2023* ***Revised:*** *04 September 2023* ***Accepted:*** *16 September 2023*

4.1.4: Classification step:

Step 4.1.3's "identified Black list IP addresses" group is checked again in step 4.13 to determine whether any of the "Source IP addresses" in the "SourceIP address" column of a dataset match any of the "identified Black list IP addresses." If the two IP addresses were the same, then label it as a "ATTACK IP," else label it as a "Normal IP." The final column of the dataset should then display the string message "ATTACK IP" or "Normal IP," depending on the desired output. We'll have a freshly compiled collection after this.

4.1.5: Split Revised Dataset:

The new 70:30 split between train and test uses the revised dataset.

4.1.6: Stacking ensemble classifier: Now, add 70 percent of the training data to the ensemble stacked classifier learning model (Naive Bayes + Random Forest classifier). This is called the "ensemble technique," and it produces a new trained model. An ensemble model is a machine learning model that combines the predictions from two or more base models to improve the suggested model's ability to predict. Figure 7 shows the basic process of how the Ensemble model works. An ML Based ensemble stacking model is a machine learning model that combines the predictions from two or more base models to improve the prediction performance of the final model.

Figure 7. Basic Ensemble Model learning Process

4.1.7: Now, give 30% of test data to test this new suggested ensemble-learned model.

4.1.8: The last step is making the final prediction. The Trained Model guesses whether network traffic is normal or an attack. For training the ensemble model, some Machine Learning methods like Linear SVC, Nave Bayes, and Random Forest were used. Table 3 shows how accurate it is to predict a DOS/DDOS attack in a network. With this, steps Experiment-1of the proposed method is done. Now we'll talk about the suggested method Experiment-2 steps.

4.2. PROPOSED METHODOLOGY:

EXPERIMENT #2 :

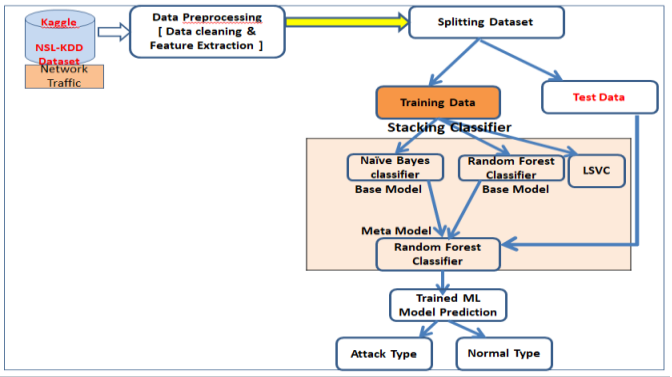

Figure 8. Proposed Methodology Experiment-2 Architecture [ensemble model]

Experiment # 2: Steps

4.2.1: Data collection step:

In this step, I use the bench mark dataset NSL-KDD dataset, which I took from the Kaggle source, for my experiment. The original KDD Cup 1999 dataset was better than the NSL-KDD dataset. It has information about network activity, including DoS attacks and other types of attacks. It is often used to test systems that look for intrusions. In Table 4.2), parts of the NSL-KDD dataset are shown.

4.2.2: Data pre-processing step:

Here, we normalize the NSL-KDD dataset by replacing missing values with their average and Categorical data is transformed into numeric data, and then features are identified based on whether or not the absolute value of the correlation coefficient is greater than a predetermined threshold.

4.2.3: Splits Dataset :

The dataset is now 70% train and 30% test. Ratios.

4.2.3: Apply Stacking ensemble classifier:

In this step, 70% of the training data is put into the ensemble stacking classifier learning model (Naive Bayes and Random Forest classifier), which is also called the "ensemble method." This makes something called the ensemble learned model, which is a new taught model.

4.2.4: Now supply 30% test data to test this ensemble model.

4.2.5: Final Prediction step:

The final phase is predicting whether network traffic is normal or attack traffic. Table 3) displays DOS/DDOS attack prediction accuracy. The Following Performance metrics [27] in Figure B. were utilized to evaluate the suggested model in this study.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
Article Received: *15 July 2023* ***Revised:*** *04 September 2023* ***Accepted:*** *16 September 2023*

$$Accuracy = \frac{TruPsv + TruNeg}{TruPsv + TruNeg + FlsPsv + FlsNeg}$$

$$Precision = \frac{TruPsv}{TruPsv + FlsPsv}$$

$$Recall = \frac{TruPsv}{TruPsv + FlsNeg}$$

$$F1 - Score = \frac{2 * precision * recall}{precision + recall}$$

Figure B) Performance Metrics

SLno	Feature	slno	feature	slno	feature
1	Duration	15	Su Attempted	29	Same Srv Rate
2	Protocol Type	16	Num Root	30	Diff Srv Rate
3	Service	17	Num File Creations	31	Srv Diff Host Rate
4	Flag	18	Num Shells	32	Dst Host Count
5	Src Bytes	19	Num Access Files	33	Dst Host Srv Count
6	Dst Bytes	20	Num Outbound Cmds	34	Dst Host Same Srv Rate
7	Land	21	Is Hot Logins	35	Dst Host Diff Srv Rate
8	Wrong Fragment	22	Is Guest Login	36	Dst Host Same Src Port Rate
9	Urgent	23	Count	37	Dst Host Srv Diff Host Rate
10	Hot	24	Srv Count	38	Dst Host Serror Rate
11	Num Failed Logins	25	Serror Rate	39	Dst Host Srv Serror Rate
12	Logged In	26	Srv Serror Rate	40	Dst Host Rerror Rate
13	Num Compromised	27	Rerror Rate	41	Dst Host Srv Rerror Rate
14	Root Shell	28	Srv Rerror Rate	42	Class

TABLE II. NSL-KDD DATASET FEATURES WITH TWO COLUMNS [FEATURE NUMBER, FEATURE NAME]

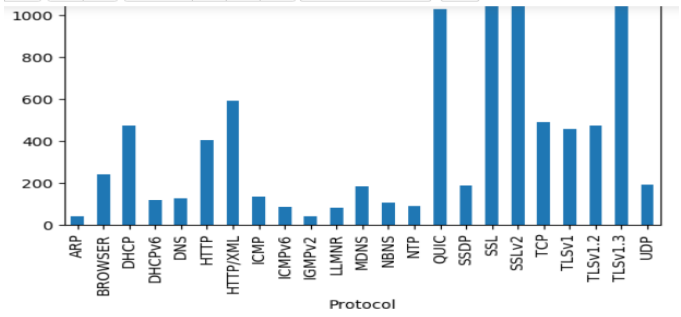

Figure 9. Network Traffic, with protocol /packet wise instances

V. EXPERIMENT RESULTS

ALGORITHM USED	Prediction Accuracy, If HomeWifi Network [own] Dataset used	Prediction Accuracy, If NSL-KDD Benchmark Dataset used
LSVC	85.23%	93.63%
NAÏVE BAYES Classifier[Gaussian]	90.5%	84.6%
RANDOM FOREST Classifier	99.60%	99.9%
Ensemble MODEL	99.62%	99.9%

TABLE III. ACCURACY DETECTION RESULTS COMPARISON

```
In [111]: plt.title("Accuracy Graph")
          plt.xlabel("MODEL")
          plt.ylabel("Prediction Accuracy")
          from matplotlib.pyplot import figure
          plt.plot(classifiers,scores)
          plt.show()
```

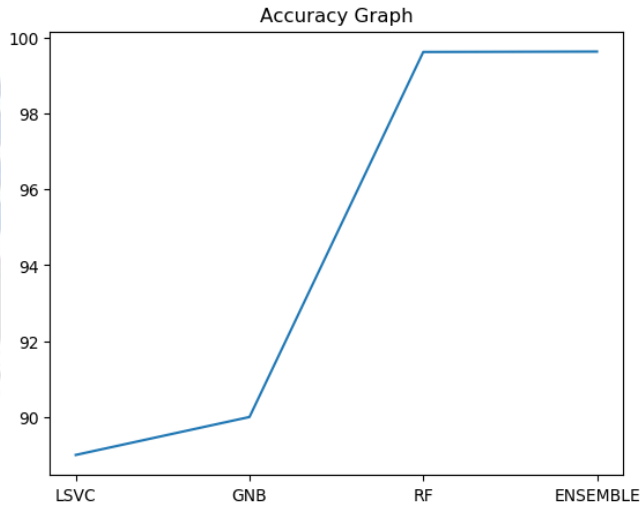

Figure 10. Accuracy Graph for different ML models in proposed Methodology Experiment-1

```
MODEL :  LinearSVC()
Accuracy of the model is:  85.72863060123007
Confusion Matrix:
 [[  4015  12613]
 [  8990 125755]]
Report:
              precision    recall  f1-score   support

           0       0.31      0.24      0.27     16628
           1       0.91      0.93      0.92    134745

    accuracy                           0.86    151373
   macro avg       0.61      0.59      0.60    151373
weighted avg       0.84      0.86      0.85    151373


==============***===============
```

Figure 11. Above Result is Linear SVC classification Report of Experiement-1

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
Article Received: *15 July 2023* ***Revised:*** *04 September 2023* ***Accepted:*** *16 September 2023*

```
==============***===============
MODEL :  GaussianNB()
Accuracy of the model is:  90.54256703639354
Confusion Matrix:
 [[  6756   9872]
 [  4444 130301]]
Report:
              precision    recall  f1-score   support

           0       0.60      0.41      0.49     16628
           1       0.93      0.97      0.95    134745

    accuracy                           0.91    151373
   macro avg       0.77      0.69      0.72    151373
weighted avg       0.89      0.91      0.90    151373


==============***===============
```

Figure 12. above Result is Naïve Bayes Algorithm Classification Report of Experiment-1

```
==============***===============
MODEL :  RandomForestClassifier()
Accuracy of the model is:  99.60759184266679
Confusion Matrix:
 [[ 16330    298]
 [   296 134449]]
Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     16628
           1       1.00      1.00      1.00    134745

    accuracy                           1.00    151373
   macro avg       0.99      0.99      0.99    151373
weighted avg       1.00      1.00      1.00    151373


==============***===============
```

Figure 13. above Result is Random forest classifier classification report of Experiement-1

```
In [32]: from sklearn.svm import LinearSVC
         svs=LinearSVC()
         svs.fit(x_train,y_train)
         y_pred=svs.predict(x_test)
         acc=accuracy_score(y_pred,y_test)
         acc

         C:\Users\GSHANKAR\AppData\Local\Progra
         near failed to converge, increase the
           ConvergenceWarning,

Out[32]: 0.9854931649190991

In [33]: from sklearn.naive_bayes import GaussianNB
         gnb=GaussianNB()
         gnb.fit(x_train,y_train)
         y_pred=gnb.predict(x_test)
         acc=accuracy_score(y_pred,y_test)
         acc

Out[33]: 0.8467943268153786
```

```
In [31]: from sklearn.metrics import accuracy_score
         from sklearn.ensemble import RandomForestClassifier
         rf=RandomForestClassifier()
         rf.fit(x_train,y_train)
         y_pred=rf.predict(x_test)
         acc=accuracy_score(y_pred,y_test)
         acc

Out[31]: 0.9993792424058405
```

ENSEMBLE MODEL

```
In [34]: #!pip install mlxtend
         from mlxtend.classifier import StackingClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.naive_bayes import GaussianNB
         from sklearn.svm import LinearSVC
         rf = RandomForestClassifier()
         gnb = GaussianNB()
         clf_stack = StackingClassifier(classifiers=[rf,gnb],meta_classifier=rf,use_probas=True,
                                        use_features_in_secondary=True)
         clf_stack.fit(x_train, y_train)
         pred_stack = clf_stack.predict(x_test)
         acc_stack = accuracy_score(y_test, pred_stack)
         acc_stack

Out[34]: 0.9993657476755327
```

Figure 14. Above results calculations represents: Accuracy calculation using LSVC, Gaussian Naïve Bayes, and Random classifier and using Ensemble Models of proposed Methodology Experiment-2

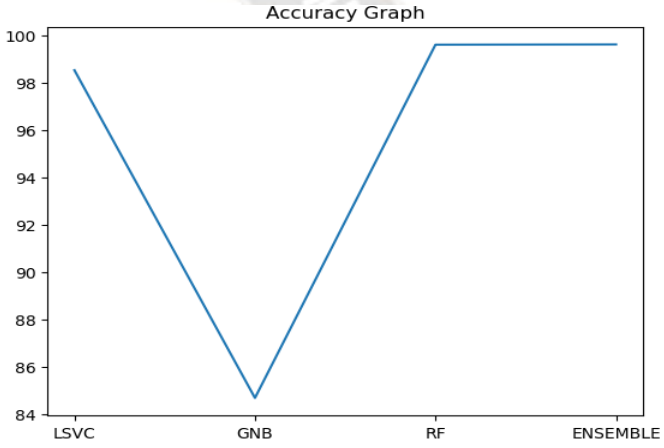

Figure 15. Accuracy Graph for all Algorithms of proposed methodology in Experiement-2

Cross-Validation:

In the next experiment, the dataset will be mixed up using the k-fold validation method to test how well the proposed method works. The data set is divided into k sections of equal size. In this work, a part of 5-fold is used.When K = 5, the data is divided into 5 folds that are roughly the same size. This makes 5 groups of data for each fold. The cross-validation test is run on each of the 5 groups of data using a 4-fold training set and a 1-fold test set. We used 5-fold cross validation to figure out how well a number of algorithms worked. It has been seen that the proposed plan works best. It is shown in below Figure 16) and Figure 17) ,Figure 18)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

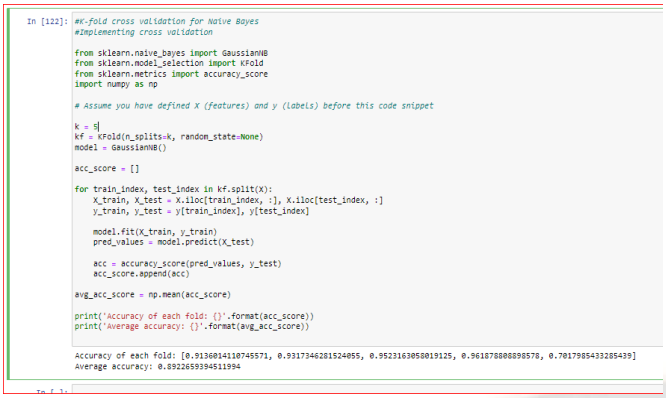

Figure 16. K-fold cross validation using Gaussian Naive Bayes

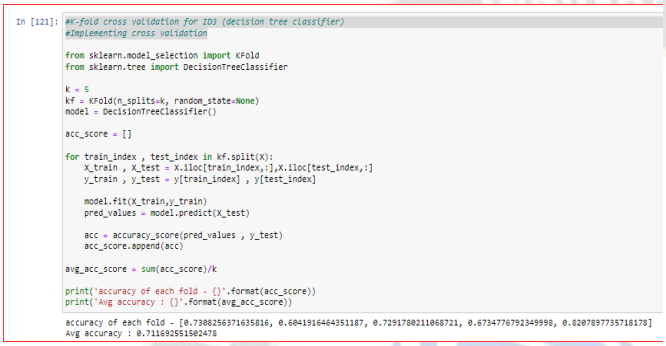

Figure 17. K-fold cross validation using Decision Tree Classifier

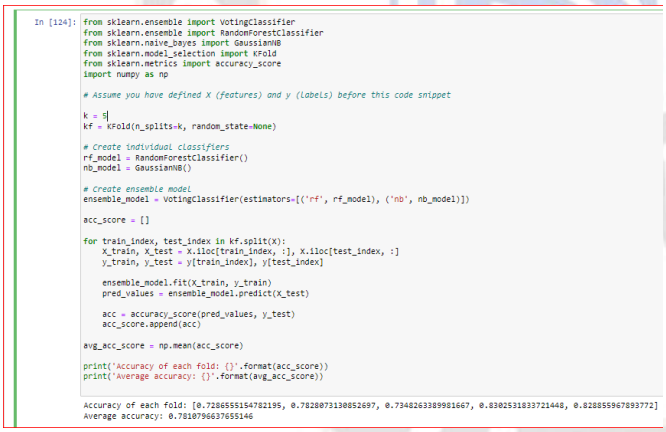

Figure 18. K-fold cross validation using Ensemble Classifier

VI. RESULTS ANALYSIS:

Experiment-1 compares the effectiveness of the proposed system to that of different Machine Learning categorization approaches. Table 3 displays the outcomes of my two experiments, whereas Figures 10--13 illustrate the effectiveness and precision of the LSVC, Naive Bayes, and Random Forest classifiers in Proposed Methodology Experiment-1. Method 1 demonstrates that the LSVC model is superior, with an accuracy of 85.7%, compared to 90.5% for the Naive Bayes model, 99% for the Random Forest algorithm, and 99.61% for the ensemble model. In contrast, Experiment-2 uses many ML techniques to evaluate the efficacy of the proposed system. The results of carrying out the second experiment suggested are depicted in Figure 14. The LSVC model achieves a 98.7% success rate, the Naive Bayes model an 84.6% success rate, the Random Forest approach a 99% success rate, and the ensemble model a 99.9% success rate. Figure 15 displays the accuracy graph for the various algorithms used in Experiment-2 of the Proposed Methodology. Figures 16), 17), and 18) display the outcomes of the cross-validation tests. The proposed model clearly provides the best results, and the suggested methods for detecting DoS/DDoS threats in network settings are effective. Networks are better able to withstand DOS/DDOS attacks because of this early identification. Moreover, it improves network security.

VII. CONCLUSION

DoS/DDOS type attacks are happening more often in spread systems like the internet and social media networks. DoS and DDoS attacks can stop computers from working, so it's important to know what causes them. DoS attack detection is also important to stop maximum damage from happening. In my project, these kinds of attacks are found by training and testing machine learning models on network traffic data. With our recommended model, which is an ensemble-learned model, Predictions with a 99% chance of coming true are feasible. The proposed method is effective enough at detecting DoS and DDoS attacks to aid in the defense of networks and prevent the utmost amount of damage from occurring. Use DL algorithms to identify DDoS/DoS-like attacks to improve the quality of this research.

REFERENCES

[1] G A. Aljuhani, “Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments,” IEEE Access, vol. 9, pp. 42236–42264, 2021, doi: 10.1109/ACCESS.2021.3062909

[2] B. Agarwal and N. Mittal, “Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques,” Procedia Technol., vol. 6, pp. 996–1003, 2012, doi: 10.1016/j.protcy.2012.10.121

[3] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, “AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification,” IEEE Access, vol. 9, pp. 146810–146821, 2021, doi: 10.1109/ACCESS.2021.3123791..

[4] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, “DDoS attack detection using machine learning techniques in cloud computing environments,” Proc. 2017 Int. Conf. Cloud Comput. Technol. Appl. CloudTech 2017, vol. 2018-Janua, pp. 1–7, 2018, doi: 10.1109/CloudTech.2017.8284731.

[5] S. Wankhede and D. Kshirsagar, “DoS Attack Detection Using Machine Learning and Neural Network,” Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018, 2018, doi: 10.1109/ICCUBEA.2018.8697702.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

[6] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS Attack via Deep Learning," 2017 IEEE Int. Conf. Smart Comput. SMARTCOMP 2017, pp. 1–8, 2017, doi: 10.1109/SMARTCOMP.2017.7946998.

[7] M. Tayyab, B. Belaton, and M. Anbar, "ICMPV6-based DOS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: A review," IEEE Access, vol. 8, pp. 170529–170547, 2020, doi: 10.1109/ACCESS.2020.3022963.

[8] M. Barati, A. Abdullah, N. I. Udzir, and ..., "Distributed Denial of Service detection using hybrid machine learning technique," Biometrics …, pp. 268–273, 2014, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7013133/

[9] B. Zhou, J. Li, J. Wu, S. Guo, Y. Gu, and Z. Li, "Machine-learning-based online distributed denial-of-service attack detection using spark streaming," IEEE Int. Conf. Commun., vol. 2018-May, 2018, doi: 10.1109/ICC.2018.8422327.

[10] A. R. A. Yusof, N. I. Udzir, A. Selamat, H. Hamdan, and M. T. Abdullah, "Adaptive feature selection for denial of services (DoS) attack," 2017 IEEE Conf. Appl. Inf. Netw. Secur. AINS 2017, vol. 2018-Janua, pp. 81–84, 2017, doi: 10.1109/AINS.2017.8270429.

[11] O. Rahman, M. A. G. Quraishi, and C. H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," Proc. - 2019 IEEE World Congr. Serv. Serv. 2019, vol. 2642–939X, pp. 184–189, 2019, doi: 10.1109/SERVICES.2019.00051.

[12] P. Shamsolmoali and M. Zareapoor, "Statistical-based filtering system against DDOS attacks in cloud computing," Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014, pp. 1234–1239, 2014, doi: 10.1109/ICACCI.2014.6968282.

[13] Ö. ASLAN, "A Methodology to Detect Distributed Denial of Service Attacks," Bilişim Teknol. Derg., vol. 15, no. 2, pp. 149–158, 2022, doi: 10.17671/gazibtd.1002178.

[14] S. Nandi, S. Phadikar, and K. Majumder, "Detection of DDoS Attack and Classification Using a Hybrid Approach," ISEA-ISAP 2020 - Proc. 3rd ISEA Int. Conf. Secur. Priv. 2020, pp. 41–47, 2020, doi: 10.1109/ISEA-ISAP49340.2020.234999.

[15] S. Sheng, C. Wu, and X. Dong, "Research on Visualization Systems for DDoS Attack Detection," Proc. - 2018 IEEE Int. Conf. Syst. Man, Cybern. SMC 2018, pp. 2986–2991, 2019, doi: 10.1109/SMC.2018.00507.

[16] F. S. De Lima Filho, F. A. F. Silveira, A. De Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, "Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning," Secur. Commun. Networks, vol. 2019, 2019, doi: 10.1155/2019/1574749.

[17] M. Kozlowski and B. Ksiezopolski, "A new method of testing machine learning models of detection for targeted DDoS attacks," Proc. 18th Int. Conf. Secur. Cryptogr. SECRYPT 2021, no. Secrypt, pp. 728–733, 2021, doi: 10.5220/0010574507280733.

[18] Y. Tao and S. Yu, "DDoS attack detection at local area networks using information theoretical metrics," Proc. - 12th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2013, pp. 233–240, 2013, doi: 10.1109/TrustCom.2013.32.

[19] S. Peneti and Hemalatha, "DDOS Attack Identification using Machine Learning Techniques," 2021 Int. Conf. Comput. Commun. Informatics, ICCCI 2021, 2021, doi: 10.1109/ICCCI50826.2021.9402441.

[20] Y. Khosroshahi and E. Ozdemir, "Detection of Sources Being Used in DDoS Attacks," Proc. - 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. CSCloud 2019 5th IEEE Int. Conf. Edge Comput. Scalable Cloud, EdgeCom 2019, pp. 163–168, 2019, doi: 10.1109/CSCloud/EdgeCom.2019.000-1.

[21] C. M. Bao, "Intrusion detection based on one-class SVM and SNMP MIB data," 5th Int. Conf. Inf. Assur. Secur. IAS 2009, vol. 2, pp. 346–349, 2009, doi: 10.1109/IAS.2009.124.

[22] T. Shon, Y. Kim, C. Lee, and J. Moon, "A machine learning framework for network anomaly detection using SVM and GA," Proc. from 6th Annu. IEEE Syst. Man Cybern. Inf. Assur. Work. SMC 2005, vol. 2005, pp. 176–183, 2005, doi: 10.1109/IAW.2005.1495950.

[23] M. Elsayed, N. Le-Khac, B. Dev, and R. Jurcut, "DDoSNet: A Deep-Learning Model for Detecting Network Attacks," in Proceedings of the 2020 21st International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1-6, 2020. DOI: 10.1109/wowmom49955.2020.00072.

[24] Perez-Diaz, Valdovinos, Choo, Zhu (2020). A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning. IEEE Access, (8), 155859-155872.https://doi.org/10.1109/access.2020.3019330

[25] Shen (2022). An Intrusion Detection Algorithm for DDoS Attacks Based on DBN and Three-way Decisions. J. Phys.: Conf. Ser., 1(2356), 012044. https://doi.org/10.1088/1742-6596/2356/1/012044

[26] Shieh, Nguyen, Lin, Lai, Horng, Miu (2022). Detection of Adversarial DDoS Attacks Using Symmetric Defense Generative Adversarial Networks. Electronics, 13(11), 1977. https://doi.org/10.3390/electronics11131977

[27] Silivery, "An Effective Deep Learning Based Multi-Class Classification of DoS and DDoS Attack Detection," Int. J. Electr. Comput. Eng. Syst., vol. 4, no. 14, pp. 421-431, 2023. DOI: 10.32985/ijeces.14.4.6.

[28] J. Jyothi, Z. Wang, S. Addepalli, and R. Karri, "BRAIN: BehavioR Based Adaptive Intrusion Detection in Networks: Using Hardware Performance Counters to Detect DDoS Attacks," in Proceedings of the 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), pp. 416-421, 2016. doi: 10.1109/vlsid.2016.115.

[29] M. A. Al-Shareeda, S. Manickam, and M. A. Saare, "DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison," IEEE Transactions on Network and Service Management, vol. 12, no. 2, pp. 930-939, Apr. 2023. ISSN: 2302-9285. DOI: 10.11591/eei.v12i2.4466.

[30] T. Ulemale, "Review on Detection of DDOS Attack using Machine Learning," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 10, no. 3, pp. 764-764, Mar. 2022. ISSN: 2321-9653. Available: www.ijraset.com

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
DOI: https://doi.org/10.17762/ijritcc.v11i9.8340
***Article Received:** 15 July 2023 **Revised:** 04 September 2023 **Accepted:** 16 September 2023*

[31] J. Perez-Diaz, I. Valdovinos, K. Choo, D. Zhu, "A Flexible Sdn-based Architecture For Identifying and Mitigating Low-rate Ddos Attacks Using Machine Learning", IEEE Access, vol. 8, p. 155859-155872, 2020. https://doi.org/10.1109/access.2020.3019330

[32] Reddy, A. ., & Waheeb , M. Q. . (2022). Enhanced Pre-Processing Based Cardiac Valve Block Detection Using Deep Learning Architectures. Research Journal of Computer Systems and Engineering, 3(1), 84–89. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/47

[33] M. Arshi, M. D. Nasreen, and K. Madhavi, "A survey of DDoS attacks using machine learning techniques," in E3S Web of Conferences, vol. 184, p. 01052, EDP Sciences, 2020. DOI: 10.1051/e3sconf/202018401052.

[34] S. Ali and Y. Li, "Learning Multilevel Auto-Encoders for DDoS Attack Detection in Smart Grid Network," IEEE Access, vol. 7, pp. 105174-105183, 2019. DOI: 10.1109/ACCESS.2019.2933304.

[35] F. A. Alhaidari, "New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning," in Proceedings of the 2019 5th International Conference on Control, Decision and Information Technologies (CoDIT), pp. 894-898, 2019.
DOI: 10.1109/CoDIT.2019.8820477.

[36] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in 2020 IEEE 23rd International Multitopic Conference (INMIC), pp. 1-6, IEEE, 2020. DOI: 10.1109/INMIC50486.2020.9318216.

[37] G. Sankara Rao et al., "Security Attacks DoS/DDoS attack Detection in Networks," NeuroQuantology, vol. 20, no. 11, pp. 8452-8463, Sep. 2022, doi: 10.48047/nq.2022.20.11.NQ66839.