DOS ATTACK DETECTION USING MACHINE LEARNING APPROACH

-Submitted by

1. Soumyaraj Roy (19BCE0200)

2. Kaustubh Dwivedi (19BCE0249)

3. Rebello Nishma Avalon (19BCE0253)

4. Saatvik Sharma (19BCE0283)

5. Tanmay Misra (19BCE0310)

in partial fulfilment for the award of the degree of

BACHELORS OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

APRIL, 2022

TABLE OF CONTENTS

ABSTRACT

One of the most hazardous attack in the area of network security is a distributed denial of service (DDoS) attack. A DDoS assault disrupts the normal operation of essential services in a variety of online applications. Rather than offering services to actual users, systems facing DDoS attacks are kept occupied with fake requests (Bots). These attacks are becoming increasingly sophisticated and therefore are increasing on a daily basis.

As a result, detecting these threats and protecting online businesses from them has grown harder. To address this issue we have developed a machine learning-based solution for detecting and classifying distinct types of network traffic. The suggested method is tested on a fresh dataset that includes a combination of different sorts of attacks such as DoS, HTTP floods and normal traffic. To recognise abnormal activity we used a range of machine learning algorithms including neural networks, Logistic Regression, and Decision Trees. Results showed hat random forest and decision tree have achieved higher accuracy compared to rest of the models.

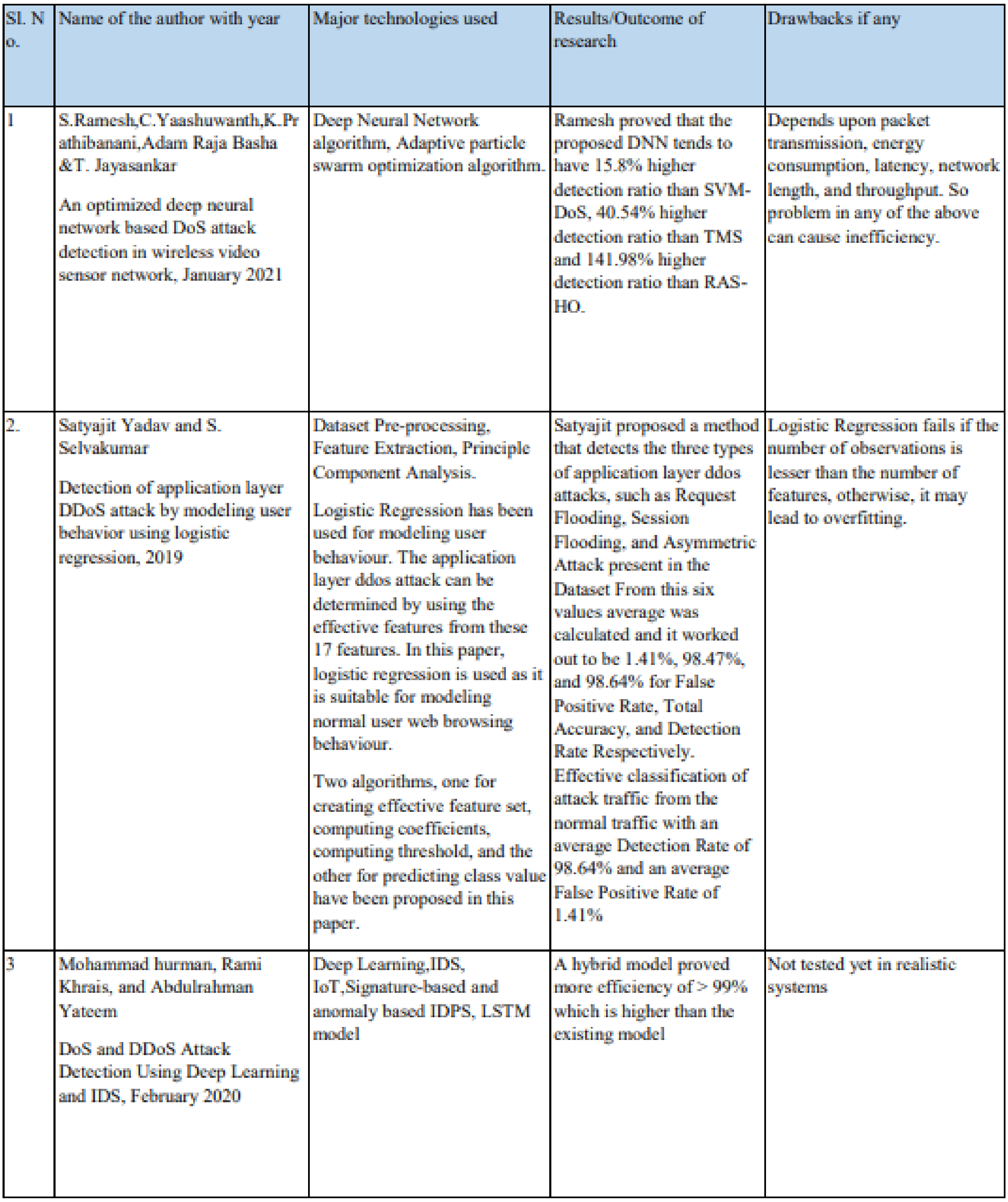
INTRODUCTION

In recent years, Internet services have become indispensable for both businesses and individuals. With the rising demand for network-based services, network hackers have escalated their attacks on these services in order to prevent genuine customers from receiving service. DDoS assaults are defined as attacks that halt or slows network services. Attackers launch a DDoS attack by taking control of millions of publicly accessible computers over the internet. As a result, servers block legitimate users' requests and remain busy with the attack-generated requests.

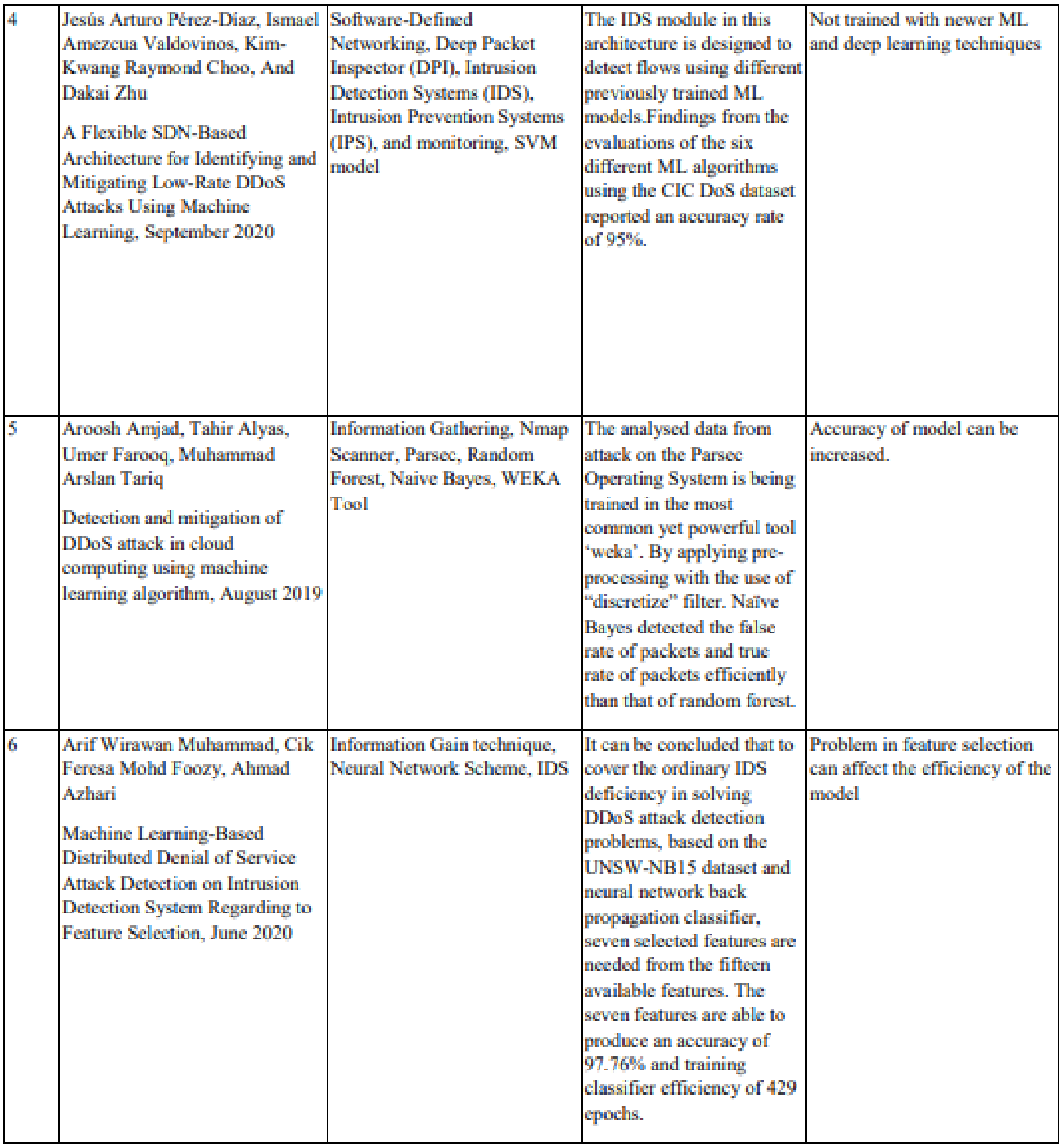
These attacks are more likely to target well-known websites, such as banks, social networking sites, and colleges. To safeguard essential data and services from attackers, more than one technology such as antivirus software, firewalls, and intrusion detection systems (IDS) should be employed in a computer network. An IDS is one of the most often utilised solutions for dealing with DDoS attacks. The confidentiality, integrity and availability of network resources are all safeguarded by an IDS.

In order to detect and then classify distinct DDoS attempts and eliminate intrusion, the IDS system employs machine learning techniques. However, achieving 100% performance accuracy is very difficult. Various publicly available datasets are outdated, lacking newer and more recent types of attack traffic such as Smurf, SIDDOS and HITP flood. As a result, a dataset was required that included a variety of new attacks. Earlier we has used KDD Cup 1999 Data, which included four different attacks: SIDDoS, H IQflood, UDP-flood, and Smurf. We later went on to the NF-UQ-NIDS dataset accessible on cloudshare because the previous dataset was older.

LITERATURE REVIEW:

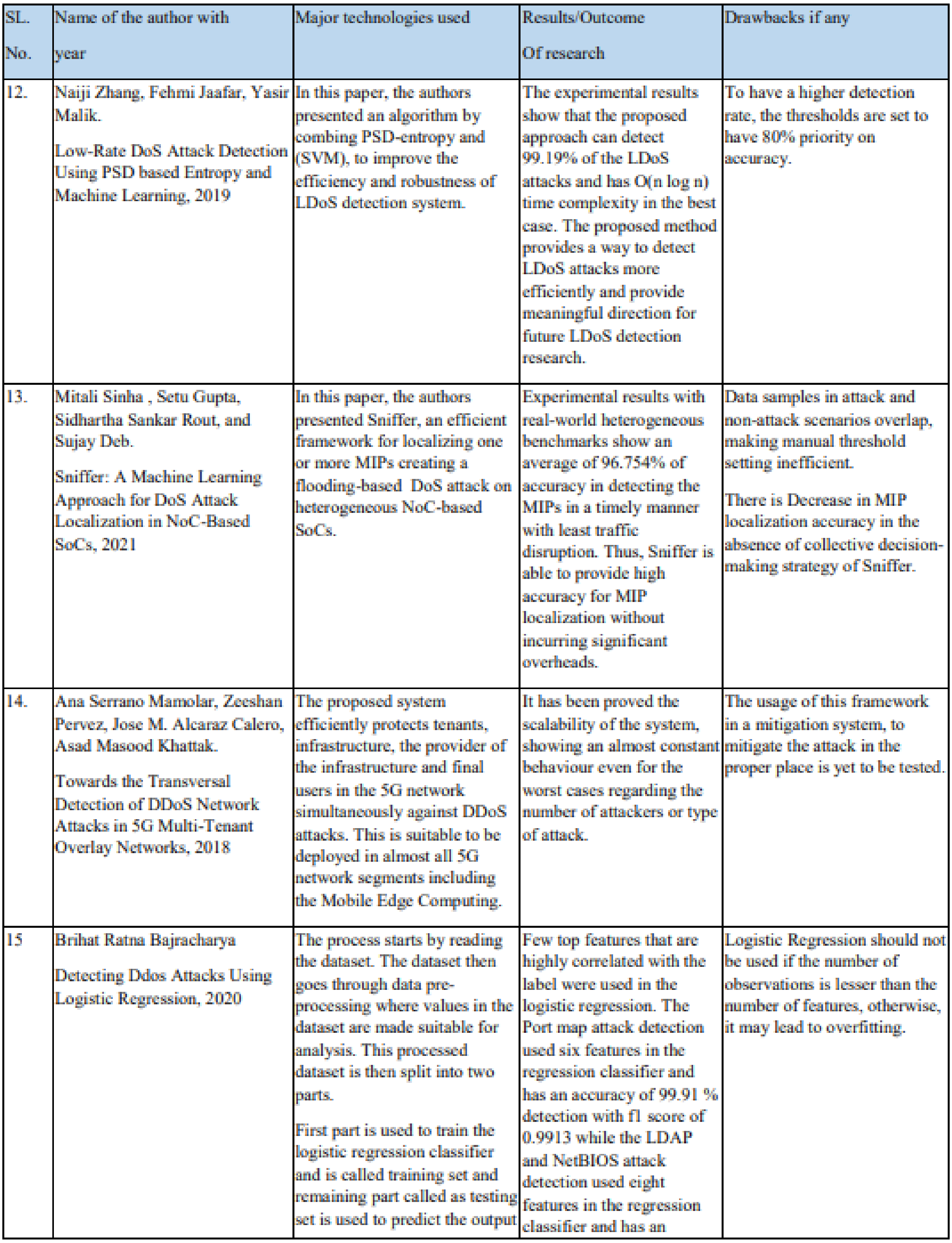


Sl. No.	Name of the author with year	Major technologies used	Results/Outcome of research	Drawbacks if any
1	S.Ramesh,C.Yaashuwanth,K.Prathibanani,Adam Raja Basha &T. Jayasankar An optimized deep neural network based DoS attack detection in wireless video sensor network, January 2021	Deep Neural Network algorithm, Adaptive particle swarm optimization algorithm.	Ramesh proved that the proposed DNN tends to have 15.8% higher detection ratio than SVM-DoS, 40.54% higher detection ratio than TMS and 141.98% higher detection ratio than RAS-HO.	Depends upon packet transmission, energy consumption, latency, network length, and throughput. So problem in any of the above can cause inefficiency.
2.	Satyajit Yadav and S. Selvakumar Detection of application layer DDoS attack by modeling user behavior using logistic regression, 2019	Dataset Pre-processing, Feature Extraction, Principle Component Analysis. Logistic Regression has been used for modeling user behaviour. The application layer ddos attack can be determined by using the effective features from these 17 features. In this paper, logistic regression is used as it is suitable for modeling normal user web browsing behaviour. Two algorithms, one for creating effective feature set, computing coefficients, computing threshold, and the other for predicting class value have been proposed in this paper.	Satyajit proposed a method that detects the three types of application layer ddos attacks, such as Request Flooding, Session Flooding, and Asymmetric Attack present in the Dataset From this six values average was calculated and it worked out to be 1.41%, 98.47%, and 98.64% for False Positive Rate, Total Accuracy, and Detection Rate Respectively. Effective classification of attack traffic from the normal traffic with an average Detection Rate of 98.64% and an average False Positive Rate of 1.41%	Logistic Regression fails if the number of observations is lesser than the number of features, otherwise, it may lead to overfitting.
3	Mohammad hurman, Rami Khrais, and Abdulrahman Yateem DoS and DDoS Attack Detection Using Deep Learning and IDS, February 2020	Deep Learning,IDS, IoT,Signature-based and anomaly based IDPS, LSTM model	A hybrid model proved more efficiency of > 99% which is higher than the existing model	Not tested yet in realistic systems

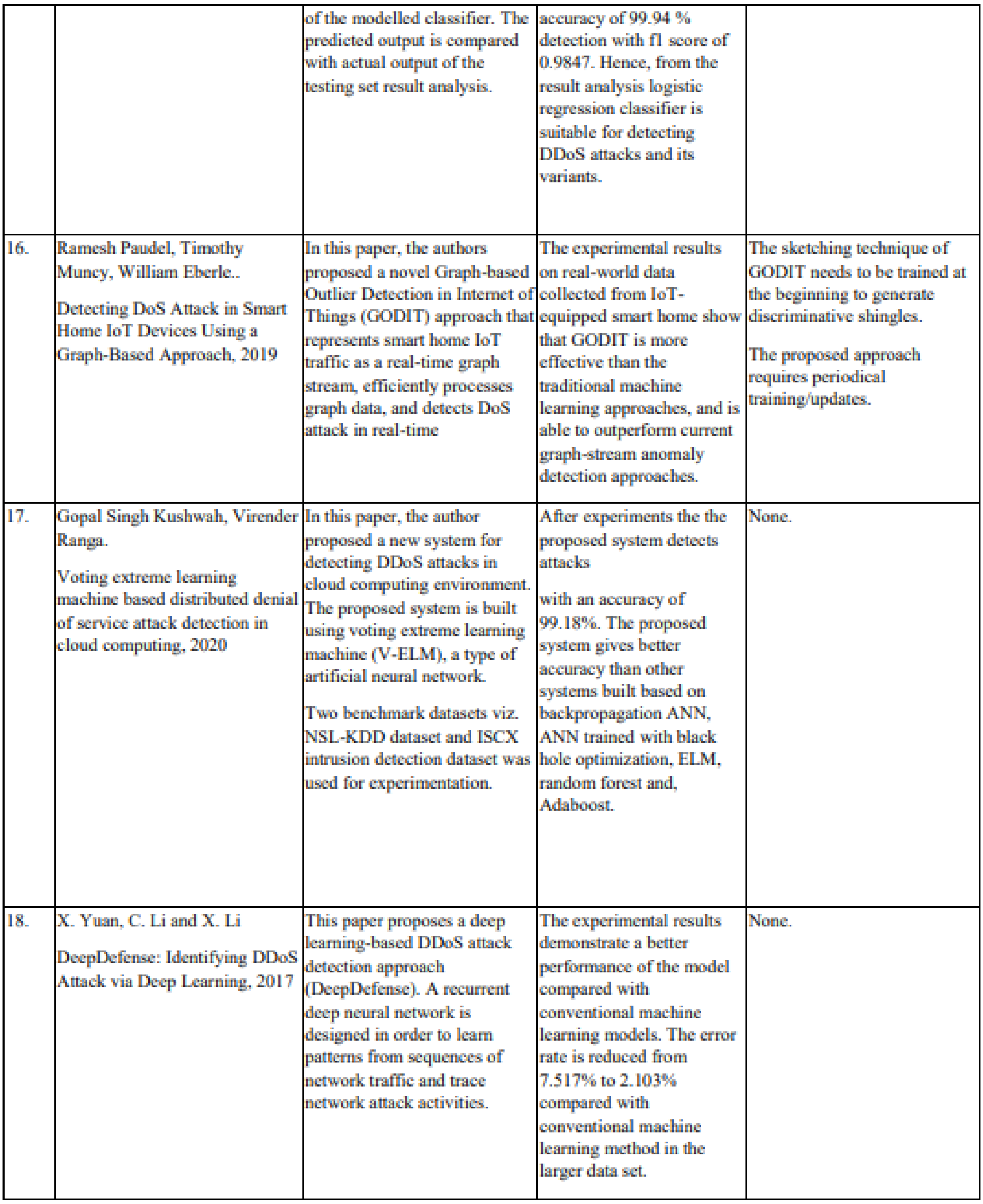


4	Jesús Arturo Pérez-Diaz, Ismael Amezcua Valdovinos, Kim-Kwang Raymond Choo, And Dakai Zhu A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning, September 2020	Software-Defined Networking, Deep Packet Inspector (DPI), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), and monitoring, SVM model	The IDS module in this architecture is designed to detect flows using different previously trained ML models.Findings from the evaluations of the six different ML algorithms using the CIC DoS dataset reported an accuracy rate of 95%.	Not trained with newer ML and deep learning techniques
5	Aroosh Amjad, Tahir Alyas, Umer Farooq, Muhammad Arslan Tariq Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm, August 2019	Information Gathering, Nmap Scanner, Parsec, Random Forest, Naive Bayes, WEKA Tool	The analysed data from attack on the Parsec Operating System is being trained in the most common yet powerful tool 'weka'. By applying pre-processing with the use of "discretize" filter. Naïve Bayes detected the false rate of packets and true rate of packets efficiently than that of random forest.	Accuracy of model can be increased.
6	Arif Wirawan Muhammad, Cik Feresa Mohd Foozy, Ahmad Azhari Machine Learning-Based Distributed Denial of Service Attack Detection on Intrusion Detection System Regarding to Feature Selection, June 2020	Information Gain technique, Neural Network Scheme, IDS	It can be concluded that to cover the ordinary IDS deficiency in solving DDoS attack detection problems, based on the UNSW-NB15 dataset and neural network back propagation classifier, seven selected features are needed from the fifteen available features. The seven features are able to produce an accuracy of 97.76% and training classifier efficiency of 429 epochs.	Problem in feature selection can affect the efficiency of the model

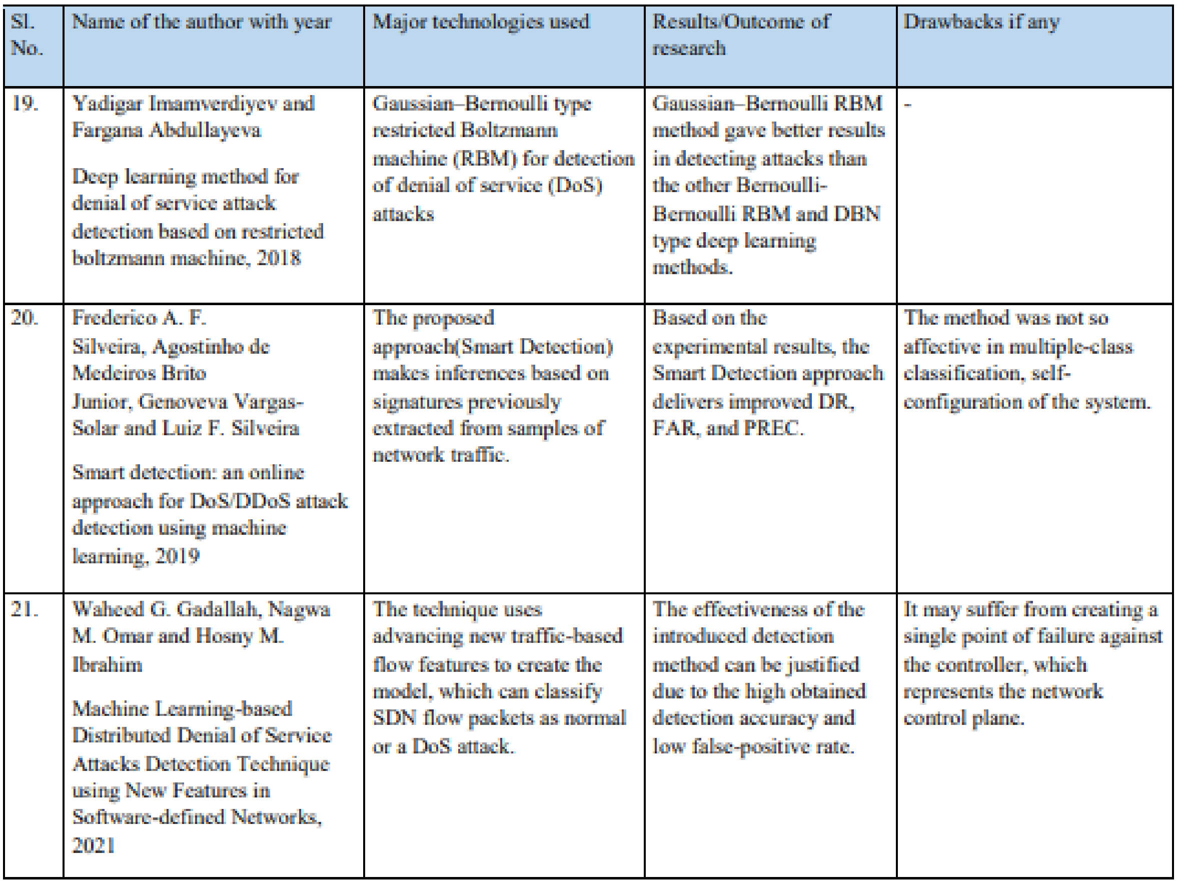
7	S.B.Gopal, C.Poongodi, D.Nanthiya, R.Snega Priya, G.Saran, M.Sathya Priya. Mitigating DoS attacks in IoT using Supervised and Unsupervised Algorithms, 2020	Intrusion Detection System (IDS), Support Vector Machine(SVM), Deep feed forward networks (DFF).	SVM and DFF have been evaluated to demonstrate the feasibility of applying these algorithms. DFF can classify the data with a higher accuracy. SVM is an appropriate choice for faster classification method.	Other external factors and maybe something amiss that may have contributed, so more in-depth research and testing is needed to get better results
8	Swathi Sambangi and Lakshmeeswari Gondi. A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression, 2020	We can use the regression analysis technique by applying one of its important variants known as multiple linear regression analysis. The research objective behind this study is to build a machine learning model that is an ensemble of feature selection using information gain and regression analysis.	Swathi observed that through this ensemble model for Friday morning dataset, a prediction accuracy of 97.86% is achieved. Similarly, for the Friday afternoon log file, the prediction accuracy is obtained as 73.79% for 16 attributes obtained through information gain-based feature selection and regression analysis-based ML model.	Limited analysis for one-day log file and in future, this research may be extended to consider all traffic log files of five days and come out with a consensus-based machine learning model
9	Manjula Suresh and R. Anitha. Evaluating Machine Learning Algorithms for Detecting DDoS Attacks	Various machine learning algorithms employed in the proposed framework. Naive Bayes, K-Mean Clustering SVM, KNN classifier KFCM Clustering	rimental results show that Fuzzy c-means clustering gives better classification and it is fast compared to the other algorithms.	Correct Classificatoin rate is approximately equal for all models making one difficult to choose
10.	Tong Anh Tuan, Hoang Viet Long, Le Hoang Son, Raghvendra Kumar, Ishaani Priyadarshini & Nguyen Thi Kim Son Performance Evaluation of Botnet DDoS attack detection using machine learning, 2020	Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), and Unsupervised Learning (USML) (K means, X-means etc.) were proposed.	This paper performed an experimental analysis of the machine learning methods for Botnet DDoS attack detection. The evaluation is done on the UNBS-NB 15 and KDD99 which are well-known publicity datasets for Botnet DDoS attack detection.	None.
11.	Jiangtao Pei, Yunli Chen, Wei Ji A DDoS Attack Detection Method Based on Machine Learning, June 2019	The characteristics of the attack traffic obtained in the model detection phase are trained in the training model based on the random forest algorithm. Finally, the test model is validated by the DDoS attack, and the SVM method in the machine learning is compared in terms of detection accuracy.	The experimental results show that the proposed DDoS attack detection method based on machine learning has a good detection rate for the current popular DDoS attacks.	None

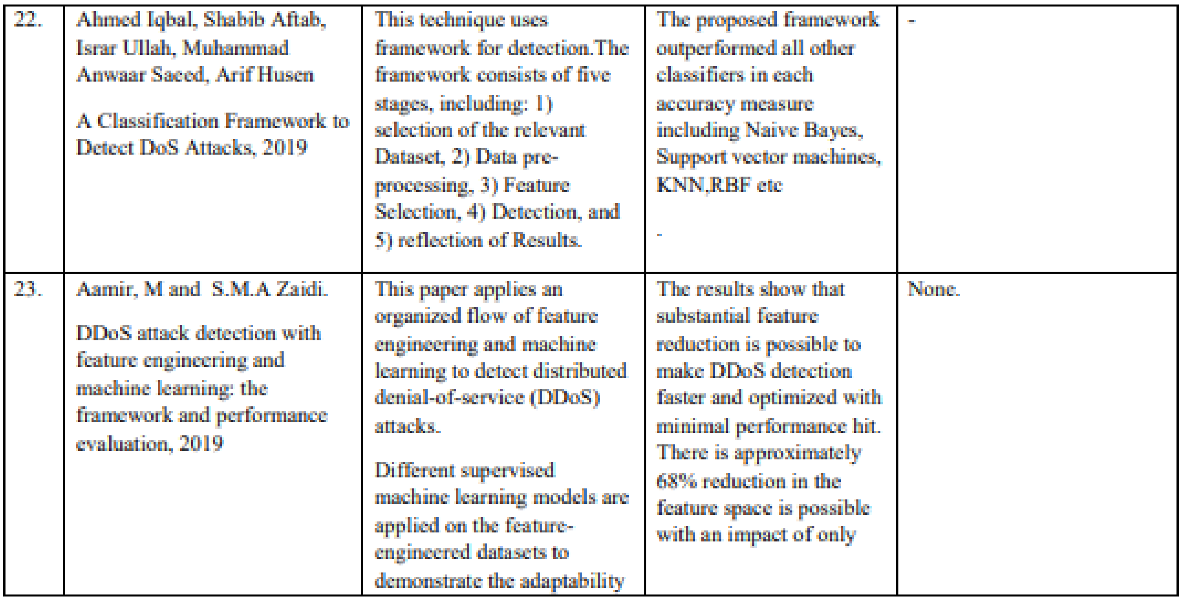


SL. No.	Name of the author with year	Major technologies used	Results/Outcome Of research	Drawbacks if any
12.	Naiji Zhang, Fehmi Jaafar, Yasir Malik. Low-Rate DoS Attack Detection Using PSD based Entropy and Machine Learning, 2019	In this paper, the authors presented an algorithm by combing PSD-entropy and (SVM), to improve the efficiency and robustness of LDoS detection system.	The experimental results show that the proposed approach can detect 99.19% of the LDoS attacks and has O(n log n) time complexity in the best case. The proposed method provides a way to detect LDoS attacks more efficiently and provide meaningful direction for future LDoS detection research.	To have a higher detection rate, the thresholds are set to have 80% priority on accuracy.
13.	Mitali Sinha , Setu Gupta, Sidhartha Sankar Rout, and Sujay Deb. Sniffer: A Machine Learning Approach for DoS Attack Localization in NoC-Based SoCs, 2021	In this paper, the authors presented Sniffer, an efficient framework for localizing one or more MIPs creating a flooding-based DoS attack on heterogeneous NoC-based SoCs.	Experimental results with real-world heterogeneous benchmarks show an average of 96.754% of accuracy in detecting the MIPs in a timely manner with least traffic disruption. Thus, Sniffer is able to provide high accuracy for MIP localization without incurring significant overheads.	Data samples in attack and non-attack scenarios overlap, making manual threshold setting inefficient. There is Decrease in MIP localization accuracy in the absence of collective decision-making strategy of Sniffer.
14.	Ana Serrano Mamolar, Zeeshan Pervez, Jose M. Alcaraz Calero, Asad Masood Khattak. Towards the Transversal Detection of DDoS Network Attacks in 5G Multi-Tenant Overlay Networks, 2018	The proposed system efficiently protects tenants, infrastructure, the provider of the infrastructure and final users in the 5G network simultaneously against DDoS attacks. This is suitable to be deployed in almost all 5G network segments including the Mobile Edge Computing.	It has been proved the scalability of the system, showing an almost constant behaviour even for the worst cases regarding the number of attackers or type of attack.	The usage of this framework in a mitigation system, to mitigate the attack in the proper place is yet to be tested.
15	Brihat Ratna Bajracharya Detecting Ddos Attacks Using Logistic Regression, 2020	The process starts by reading the dataset. The dataset then goes through data pre-processing where values in the dataset are made suitable for analysis. This processed dataset is then split into two parts. First part is used to train the logistic regression classifier and is called training set and remaining part called as testing set is used to predict the output	Few top features that are highly correlated with the label were used in the logistic regression. The Port map attack detection used six features in the regression classifier and has an accuracy of 99.91 % detection with f1 score of 0.9913 while the LDAP and NetBIOS attack detection used eight features in the regression classifier and has an	Logistic Regression should not be used if the number of observations is lesser than the number of features, otherwise, it may lead to overfitting.



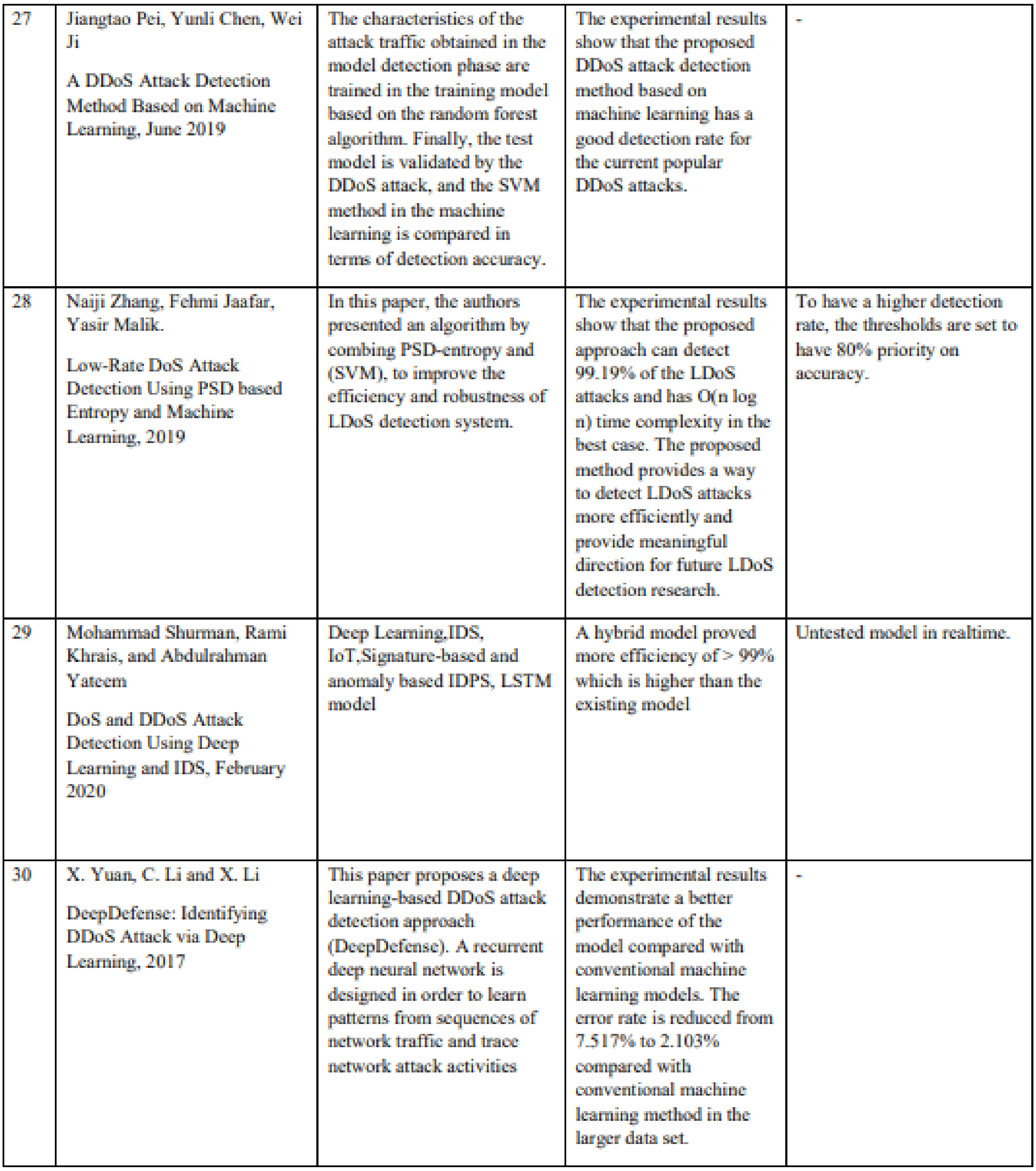
		of the modelled classifier. The predicted output is compared with actual output of the testing set result analysis.	accuracy of 99.94 % detection with f1 score of 0.9847. Hence, from the result analysis logistic regression classifier is suitable for detecting DDoS attacks and its variants.	
16.	Ramesh Paudel, Timothy Muncy, William Eberle.. Detecting DoS Attack in Smart Home IoT Devices Using a Graph-Based Approach, 2019	In this paper, the authors proposed a novel Graph-based Outlier Detection in Internet of Things (GODIT) approach that represents smart home IoT traffic as a real-time graph stream, efficiently processes graph data, and detects DoS attack in real-time	The experimental results on real-world data collected from IoT-equipped smart home show that GODIT is more effective than the traditional machine learning approaches, and is able to outperform current graph-stream anomaly detection approaches.	The sketching technique of GODIT needs to be trained at the beginning to generate discriminative shingles. The proposed approach requires periodical training/updates.
17.	Gopal Singh Kushwah, Virender Ranga. Voting extreme learning machine based distributed denial of service attack detection in cloud computing, 2020	In this paper, the author proposed a new system for detecting DDoS attacks in cloud computing environment. The proposed system is built using voting extreme learning machine (V-ELM), a type of artificial neural network. Two benchmark datasets viz. NSL-KDD dataset and ISCX intrusion detection dataset was used for experimentation.	After experiments the the proposed system detects attacks with an accuracy of 99.18%. The proposed system gives better accuracy than other systems built based on backpropagation ANN, ANN trained with black hole optimization, ELM, random forest and, Adaboost.	None.
18.	X. Yuan, C. Li and X. Li DeepDefense: Identifying DDoS Attack via Deep Learning, 2017	This paper proposes a deep learning-based DDoS attack detection approach (DeepDefense). A recurrent deep neural network is designed in order to learn patterns from sequences of network traffic and trace network attack activities.	The experimental results demonstrate a better performance of the model compared with conventional machine learning models. The error rate is reduced from 7.517% to 2.103% compared with conventional machine learning method in the larger data set.	None.





Sl. No.	Name of the author with year	Major technologies used	Results/Outcome of research	Drawbacks if any
19.	Yadigar Imamverdiyev and Fargana Abdullayeva Deep learning method for denial of service attack detection based on restricted boltzmann machine, 2018	Gaussian–Bernoulli type restricted Boltzmann machine (RBM) for detection of denial of service (DoS) attacks	Gaussian–Bernoulli RBM method gave better results in detecting attacks than the other Bernoulli-Bernoulli RBM and DBN type deep learning methods.	-
20.	Frederico A. F. Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar and Luiz F. Silveira Smart detection: an online approach for DoS/DDoS attack detection using machine learning, 2019	The proposed approach(Smart Detection) makes inferences based on signatures previously extracted from samples of network traffic.	Based on the experimental results, the Smart Detection approach delivers improved DR, FAR, and PREC.	The method was not so affective in multiple-class classification, self-configuration of the system.
21.	Waheed G. Gadallah, Nagwa M. Omar and Hosny M. Ibrahim Machine Learning-based Distributed Denial of Service Attacks Detection Technique using New Features in Software-defined Networks, 2021	The technique uses advancing new traffic-based flow features to create the model, which can classify SDN flow packets as normal or a DoS attack.	The effectiveness of the introduced detection method can be justified due to the high obtained detection accuracy and low false-positive rate.	It may suffer from creating a single point of failure against the controller, which represents the network control plane.
22.	Ahmed Iqbal, Shabib Aftab, Israr Ullah, Muhammad Anwaar Saeed, Arif Husen A Classification Framework to Detect DoS Attacks, 2019	This technique uses framework for detection.The framework consists of five stages, including: 1) selection of the relevant Dataset, 2) Data pre-processing, 3) Feature Selection, 4) Detection, and 5) reflection of Results.	The proposed framework outperformed all other classifiers in each accuracy measure including Naive Bayes, Support vector machines, KNN,RBF etc -	-
23.	Aamir, M and S.M.A Zaidi. DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation, 2019	This paper applies an organized flow of feature engineering and machine learning to detect distributed denial-of-service (DDoS) attacks. Different supervised machine learning models are applied on the feature-engineered datasets to demonstrate the adaptability	The results show that substantial feature reduction is possible to make DDoS detection faster and optimized with minimal performance hit. There is approximately 68% reduction in the feature space is possible with an impact of only	None.

		of datasets for machine learning under optimal tuning of parameters within given sets of values.	about 0.03% on accuracy.	
24.	S.Shanmuga Priya, M.Sivaram, D.Yuvaraj A.Jayanthiladevi Machine Learning based DDOS Detection, 2020	Automated DDoS detector using ML which can run on any commodity hardware. Three classification algorithms KNN, RF and NB were used to classify DDoS packets from normal packets by two features, delta time and packet size.	The results are 98.5 % accurate	Detection model has been trained with some of the popular DDoS tools such as hping3, it may not detect Ddos attacks which are created by other DDoS tools.
25.	Gloria C.Y. Tsang, Patrick P.K. Chan, Daneji. S. Yeung, Eric C.C. Tsang Denial Of Service Detection By Support Vector Machines And Radial-Basis Function Neural Nework	RBFNN was proposed by Gloria in contrast to SVM, the former combining nonlinear basis functions to serve as a universal nonlinear approximator. The construction of RBF" only involves three layers in its basic forms. The input layer is made up of source nodes that connect the network to its environment. The hidden layer applies a nonlinear transformation from the input space to the hidden space which is of high dimensionality. The output layer performs linear combination of the hidden output.	SVM has slightly higher accuracy than RBFNN	-
26	Frederico A. F. Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar and Luiz F. Silveira Smart detection: an online approach for DoS/DDoS attack detection using machine learning, 2019	The proposed approach(Smart Detection) makes inferences based on signatures previously extracted from samples of network traffic.	Based on the experimental results, the Smart Detection approach delivers improved DR, FAR, and PREC.	The method was not so affective in multiple-class classification, self-configuration of the system.



27	Jiangtao Pei, Yunli Chen, Wei Ji A DDoS Attack Detection Method Based on Machine Learning, June 2019	The characteristics of the attack traffic obtained in the model detection phase are trained in the training model based on the random forest algorithm. Finally, the test model is validated by the DDoS attack, and the SVM method in the machine learning is compared in terms of detection accuracy.	The experimental results show that the proposed DDoS attack detection method based on machine learning has a good detection rate for the current popular DDoS attacks.	-
28	Naiji Zhang, Fehmi Jaafar, Yasir Malik. Low-Rate DoS Attack Detection Using PSD based Entropy and Machine Learning, 2019	In this paper, the authors presented an algorithm by combing PSD-entropy and (SVM), to improve the efficiency and robustness of LDoS detection system.	The experimental results show that the proposed approach can detect 99.19% of the LDoS attacks and has O(n log n) time complexity in the best case. The proposed method provides a way to detect LDoS attacks more efficiently and provide meaningful direction for future LDoS detection research.	To have a higher detection rate, the thresholds are set to have 80% priority on accuracy.
29	Mohammad Shurman, Rami Khrais, and Abdulrahman Yateem DoS and DDoS Attack Detection Using Deep Learning and IDS, February 2020	Deep Learning,IDS, IoT,Signature-based and anomaly based IDPS, LSTM model	A hybrid model proved more efficiency of > 99% which is higher than the existing model	Untested model in realtime.
30	X. Yuan, C. Li and X. Li DeepDefense: Identifying DDoS Attack via Deep Learning, 2017	This paper proposes a deep learning-based DDoS attack detection approach (DeepDefense). A recurrent deep neural network is designed in order to learn patterns from sequences of network traffic and trace network attack activities	The experimental results demonstrate a better performance of the model compared with conventional machine learning models. The error rate is reduced from 7.517% to 2.103% compared with conventional machine learning method in the larger data set.	-

PROPOSED MODEL:

Architecture

Smart Detection combats DDoS attacks by collecting and analysing network traffic samples, followed by subsequent classification into normal or malicious traffic based on various parameters. As soon as an attack is detected, notification messages are sent using cloud platform.

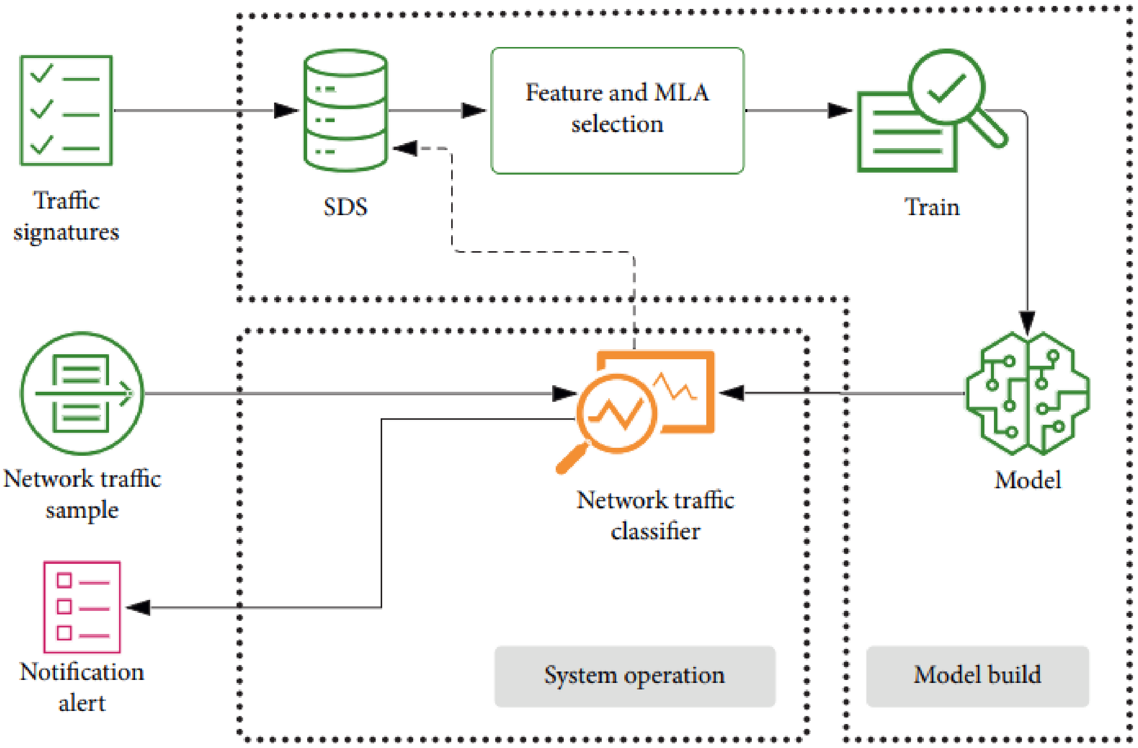


Figure 1. Overview of model

Classification is based on characteristics such as

1. average packet size
2. bit rate,
3. packet size
4. inter arrival time etc.

These characteristics are used to classify network traffic as normal or DDoS attack. The average packet size in most DDoS attempts is the same. As a result of examining datasets, machine learning techniques determine if traffic in the network is attack traffic or normal traffic. In order to detect threats in network traffic

Work Environment: Jupyter Notebook and Google Colab as our working environment**.**

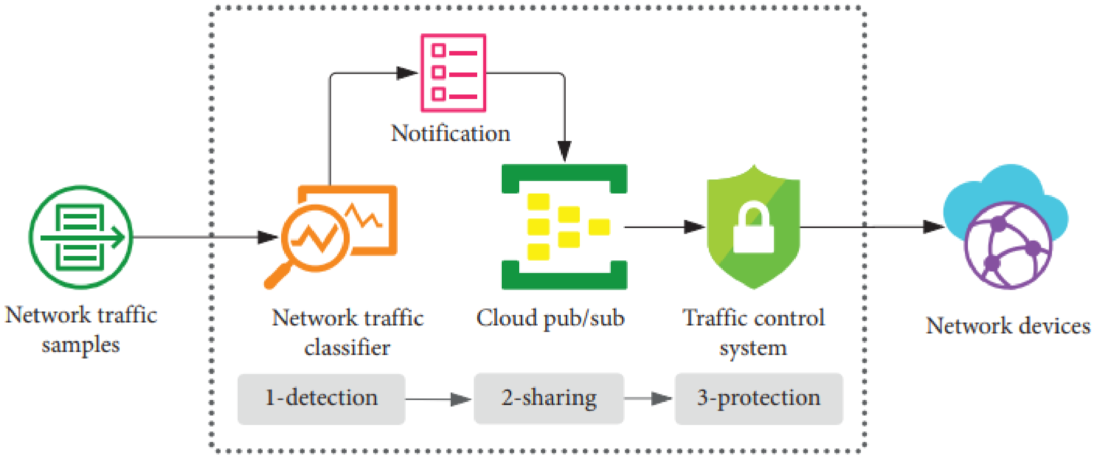
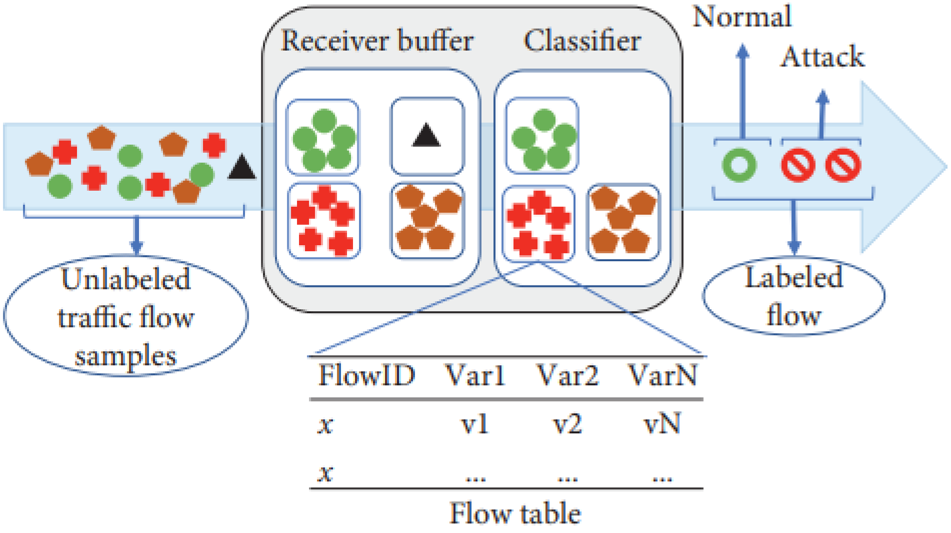


Figure 2. High-level architecture



FlowID	Var1	Var2	VarN
x	v1	v2	vN
x	...	...	...

Figure 3: Traffic Classification

WORKFLOW:

A Signature Dataset (SDS) (dataset in use) and a machine learning algorithm are at the heart of the detection system.

- First, signatures from normal traffic and DoS attacks were retrieved, classified, and saved in a database. Using feature selection approaches, SDS was then built. Finally, the MLA with the highest accuracy was chosen, trained, and placed into the traffic classification system.

-The detection-system's architecture was created to deal with network traffic samples acquired from network devices using traffic sampling techniques of industrial standard.

- Here, the receiver buffer, recieves unlabeled samples and groups them into flow tables. As and when, the table length exceeds or equals the reference value, subsequently they are handed over to the classifier for labelling. If the flow table is about to expire, it can be processed again.

-Small flow tables are more common at lower sampling rates or during certain forms of DoS attacks, such as SYN flood attacks.

-Traffic samples are acquired and stored in flow tables during each cycle of detection. In stages 1 and 2, a unique identificationnumber (FlowID) is calculated separately for each new flow based on 5 factors (src port, dst port, src IP, dst IP, and transport protocol).

-If the flow is new, (i.e no other flow table has the same FlowID), it is then registered in a shared memory buffer.

-If a previously registered flow table has the same FlowID as the new one , the existing flow's data will be merged with the new flow in stages 3 and 4.

-If the table length exceeds or equals the reference value, clssification of the flow table takes place, and if an attack is detected, a notification is sent. Or else, it is put back into the memory buffer(shared)

-In step 7, the cleanup task searches the shared buffer for expired flow tables, i.e flow tables that have surpassed the system's expiration time

-The table length is checked for each of the expired flow tables by the system. If the length of the flow table is below then or equals the minimum reference value it is processed in stage 8.

-A new FlowID is computed using 3-tuples (transport_protocol, src_IP and dst_IP), as the flow table is directed back to stages 3 and 4

-In the presented flowchart, the detecting system's entire algorithm is summarised.

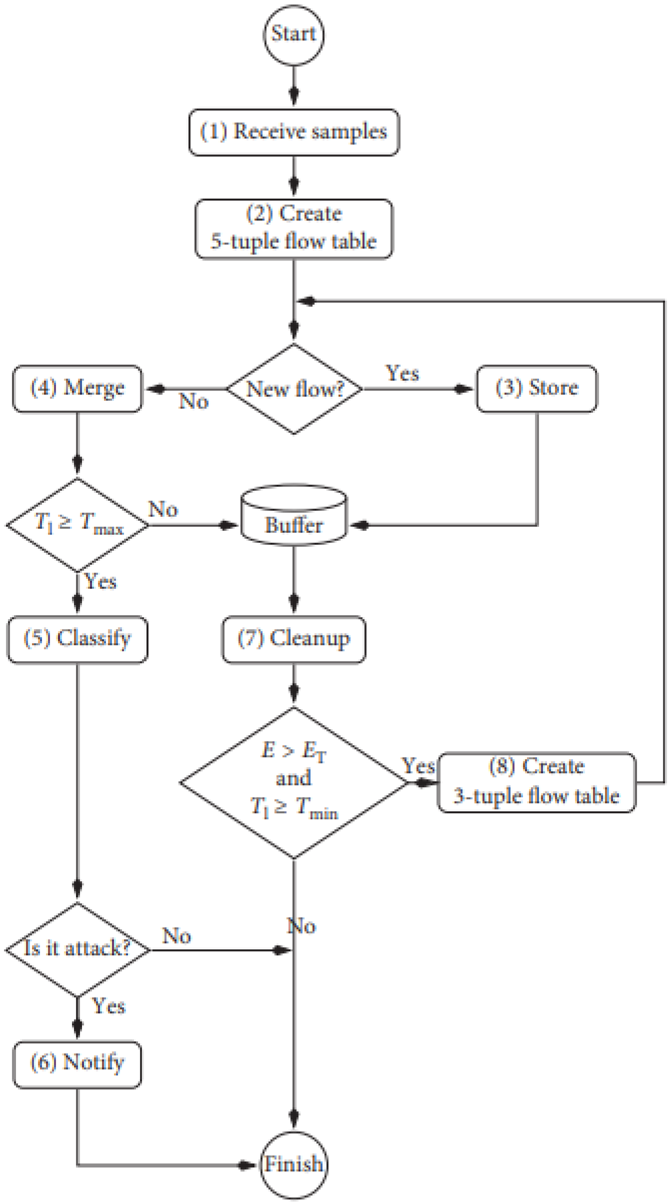


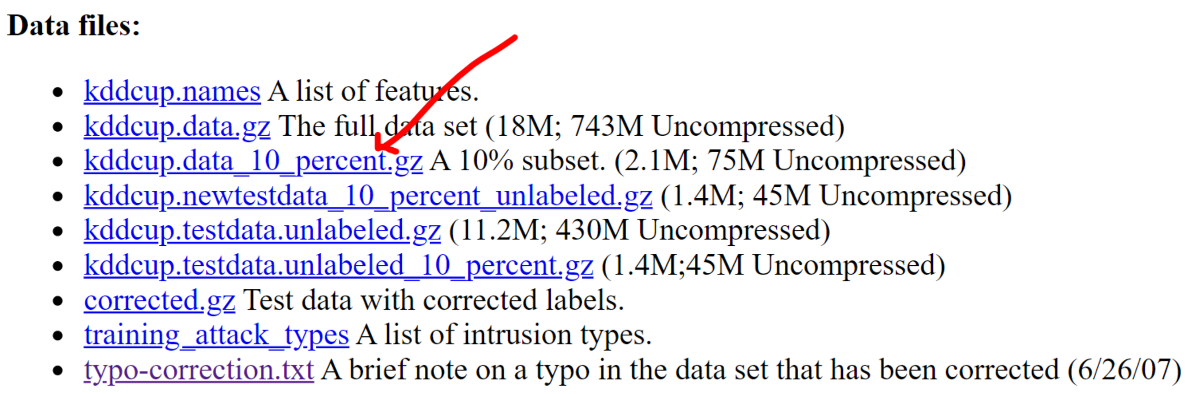
Figure 4. Flowchart of workflow

\

DATASET USED:

KDD Cup 1999 Data (uci.edu)

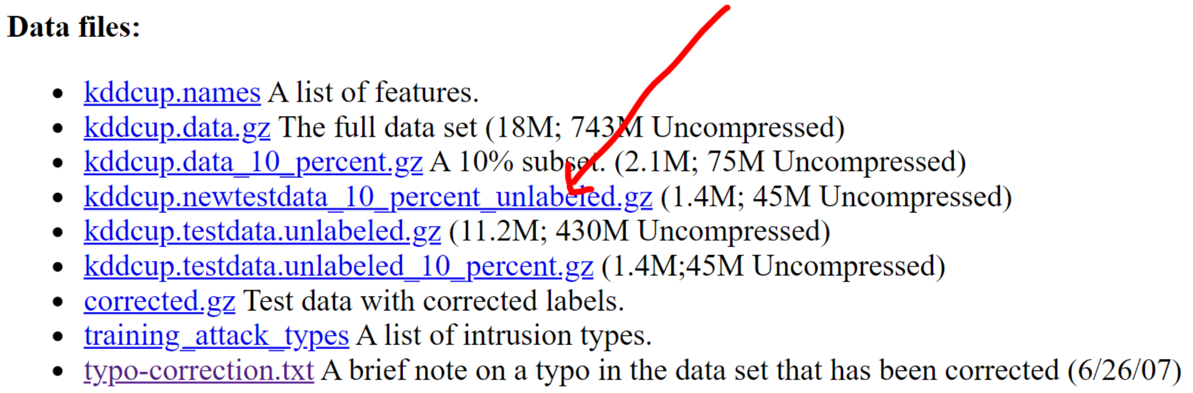
For Training we have used the kddcup.data_10_percent.gz which is a 10% of the original dataset.



Data files:

- kddcup.names A list of features.
- kddcup.data.gz The full data set (18M; 743M Uncompressed)
- kddcup.data_10_percent.gz A 10% subset. (2.1M; 75M Uncompressed)
- kddcup.newtestdata_10_percent_unlabeled.gz (1.4M; 45M Uncompressed)
- kddcup.testdata.unlabeled.gz (11.2M; 430M Uncompressed)
- kddcup.testdata.unlabeled_10_percent.gz (1.4M;45M Uncompressed)
- corrected.gz Test data with corrected labels.
- training_attack_types A list of intrusion types.
- typo-correction.txt A brief note on a typo in the data set that has been corrected (6/26/07)

For prediction of attack we have used kddcup.newtestdata_10_percent_unlabeled.gz

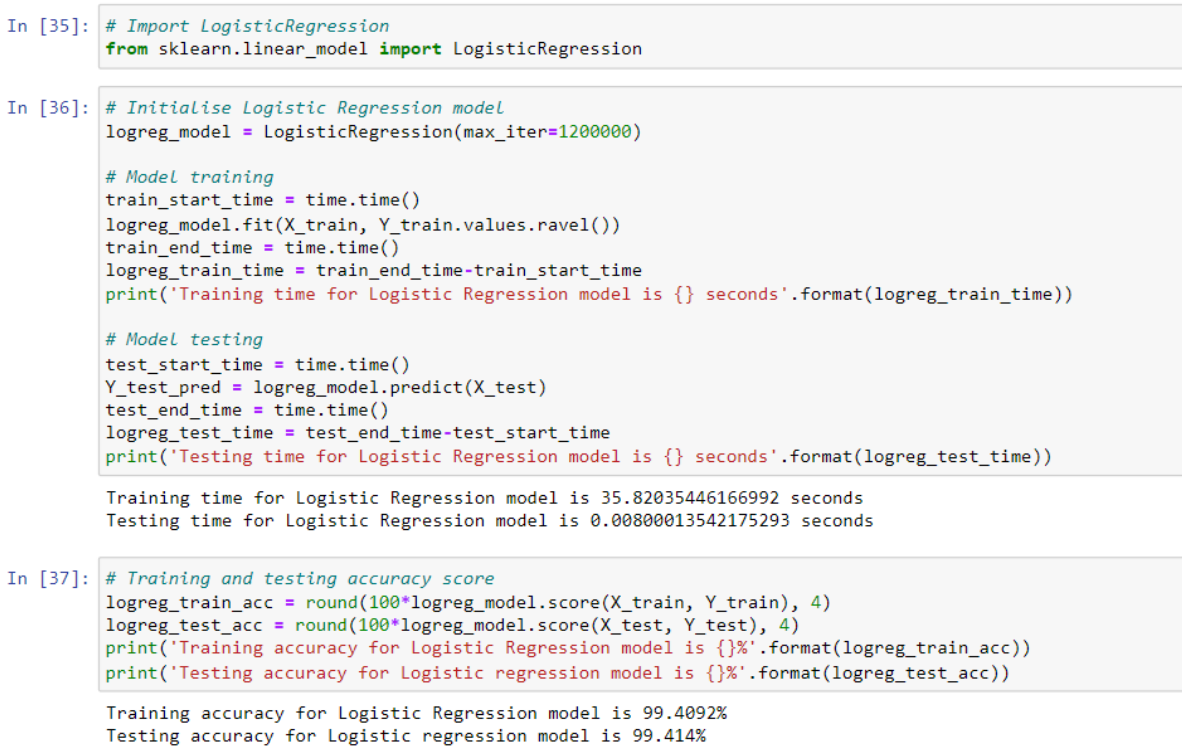


Data files:

- kddcup.names A list of features.
- kddcup.data.gz The full data set (18M; 743M Uncompressed)
- kddcup.data_10_percent.gz A 10% subset. (2.1M; 75M Uncompressed)
- kddcup.newtestdata_10_percent_unlabeled.gz (1.4M; 45M Uncompressed)
- kddcup.testdata.unlabeled.gz (11.2M; 430M Uncompressed)
- kddcup.testdata.unlabeled_10_percent.gz (1.4M;45M Uncompressed)
- corrected.gz Test data with corrected labels.
- training_attack_types A list of intrusion types.
- typo-correction.txt A brief note on a typo in the data set that has been corrected (6/26/07)

MODELS USED:

1. **Logistic regression:** It is a classification algorithm based on supervised learning, often used to predict probabilities of target variables. Because the dichotomous nature of the dependent there are only two classifications. In simple terms, the target variable has a binary nature, with data coded either 1 (representing success) or 0 (representing failure).



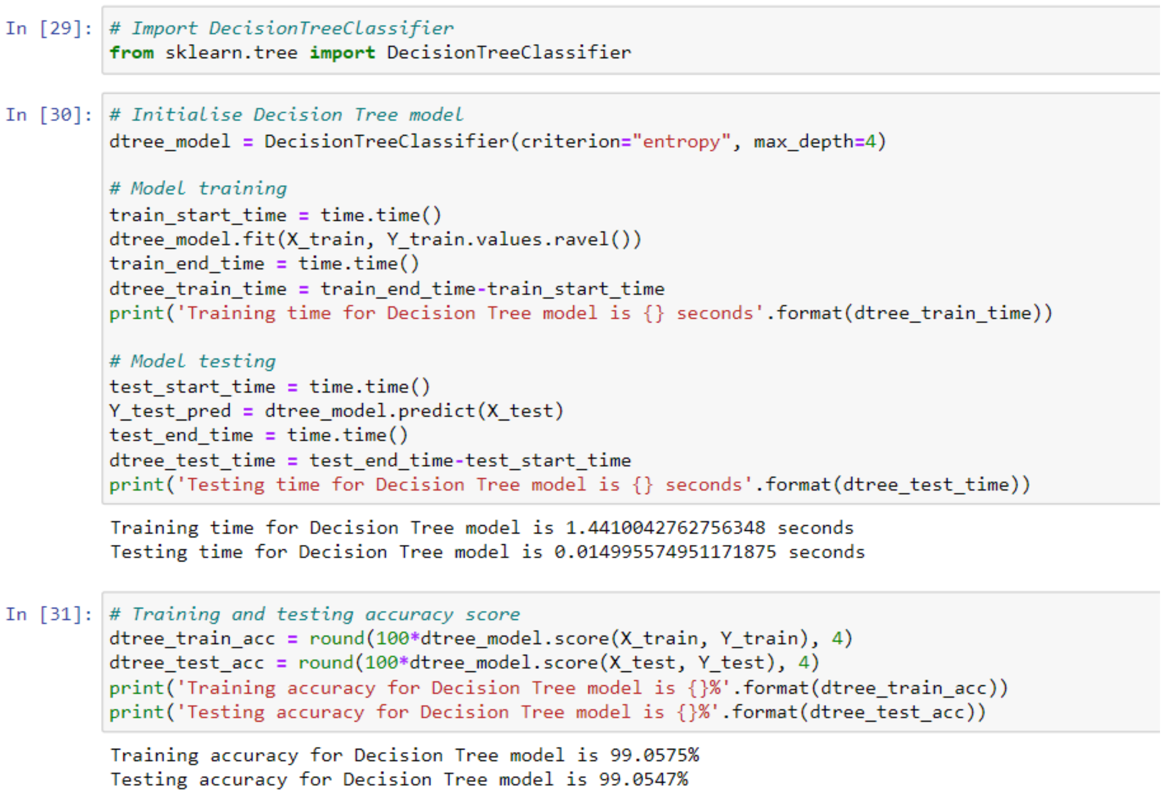
```
In [35]: # Import LogisticRegression
         from sklearn.linear_model import LogisticRegression
```

```
In [36]: # Initialise Logistic Regression model
         logreg_model = LogisticRegression(max_iter=1200000)

         # Model training
         train_start_time = time.time()
         logreg_model.fit(X_train, Y_train.values.ravel())
         train_end_time = time.time()
         logreg_train_time = train_end_time-train_start_time
         print('Training time for Logistic Regression model is {} seconds'.format(logreg_train_time))

         # Model testing
         test_start_time = time.time()
         Y_test_pred = logreg_model.predict(X_test)
         test_end_time = time.time()
         logreg_test_time = test_end_time-test_start_time
         print('Testing time for Logistic Regression model is {} seconds'.format(logreg_test_time))

         Training time for Logistic Regression model is 35.82035446166992 seconds
         Testing time for Logistic Regression model is 0.00800013542175293 seconds
```

```
In [37]: # Training and testing accuracy score
         logreg_train_acc = round(100*logreg_model.score(X_train, Y_train), 4)
         logreg_test_acc = round(100*logreg_model.score(X_test, Y_test), 4)
         print('Training accuracy for Logistic Regression model is {}%'.format(logreg_train_acc))
         print('Testing accuracy for Logistic regression model is {}%'.format(logreg_test_acc))

         Training accuracy for Logistic Regression model is 99.4092%
         Testing accuracy for Logistic regression model is 99.414%
```

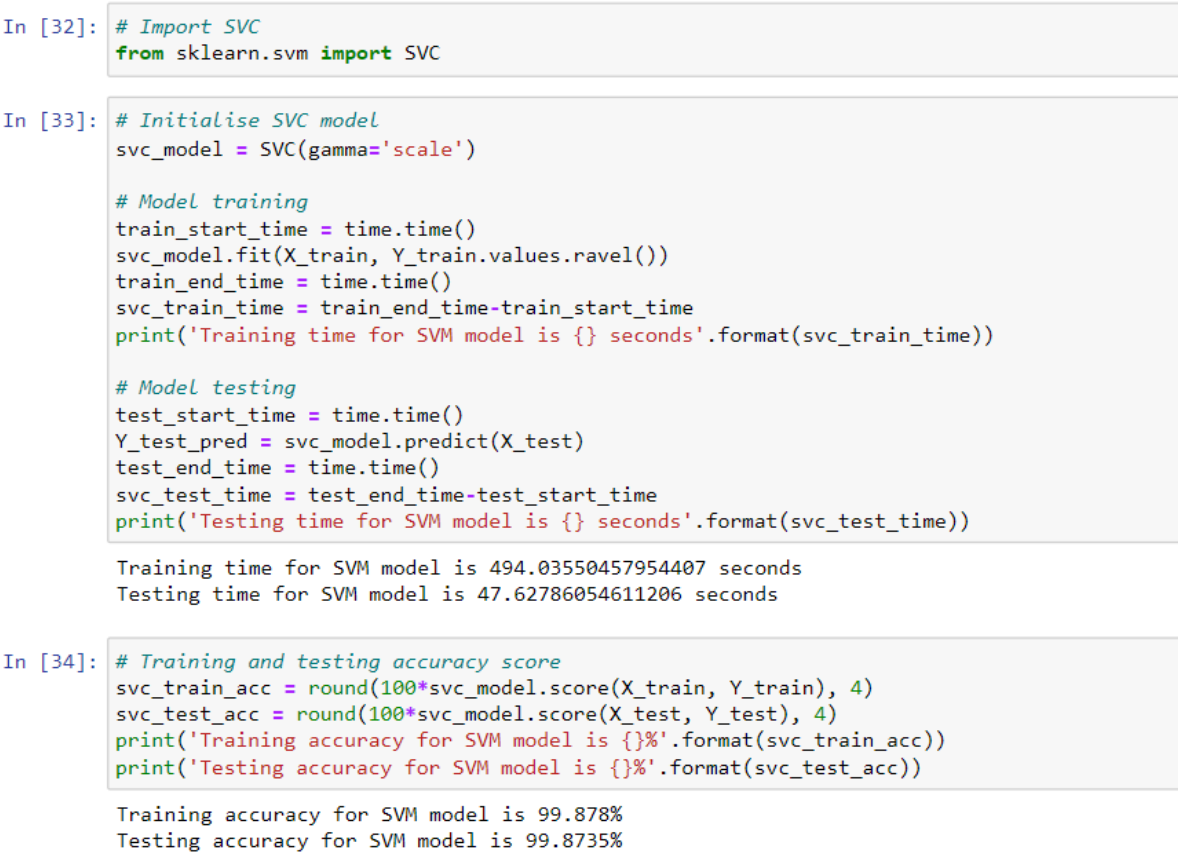
ii. Decision Tree Classifier:

The Decision Tree algorithm is part of the supervised learning algorithm family. The decision tree approach, unlike the other supervised learning algorithms, could also be utilised to solve regression and classification problems. The goal of utilizing a Decision Tree is to build a training model that can be used to predict the class or the value of a target variables by learning decision rules deduced from previous data (training data).



```
In [29]: # Import DecisionTreeClassifier
         from sklearn.tree import DecisionTreeClassifier

In [30]: # Initialise Decision Tree model
         dtree_model = DecisionTreeClassifier(criterion="entropy", max_depth=4)

         # Model training
         train_start_time = time.time()
         dtree_model.fit(X_train, Y_train.values.ravel())
         train_end_time = time.time()
         dtree_train_time = train_end_time-train_start_time
         print('Training time for Decision Tree model is {} seconds'.format(dtree_train_time))

         # Model testing
         test_start_time = time.time()
         Y_test_pred = dtree_model.predict(X_test)
         test_end_time = time.time()
         dtree_test_time = test_end_time-test_start_time
         print('Testing time for Decision Tree model is {} seconds'.format(dtree_test_time))

         Training time for Decision Tree model is 1.4410042762756348 seconds
         Testing time for Decision Tree model is 0.014995574951171875 seconds

In [31]: # Training and testing accuracy score
         dtree_train_acc = round(100*dtree_model.score(X_train, Y_train), 4)
         dtree_test_acc = round(100*dtree_model.score(X_test, Y_test), 4)
         print('Training accuracy for Decision Tree model is {}%'.format(dtree_train_acc))
         print('Testing accuracy for Decision Tree model is {}%'.format(dtree_test_acc))

         Training accuracy for Decision Tree model is 99.0575%
         Testing accuracy for Decision Tree model is 99.0547%
```

iii. Support Vector Machines: It is a popular Supervised Machine Learning algorithm, used to solve both classification and regression tasks. The SVM algorithm's objective is to create the most optimum line or decision boundary which can categorize the n-dimensional space into classes, by which we can easily put new data points in correct categories in future. Hyperplane is the name for the best choice boundary. The extreme points that help to create the hyperplane are chosen via SVM. Support vectors are the extreme instances, and the algorithm is called a Support Vector Machine.

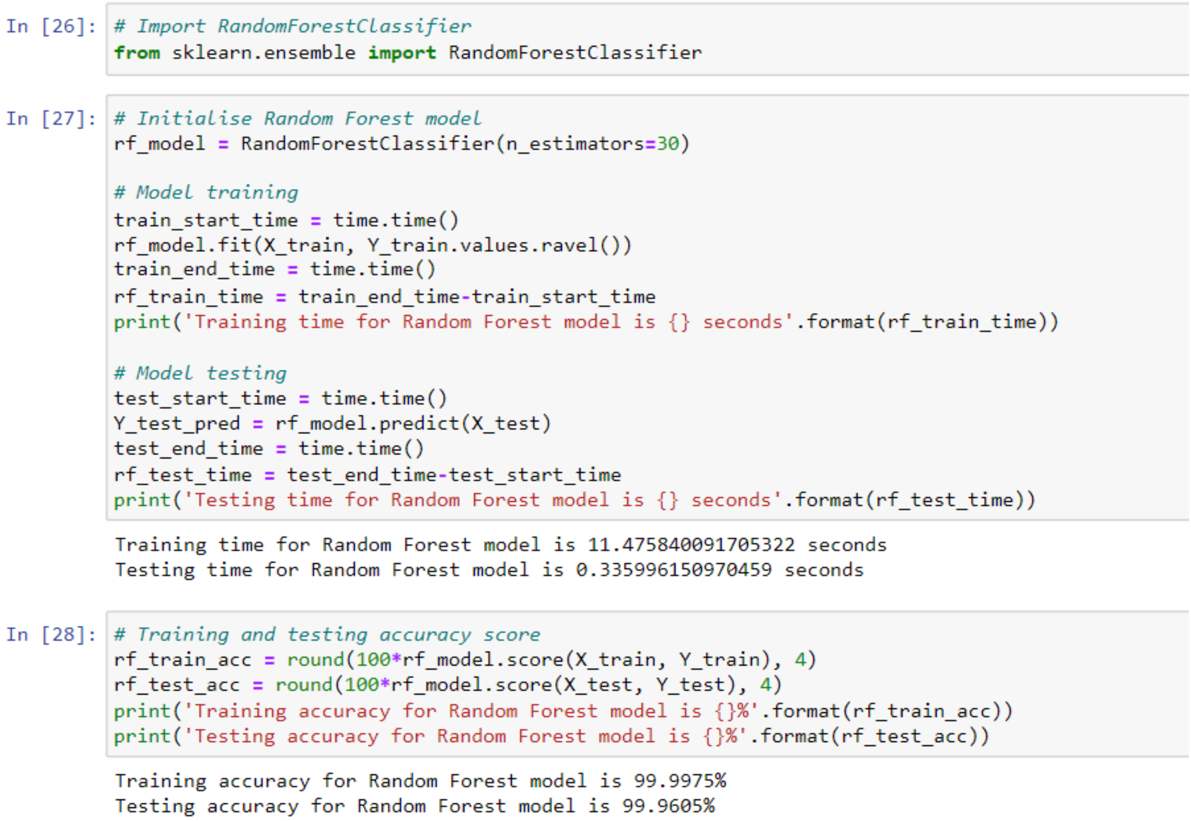


```
In [32]: # Import SVC
         from sklearn.svm import SVC
```

```
In [33]: # Initialise SVC model
         svc_model = SVC(gamma='scale')

         # Model training
         train_start_time = time.time()
         svc_model.fit(X_train, Y_train.values.ravel())
         train_end_time = time.time()
         svc_train_time = train_end_time-train_start_time
         print('Training time for SVM model is {} seconds'.format(svc_train_time))

         # Model testing
         test_start_time = time.time()
         Y_test_pred = svc_model.predict(X_test)
         test_end_time = time.time()
         svc_test_time = test_end_time-test_start_time
         print('Testing time for SVM model is {} seconds'.format(svc_test_time))

         Training time for SVM model is 494.03550457954407 seconds
         Testing time for SVM model is 47.62786054611206 seconds
```

```
In [34]: # Training and testing accuracy score
         svc_train_acc = round(100*svc_model.score(X_train, Y_train), 4)
         svc_test_acc = round(100*svc_model.score(X_test, Y_test), 4)
         print('Training accuracy for SVM model is {}%'.format(svc_train_acc))
         print('Testing accuracy for SVM model is {}%'.format(svc_test_acc))

         Training accuracy for SVM model is 99.878%
         Testing accuracy for SVM model is 99.8735%
```

iv. Random Forest Classifier: Random Forest is a supervised learning model based on ensemble learning, which is a method of integrating several classifiers to solve complex problems and increase the model's performance. It contains decision trees on different subsets of the dataset and takes an average to enhance the predicted accuracy of the dataset in use. Rather than relying on a single decision tree, random forest collects the forecasts from each tree and predicts the final output based on the majority votes of predictions.

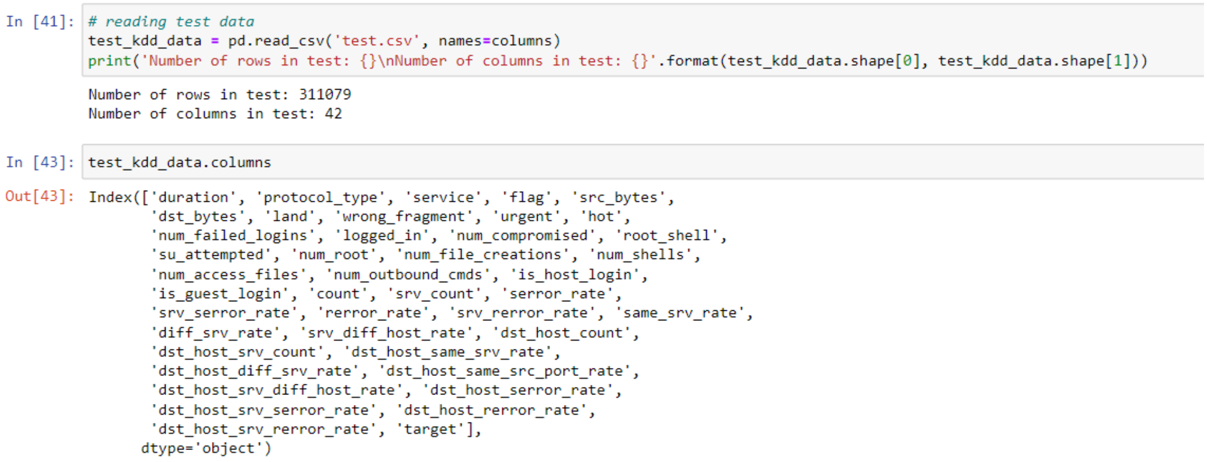


```
In [26]: # Import RandomForestClassifier
         from sklearn.ensemble import RandomForestClassifier
```

```
In [27]: # Initialise Random Forest model
         rf_model = RandomForestClassifier(n_estimators=30)

         # Model training
         train_start_time = time.time()
         rf_model.fit(X_train, Y_train.values.ravel())
         train_end_time = time.time()
         rf_train_time = train_end_time-train_start_time
         print('Training time for Random Forest model is {} seconds'.format(rf_train_time))

         # Model testing
         test_start_time = time.time()
         Y_test_pred = rf_model.predict(X_test)
         test_end_time = time.time()
         rf_test_time = test_end_time-test_start_time
         print('Testing time for Random Forest model is {} seconds'.format(rf_test_time))

         Training time for Random Forest model is 11.475840091705322 seconds
         Testing time for Random Forest model is 0.335996150970459 seconds
```

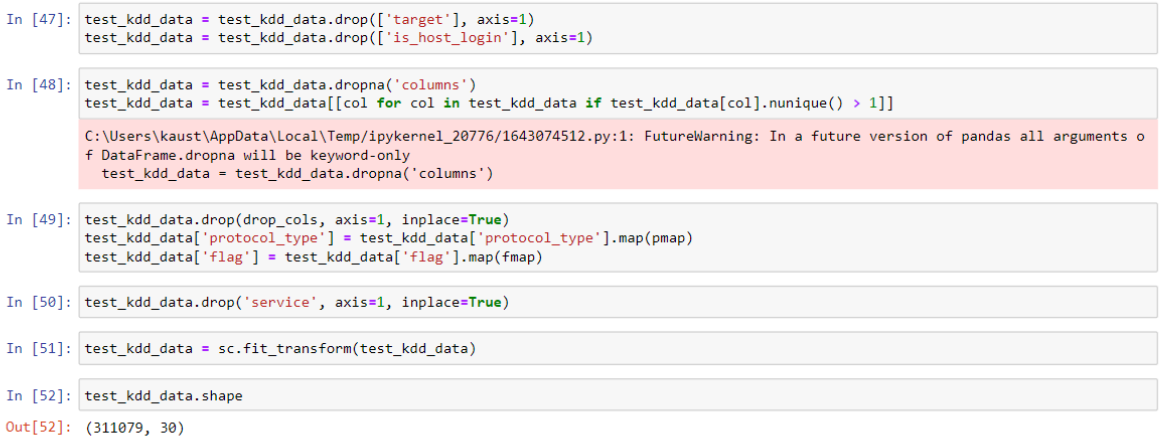
```
In [28]: # Training and testing accuracy score
         rf_train_acc = round(100*rf_model.score(X_train, Y_train), 4)
         rf_test_acc = round(100*rf_model.score(X_test, Y_test), 4)
         print('Training accuracy for Random Forest model is {}%'.format(rf_train_acc))
         print('Testing accuracy for Random Forest model is {}%'.format(rf_test_acc))

         Training accuracy for Random Forest model is 99.9975%
         Testing accuracy for Random Forest model is 99.9605%
```

USING ANOTHER DATASET:



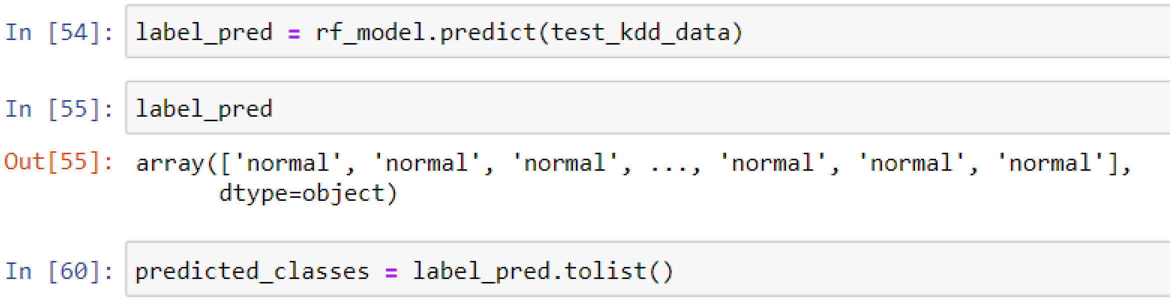
```
In [41]: # reading test data
         test_kdd_data = pd.read_csv('test.csv', names=columns)
         print('Number of rows in test: {}\nNumber of columns in test: {}'.format(test_kdd_data.shape[0], test_kdd_data.shape[1]))

         Number of rows in test: 311079
         Number of columns in test: 42

In [43]: test_kdd_data.columns

Out[43]: Index(['duration', 'protocol_type', 'service', 'flag', 'src_bytes',
                'dst_bytes', 'land', 'wrong_fragment', 'urgent', 'hot',
                'num_failed_logins', 'logged_in', 'num_compromised', 'root_shell',
                'su_attempted', 'num_root', 'num_file_creations', 'num_shells',
                'num_access_files', 'num_outbound_cmds', 'is_host_login',
                'is_guest_login', 'count', 'srv_count', 'serror_rate',
                'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate', 'same_srv_rate',
                'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count',
                'dst_host_srv_count', 'dst_host_same_srv_rate',
                'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate',
                'dst_host_srv_diff_host_rate', 'dst_host_serror_rate',
                'dst_host_srv_serror_rate', 'dst_host_rerror_rate',
                'dst_host_srv_rerror_rate', 'target'],
               dtype='object')
```

Preprocessing:



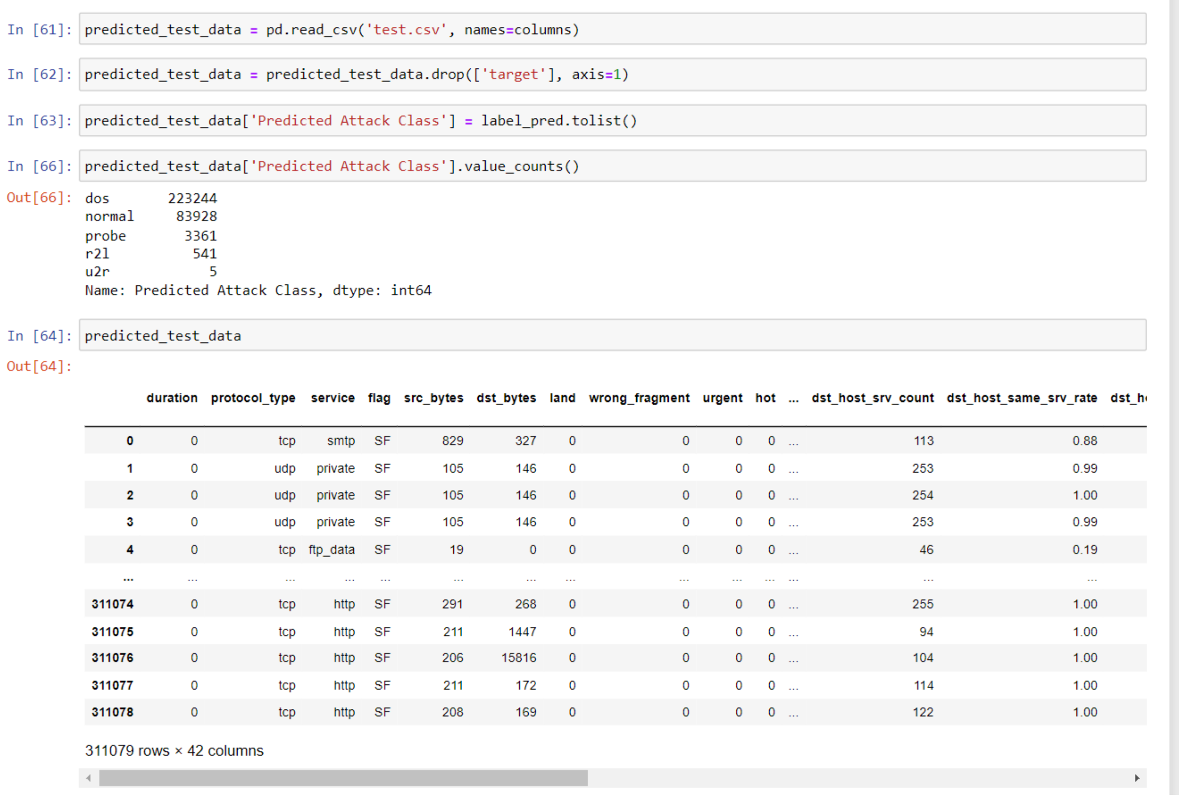
```
In [47]: test_kdd_data = test_kdd_data.drop(['target'], axis=1)
         test_kdd_data = test_kdd_data.drop(['is_host_login'], axis=1)

In [48]: test_kdd_data = test_kdd_data.dropna('columns')
         test_kdd_data = test_kdd_data[[col for col in test_kdd_data if test_kdd_data[col].nunique() > 1]]

         C:\Users\kaust\AppData\Local\Temp/ipykernel_20776/1643074512.py:1: FutureWarning: In a future version of pandas all arguments o
         f DataFrame.dropna will be keyword-only
           test_kdd_data = test_kdd_data.dropna('columns')

In [49]: test_kdd_data.drop(drop_cols, axis=1, inplace=True)
         test_kdd_data['protocol_type'] = test_kdd_data['protocol_type'].map(pmap)
         test_kdd_data['flag'] = test_kdd_data['flag'].map(fmap)

In [50]: test_kdd_data.drop('service', axis=1, inplace=True)

In [51]: test_kdd_data = sc.fit_transform(test_kdd_data)

In [52]: test_kdd_data.shape

Out[52]: (311079, 30)
```

Using random forest model for prediction:



```
In [54]: label_pred = rf_model.predict(test_kdd_data)

In [55]: label_pred

Out[55]: array(['normal', 'normal', 'normal', ..., 'normal', 'normal', 'normal'],
               dtype=object)

In [60]: predicted_classes = label_pred.tolist()
```

Appending our predictions in our test dataset:



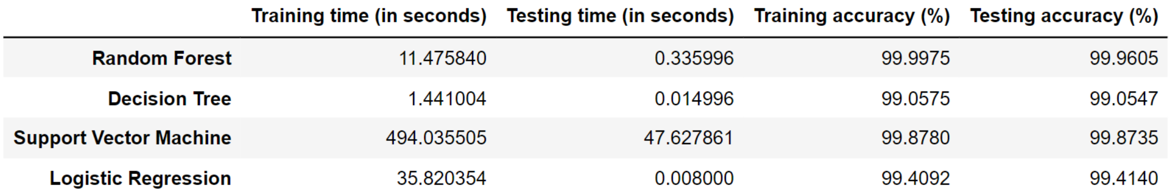
```
In [61]: predicted_test_data = pd.read_csv('test.csv', names=columns)

In [62]: predicted_test_data = predicted_test_data.drop(['target'], axis=1)

In [63]: predicted_test_data['Predicted Attack Class'] = label_pred.tolist()

In [66]: predicted_test_data['Predicted Attack Class'].value_counts()

Out[66]: dos       223244
         normal     83928
         probe       3361
         r2l          541
         u2r            5
         Name: Predicted Attack Class, dtype: int64

In [64]: predicted_test_data

Out[64]:
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_h
0	0	tcp	smtp	SF	829	327	0	0	0	0	...	113	0.88	
1	0	udp	private	SF	105	146	0	0	0	0	...	253	0.99	
2	0	udp	private	SF	105	146	0	0	0	0	...	254	1.00	
3	0	udp	private	SF	105	146	0	0	0	0	...	253	0.99	
4	0	tcp	ftp_data	SF	19	0	0	0	0	0	...	46	0.19	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
311074	0	tcp	http	SF	291	268	0	0	0	0	...	255	1.00	
311075	0	tcp	http	SF	211	1447	0	0	0	0	...	94	1.00	
311076	0	tcp	http	SF	206	15816	0	0	0	0	...	104	1.00	
311077	0	tcp	http	SF	211	172	0	0	0	0	...	114	1.00	
311078	0	tcp	http	SF	208	169	0	0	0	0	...	122	1.00	

311079 rows × 42 columns

ACCURACY COMPARISON:

Below is the table created from pandas data frame showing different accuracies of training testing and also the time required for training and testing.



	Training time (in seconds)	Testing time (in seconds)	Training accuracy (%)	Testing accuracy (%)
Random Forest	11.475840	0.335996	99.9975	99.9605
Decision Tree	1.441004	0.014996	99.0575	99.0547
Support Vector Machine	494.035505	47.627861	99.8780	99.8735
Logistic Regression	35.820354	0.008000	99.4092	99.4140

CONCLUSION:

Finally, we see that the Random Forest model achieves the highest accuracy on both training and tesing datasets. Logistic Regression is slow and also has less accuracy than others (SVM and Random Forest) SVC model is very slow but it has a very high accuracy, as it is very slow so can't be used in real life detection system.

However, the Decision Tree model is the quickest in terms of training and testing time. It also gives fairly high accuracy. Hence, in terms of time and accuracy combined the best model out of the 4 is the Decision Tree model as it achieves high accuracy while also minimizing training and testing time.

While if accuracy is prime focus then the Random Forest Model is to be preferred. Such a model can be used in intrusion detection systems to make these systems more robust and intelligent in detecting malicious attacks such as Denial of Service attacks.

REFERENCES:

1. Ramesh, S., Yaashuwanth, C., Prathibanandhi, K., Basha, A. R., & Jayasankar, T. (2021). An optimized deep neural network based DoS attack detection in wireless video sensor network. *Journal of Ambient Intelligence and Humanized Computing*, 1-14.
2. Yadav, S., & Selvakumar, S. (2015, September). Detection of application layer DDoS attack by modeling user behavior using logistic regression. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)* (pp. 1-6). IEEE.
3. Shurman, M. M., Khrais, R. M., & Yateem, A. A. (2019, December). IoT denial-of-service attack detection and prevention using hybrid IDS. In *2019 International Arab Conference on Information Technology (ACIT)* (pp. 252-254). IEEE.
4. Perez-Diaz, J. A., Valdovinos, I. A., Choo, K. K. R., & Zhu, D. (2020). A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. *IEEE Access*, *8*, 155859-155872.
5. Amjad, A., Alyas, T., Farooq, U., & Tariq, M. A. (2019). Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm. *EAI Endorsed Transactions on Scalable Information Systems*, *6*(26).
6. Muhammad, A. W., Foozy, C. F. M., & Azhari, A. (2020). Machine Learning-Based Distributed Denial of Service Attack Detection on Intrusion Detection System Regarding to Feature Selection. *International Journal of Artificial Intelligence Research*, *4*(1), 1-8.
7. Gopal, S. B., Poongodi, C., Nanthiya, D., Priya, R. S., Saran, G., & Priya, M. S. (2021, February). Mitigating DoS attacks in IoT using Supervised and Unsupervised

Algorithms–A Survey. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1055, No. 1, p. 012072). IOP Publishing.

8. Sambangi, S., & Gondi, L. (2020). A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression. *Multidisciplinary Digital Publishing Institute Proceedings*, *63*(1), 51.
9. Suresh, M., & Anitha, R. (2011, July). Evaluating machine learning algorithms for detecting DDoS attacks. In *International Conference on Network Security and Applications* (pp. 441-452). Springer, Berlin, Heidelberg.
10. Tuan, T. A., Long, H. V., Son, L. H., Kumar, R., Priyadarshini, I., & Son, N. T. K. (2020). Performance evaluation of Botnet DDoS attack detection using machine learning. *Evolutionary Intelligence*, *13*(2), 283-294.
11. Pei, J., Chen, Y., & Ji, W. (2019, June). A ddos attack detection method based on machine learning. In *Journal of Physics: Conference Series* (Vol. 1237, No. 3, p. 032040). IOP Publishing.
12. Zhang, N., Jaafar, F., & Malik, Y. (2019, June). Low-rate DoS attack detection using PSD based entropy and machine learning. In *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 59-62). IEEE.
13. Sinha, M., Gupta, S., Rout, S. S., & Deb, S. (2021). Sniffer: A machine learning approach for DoS attack localization in NoC-based SoCs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *11*(2), 278-291.
14. Mamolar, A. S., Pervez, Z., Calero, J. M. A., & Khattak, A. M. (2018). Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks. *Computers & Security*, *79*, 132-147.
15. Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials*, *15*(4), 2046-2069.
16. Paudel, R., Muncy, T., & Eberle, W. (2019, December). Detecting dos attack in smart home iot devices using a graph-based approach. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 5249-5258). IEEE.
17. Voting extreme learning machine based distributed denial of service attack detection in cloud computing, 2020
18. Yuan, X., Li, C., & Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8). IEEE.
19. Imamverdiyev, Y., & Abdullayeva, F. (2018). Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big data*, *6*(2), 159-169.
20. Lima Filho, F. S. D., Silveira, F. A., de Medeiros Brito Junior, A., Vargas-Solar, G., & Silveira, L. F. (2019). Smart detection: an online approach for DoS/DDoS attack detection using machine learning. *Security and Communication Networks*, *2019*.
21. Gadallah, W. G., Omar, N. M., & Ibrahim, H. M. (2021). Machine Learning-based Distributed Denial of Service Attacks Detection Technique using New Features in Software-defined Networks. *International Journal of Computer Network and Information Security (IJCNIS)*, *13*(3), 15-27.

22. Iqbal, A., Aftab, S., Ullah, I., Saeed, M. A., & Husen, A. (2019). A Classification Framework to Detect DoS Attacks. *International Journal of Computer Network & Information Security*, *11*(9).
23. Aamir, M., & Zaidi, S. M. A. (2019). DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation. *International Journal of Information Security*, *18*(6), 761-785.
24. Priya, S. S., Sivaram, M., Yuvaraj, D., & Jayanthiladevi, A. (2020, March). Machine learning based DDoS detection. In *2020 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 234-237). IEEE.
25. Tsang, G. C., Chan, P. P., Yeung, D. S., & Tsang, E. C. (2004, August). Denial of service detection by support vector machines and radial-basis function neural network. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)* (Vol. 7, pp. 4263-4268). IEEE.
26. Suresh, M., & Anitha, R. (2011, July). Evaluating machine learning algorithms for detecting DDoS attacks. In *International Conference on Network Security and Applications* (pp. 441-452). Springer, Berlin, Heidelberg
27. Mamolar, A. S., Pervez, Z., Calero, J. M. A., & Khattak, A. M. (2018). Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks. *Computers & Security*, *79*, 132-147.
28. Tuan, T. A., Long, H. V., Son, L. H., Kumar, R., Priyadarshini, I., & Son, N. T. K. (2020). Performance evaluation of Botnet DDoS attack detection using machine learning. *Evolutionary Intelligence*, *13*(2), 283-294.
29. Sinha, M., Gupta, S., Rout, S. S., & Deb, S. (2021). Sniffer: A machine learning approach for DoS attack localization in NoC-based SoCs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, *11*(2), 278-291.
30. Imamverdiyev, Y., & Abdullayeva, F. (2018). Deep learning method for denial of service attack detection based on restricted boltzmann machine. *Big data*, *6*(2), 159-169.

Sentence

ABSTRACT One of the most hazardous attack in the area of network security is a distributed denial of service (DDoS) attack. A DDoS assault disrupts the normal operation of essential services in a variety of online applications. Rather than offering services to actual users, systems facing DDoS attacks are kept occupied with fake requests (Bots). These attacks are becoming increasingly sophisticated and therefore are increasing on a daily basis. As a result, detecting these threats and protecting online businesses from them has grown harder. To address this issue we have developed a machine learning-based solution for detecting and classifying distinct types of network traffic. The suggested method is tested on a fresh dataset that includes a combination of different sorts of attacks such as DoS, HTTP floods and normal traffic. To recognise abnormal activity we used a range of machine learning algorithms including neural networks, Logistic Regression, and Decision Trees. Results showed hat random forest and decision tree have achieved higher accuracy compared to rest of the models. - 4 - INTRODUCTION In recent years, Internet services have become indispensable for both businesses and individuals. With the rising demand for network-based services, network hackers have escalated their attacks on these services in order to prevent genuine customers from receiving service. DDoS assaults are defined as attacks that halt or slows network services. Attackers launch a DDoS attack by taking control of millions of publicly accessible computers over the internet. As a result, servers block legitimate users' requests and remain busy with the attackgenerated requests. These attacks are more likely to target well-known websites, such as banks, social networking sites, and colleges. To safeguard essential data and services from attackers, more than one technology such as antivirus software, firewalls, and intrusion detection systems (IDS) should be employed in a computer network. An IDS is one of the most often utilised solutions for dealing with DDoS attacks. The confidentiality, integrity and availability of network resources are all safeguarded by an IDS. In order to detect and then classify distinct DDoS attempts and eliminate intrusion, the IDS system employs machine learning techniques. However, achieving 100% performance accuracy is very difficult. Various publicly available datasets are outdated, lacking newer and more recent types of attack traffic such as Smurf, SIDDOS and HITP flood. As a result, a dataset was required that included a variety of new attacks. Earlier we has used KDD Cup 1999 Data, which included four different attacks: SIDDoS, H IQflood, UDP-flood, and Smurf. We later went on to the NF-UQ-NIDS dataset accessible on cloudshare because the previous dataset was older. - 5 - LITERATURE REVIEW: - 6 - - 7 - - 8 - - 9 - - 10 - - 11 - - 12 - - 13 - PROPOSED MODEL: Architecture Smart Detection combats DDoS attacks by collecting and analysing network traffic samples, followed by subsequent classification into normal or malicious traffic based on various parameters. As soon as an attack is detected, notification messages are sent using cloud platform. Figure 1. Overview of model Classification is based on characteristics such as 1. average packet size 2. bit rate, 3. packet size 4. inter arrival time etc. These characteristics are used to classify network traffic as normal or DDoS attack. The average packet size in most DDoS attempts is the same. As a result of examining datasets, machine learning techniques determine if traffic in the network is attack traffic or normal traffic. In order to detect threats in network traffic Work Environment: Jupyter Notebook and Google Colab as our working environment. - 14 - Figure 2. High-level architecture Figure 3: Traffic Classification - 15 - WORKFLOW: A Signature Dataset (SDS) (dataset in use) and a machine learning algorithm are at the heart of the detection system. - First, signatures from normal traffic and DoS attacks were retrieved, classified, and saved in a database. Using feature selection approaches, SDS was then built. Finally, the MLA with the highest accuracy was chosen, trained, and placed into the traffic classification system. -The detection-system's architecture was created to deal with network traffic samples acquired from network devices using traffic sampling techniques of industrial standard. - Here, the receiver buffer, recieves unlabeled samples and groups them into flow tables. As and when, the table length exceeds or equals the reference value, subsequently they are handed over to the classifier for labelling. If the flow table is about to expire, it can be processed again. -Small flow tables are more common at lower sampling rates or during certain forms of DoS attacks, such as SYN flood attacks. -Traffic samples are acquired and stored in flow tables during each cycle of detection. In stages 1 and 2, a unique identificationnumber (FlowID) is calculated separately for each new flow based on 5 factors (src port, dst port, src IP, dst IP, and transport protocol). -If the flow is new, (i.e no other flow table has the same FlowID), it is then registered in a shared memory buffer. -If a previously registered flow table has the same FlowID as the new one , the existing flow's data will be merged with the new flow in stages 3 and 4. -If the table length exceeds or equals the reference value, clssification of the flow table takes place, and if an attack is detected, a notification is sent. Or else, it is put back into the memory buffer(shared) -In step 7, the cleanup task searches the shared buffer for expired flow tables, i.e flow tables that have surpassed the system's expiration time -The table length is checked for each of the expired flow tables by the system. If the length of the flow table is below then or equals the minimum reference value it is processed in stage 8. -A new FlowID is computed using 3-tuples (transport_protocol, src_IP and dst_IP), as the flow table is directed back to stages 3 and 4 - 16 - -In the presented flowchart, the detecting system's entire algorithm is summarised. Figure 4. Flowchart of workflow \ - 17 - DATASET USED: KDD Cup 1999 Data (uci.edu) For Training we have used the kddcup.data_10_percent.gz which is a 10% of the original dataset. For prediction of attack we have used kddcup.newtestdata_10_percent_unlabeled.gz - 18 - MODELS USED: 1. Logistic regression: It is a classification algorithm based on supervised learning, often used to predict probabilities of target variables. Because the dichotomous nature of the dependent there are only two classifications. In simple terms, the target variable has a binary nature, with data coded either 1 (representing success) or 0 (representing failure). - 19 - ii. Decision Tree Classifier: The Decision Tree algorithm is part of the supervised learning algorithm family. The decision tree approach, unlike the other supervised learning algorithms, could also be utilised to solve regression and classification problems. The goal of utilizing a Decision Tree is to build a training model that can be used to predict the class or the value of a target variables by learning decision rules deduced from previous data (training data). - 20 - iii. Support Vector Machines: It is a popular Supervised Machine Learning algorithm, used to solve both classification and regression tasks. The SVM algorithm's objective is to create the most optimum line or decision boundary which can categorize the n-dimensional space into classes, by which we can easily put new data points in correct categories in future. Hyperplane is the name for the best choice boundary. The extreme points that help to create the hyperplane are chosen via SVM. Support vectors are the extreme instances, and the algorithm is called a Support Vector Machine. - 21 - iv. Random Forest Classifier: Random Forest is a supervised learning model based on ensemble learning, which is a method of integrating several classifiers to solve complex problems and increase the model's performance. It contains decision trees on different subsets of the dataset and takes an average to enhance the predicted accuracy of the dataset in use. Rather than relying on a single decision tree, random forest collects the forecasts from each tree and predicts the final output based on the majority votes of predictions. - 22 - USING ANOTHER DATASET: Preprocessing: Using random forest model for prediction: - 23 - Appending our predictions in our test dataset: ACCURACY COMPARISON: Below is the table created from pandas data frame showing different accuracies of training testing and also the time required for training and testing. - 24 - CONCLUSION: Finally, we see that the Random Forest model achieves the highest accuracy on both training and tesing datasets. Logistic Regression is slow and also has less accuracy than others (SVM and Random Forest) SVC model is very slow but it has a very high accuracy, as it is very slow so can't be used in real life detection system. However, the Decision Tree model is the quickest in terms of training and testing time. It also gives fairly high accuracy. Hence, in terms of time and accuracy combined the best model out of the 4 is the Decision Tree model as it achieves high accuracy while also minimizing training and testing time. While if accuracy is prime focus then the Random Forest Model is to be preferred. Such a model can be used in intrusion detection systems to make these systems more robust and intelligent in detecting malicious attacks such as Denial of Service attacks.

Report Title:	ism
Report Link: (Use this link to send report to anyone)	https://www.check-plagiarism.com/plag-report/73933a8f4c6a826a51a55d4f2b484ccd493b11651255802
Report Generated Date:	29 April, 2022
Total Words:	1406
Total Characters:	9643
Keywords/Total Words Ratio:	0%
Excluded URL:	No
Unique:	**91%**
Matched:	**9%**

Sentence wise detail:

ABSTRACT One of the most hazardous attack in the area of network security is a distributed denial of service (DDoS) attack.

A DDoS assault disrupts the normal operation of essential services in a variety of online applications.

Rather than offering services to actual users, systems facing DDoS attacks are kept occupied with fake requests (Bots).

These attacks are becoming increasingly sophisticated and therefore are increasing on a daily basis.

As a result, detecting these threats and protecting online businesses from them has grown harder.

To address this issue we have developed a machine learning-based solution for detecting and classifying distinct types of network traffic.

The suggested method is tested on a fresh dataset that includes a combination of different sorts of attacks such as DoS, HTTP floods and normal traffic.

To recognise abnormal activity we used a range of machine learning algorithms including neural networks, Logistic Regression, and Decision Trees.

Results showed hat random forest and decision tree have achieved higher accuracy compared to rest of the models.

- 4 - INTRODUCTION In recent years, Internet services have become indispensable for both businesses and individuals.

With the rising demand for network-based services, network hackers have escalated their attacks on these services in order to prevent genuine customers from receiving service. (0)

DDoS assaults are defined as attacks that halt or slows network services.

Attackers launch a DDoS attack by taking control of millions of publicly accessible computers over the internet. As a result, servers block legitimate users requests and remain busy with the attackgenerated requests. (1)

These attacks are more likely to target well-known websites, such as banks, social networking sites, and colleges.

To safeguard essential data and services from attackers, more than one technology such as

antivirus software, firewalls, and intrusion detection systems (IDS) should be employed in a computer network.

An IDS is one of the most often utilised solutions for dealing with DDoS attacks.

The confidentiality, integrity and availability of network resources are all safeguarded by an IDS.

In order to detect and then classify distinct DDoS attempts and eliminate intrusion, the IDS system employs machine learning techniques.

However, achieving 100% performance accuracy is very difficult.

Various publicly available datasets are outdated, lacking newer and more recent types of attack traffic such as Smurf, SIDDOS and HITP flood.

As a result, a dataset was required that included a variety of new attacks.

Earlier we has used KDD Cup 1999 Data, which included four different attacks: SIDDoS, H IQflood, UDP-flood, and Smurf.

We later went on to the NF-UQ-NIDS dataset accessible on cloudshare because the previous dataset was older. - 5 - LITERATURE REVIEW: - 6 - - 7 - - 8 - - 9 - - 10 - - 11 - (2)

- 12 - - 13 - PROPOSED MODEL: Architecture Smart Detection combats DDoS attacks by collecting and analysing network traffic samples, followed by subsequent

classification into normal or malicious traffic based on various parameters.

As soon as an attack is detected, notification messages are sent using cloud platform. Figure 1.

Overview of model Classification is based on characteristics such as 1.

average packet size 2. bit rate, 3. packet size 4.

inter arrival time etc.

These characteristics are used to classify network traffic as normal or DDoS attack.

The average packet size in most DDoS attempts is the same.

As a result of examining datasets, machine learning techniques determine if traffic in the network is attack traffic or normal traffic.

In order to detect threats in network traffic Work Environment: Jupyter Notebook and Google Colab as our working environment. - 14 - Figure 2.

High-level architecture Figure 3: Traffic Classification - 15 - WORKFLOW: A Signature Dataset

(SDS) (dataset in use) and a machine learning algorithm are at the heart of the detection system. (3)

- First, signatures from normal traffic and DoS attacks were retrieved, classified, and saved in a database.

Using feature selection approaches, SDS was then built.

Finally, the MLA with the highest accuracy was chosen, trained, and placed into the traffic classification system.

-The detection-systems architecture was created to deal with network traffic samples acquired from network devices using traffic sampling techniques of industrial standard. (4)

-The detection-systems architecture was created to deal with network traffic samples acquired from network devices using traffic sampling techniques of industrial - Here, the receiver buffer, recieves unlabeled samples and groups them into flow tables. (5)

As and when, the table length exceeds or equals the reference value, subsequently they are handed over to the classifier for labelling.

If the flow table is about to expire, it can be processed again.

-Small flow tables are more common at lower sampling rates or during certain forms of DoS attacks, such as SYN flood attacks.

-Traffic samples are acquired and stored in flow tables during each cycle of detection. In stages 1 and 2, a unique identificationnumber (FlowID) is calculated separately for (6)

each new flow based on 5 factors (src port, dst port, src IP, dst IP, and transport protocol). (7)

-If the flow is new, (i.

e no other flow table has the same FlowID), it is then registered in a shared memory buffer.

-If a previously registered flow table has the same FlowID as the new one , the existing flow&#039;s data will be merged with the new flow in stages 3 and 4.

-If the table length exceeds or equals the reference value, clssification of the flow table takes place, and if an attack is detected, a notification is sent.

Or else, it is put back into the memory buffer(shared) -In step 7, the cleanup task searches the shared buffer for expired flow tables, i.

e flow tables that have surpassed the system&#039;s expiration time -The table length is checked for each of the expired flow tables by the system.

If the length of the flow table is below then or equals the minimum reference value it is processed in stage 8.

-A new FlowID is computed using 3-tuples (transport_protocol, src_IP and dst_IP), as the flow table is directed back

to stages 3 and 4 - 16 - -In the presented flowchart, the detecting system&#039;s entire algorithm is summarised.

Figure 4.

Flowchart of workflow \ - 17 - DATASET USED: KDD Cup 1999 Data (uci.

edu) For Training we have used the kddcup. data_10_percent.

gz which is a 10% of the original dataset.

For prediction of attack we have used kddcup.

newtestdata_10_percent_unlabeled.

gz - 18 - MODELS USED: 1.

Logistic regression: It is a classification algorithm based on supervised learning, often used to predict probabilities of target variables.

Because the dichotomous nature of the dependent there are only two classifications.

In simple terms, the target variable has a binary nature, with data coded either 1 (representing success) or 0 (representing failure). - 19 - ii.

Decision Tree Classifier: The Decision Tree algorithm is part of the supervised learning algorithm family.

The decision tree approach, unlike the other supervised learning algorithms, could also be utilised to solve regression and classification problems.

- 24 - CONCLUSION: Finally, we see that the Random Forest model achieves the highest accuracy on both training and tesing datasets.

Logistic Regression is slow and also has less accuracy than others (SVM and Random Forest) SVC model is very slow

but it has a very high accuracy, as it is very slow so can&#039;t be used in real life detection system.

However, the Decision Tree model is the quickest in terms of training and testing time.

It also gives fairly high accuracy.

Hence, in terms of time and accuracy combined the best model out of the 4

is the Decision Tree model as it achieves high accuracy while also minimizing training and testing time. (8)

While if accuracy is prime focus then the Random Forest Model is to be preferred.

Such a model can be used in intrusion detection systems to make these

systems more robust and intelligent in detecting malicious attacks such as Denial of Service

attacks.

Match Urls:

0: https://www.merriam-webster.com/dictionary/service

1: https://www.chegg.com/homework-help/questions-and-answers/following-e-mail-privacy-policy-stipulations--defines-legitimate-e-mail-users-b-explains-b-q2519929

2: https://www.sciencedirect.com/science/article/pii/S0040603112003917

3: https://www.scirp.org/journal/paperinformation.aspx?paperid=113889

4: https://www.dictionary.com/browse/standard

5: https://www.coursehero.com/file/p28dmcq/works-as-this-the-receiver-buffer-receives-data-until-the-buffer-is-filled-then/

6: https://www.chegg.com/homework-help/questions-and-answers/please-help-fill-calculated-rotations-error-ee-identification-unknown-difference-determine-q93722390

7: https://www.researchgate.net/publication/336520750_Smart_Detection_An_Online_Approach_for_DoSDDoS_Attack_Detection_Using_Machine_Learning

8: https://www.timeanddate.com/worldclock/

Keywords Density

One Word	2 Words	3 Words
data 3.11%	flow table 1.38%	random forest model 0.35%
attack 3.11%	decision tree 1.15%	machine learning algorithm 0.35%
flow 2.77%	random forest 0.92%	training testing time 0.35%
work 2.19%	machine learning 0.69%	traffic samples acquired 0.23%
class 2.08%	flow tables 0.69%	expired flow tables 0.23%

Plagiarism Report

By check-plagiarism.com