National College of Ireland

DDoS Attack prediction and classification at Application Layer for Web protocol using Kmeans – SVM Machine Learning Algorithm

MSc Research Project
Cybersecurity

Ramesh Jaiswar
Student ID: x20102691

School of Computing
National College of Ireland

Supervisor: Ross Spelman

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ramesh Jaiswar
Student ID:	X20102691
Programme:	MSC in Cybersecurity **Year:** 2020 – 2021
Module:	MSC Internship
Supervisor:	Ross Spelman
Submission Due Date:	16/08/2021
Project Title:	DDoS Attack prediction and classification at Application Layer for Web protocol using Kmeans – SVM Machine Learning Algorithm
Word Count:	6145
Page Count:	28

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……

Date: 16th August 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DDoS Attack prediction and classification at Application Layer for Web protocol using Kmeans – SVM Machine Learning Algorithm

Ramesh Jaiswar
x20102691

Abstract

Web servers are normally situated in a highly structured network architecture where they allow access to the external internet through backbones. However, the Application Layer DDoS attacks are real threats for those web servers, particularly for the organizational web servers. The intruder transmits the attack requests using legitimate HTTP requests, making it difficult for the detection systems to classify the attack traffic and legit traffic. This study proposes a novel model for identifying and classifying such attack traffics using semi-supervised machine learning algorithms. The model is applied to the CICIDS 2017 Dataset, which contains Application Layer DDoS attack characteristics. The model is created by using correlation analysis to select features and reduce the dataset's dimension, then applying K-Means Clustering to an unlabeled feature-selected dataset to generate clusters, which are then labeled based on their nature (Benign or Attack label), and finally feeding the labeled clustered dataset to Support Vector Machine to train and test the model. The model successfully classifies web traffic based on its nature (Benign or Attack traffic) and on evaluation the model outperforms on the tested dataset when compared to the available classification algorithms.

1. Introduction

Now a days the internet network infrastructure is victimized by various attack. These attacks target the network availability of hosts, and services, and the confidentiality and integrity of the network traffics. Distributed Denial of Service (DDoS) attacks are launched from multiple attack sources using Botnet mechanism, are a purposeful effort to render an online service unavailable to authorized users by exhausting the servers' resources[1]. Classification of the DDoS attack can be done in three forms, that is Volume-based attacks, Protocol Based Attacks, and Application layer attacks [2]. The report emphasizes on the Application Layer DDoS Attack. The primary objective of the attacker at the Application Layer is to target processes and to overuse specific website features to make them unusable[3].

Application layer attacks have become more effective because of the rise in nature of skills and strategies available to attackers. Ability to detect a DDoS attack for the HTTP/HTTPs protocol is complicated because such attacks sometimes seem to be a valid request. Failure to identify attack traffic can lead to shutting down services, gaining database access and stealing critical data, and then demanding a ransom to fix the problem. The cost of a DDoS attack includes not just monetary value, but also non-monetary elements such as harm to the organization's reputation, customer loss, and administrative expenses of locating susceptible nodes and repairing the damage[4]. To avoid these losses, businesses should implement a system that can identify and classify DDoS attack traffic, as well as stop it.

Web based DDoS attacks are just as productive as volume-based attacks since they emphasis on extremely complicated attack patterns instead of volume. As a response, researchers are concentrating on machine learning techniques in order to develop a system that can successfully identify attack traffic. The type of the machine learning model used to classify attack traffic has a significant influence in its development. When utilizing the unsupervised learning approach, training the system using unlabeled and high-dimensional datasets complicates cluster creation and consumes a substantial amount of time.

Both supervised and unsupervised learning approaches have proved beneficial and reliable in identifying DDoS attacks on the web protocol. In this report, we will address the use of semi-supervised machine learning to increase the speed and accuracy in recognizing DDoS attacks. CICIDS 2017 was used as the dataset for this model. Both benign and DDoS attack traffic are included in the dataset. With 79 features, the data comprises four types of DDoS attacks: DoS Hulk, DoS Slowloris, DoS Slowhttptest, and DoS Goldeneye [5]. Given the amount of the features in the dataset, the correlation coefficient approach is used to reduce the dataset's dimension/features, because too many features might cause computing difficulty. K-means Clustering is used to process the dataset with reduced features and unlabeled data. The clusters formed as a result of this processing is labeled based on their nature (Benign or Attack). After that, the Support Vector Machine Algorithm is employed on the labeled dataset for classification.

A detailed literature review was conducted to investigate and identify the best semi-supervised learning algorithm to employ. Prior research on supervised and unsupervised machine learning had also been investigated to establish norms for semi-supervised learning, which was used to detect DDoS attack traffic.

The following sections will be covered in the remainder of this article: Section 2 examines the previous research that has been done in a similar way, and then compare their views. Approach and methods used to create the model is discussed in Section 3. Section 4 comprises the design specification for the model. The proposed model will be implemented in Section 5, and the output of the model will be evaluated in Section 6. Finally, section 7 will bring our study to a closure with a conclusion and directions for any future work.

2. Related work

In order to completely understand the background of research conducted on DDoS attack identification and classification using multiples approaches, a detailed literature review was performed.

2.1 Use of Unsupervised Learning for DDoS attack detection

Because of the positive outcome of many investigations in this subject, machine-learning has seen widespread usage in virtually every field and business in recent years. The reliability of the data utilized, and the machine-learning methods used are at the heart of every successful ML-based study. Using publicly accessible datasets published by government and commercial entities, various prior researchers have made significant efforts in attempting to identify and categorize DDoS attacks over the internet. The study conducted by researchers at [6] highlights the use of unsupervised learning using MeanShift Clustering algorithm to detect the attack traffics by using an offline KDD 99 dataset. The dataset included the features of DDoS attack, Remote to Local Attack, User to Root Attack, and Probe Attack. The data traffic is normalized using K-means clustering and then supplied to Meanshift Clustering to accurately classify the attack. The implemented model was able to classify only the clusters of the DDoS attack. The model, on the other hand, did not shown to be accurate in detecting other attack traffic. Remote to Local and User to Root assaults were not detected by the model, although probing attacks were detected at a rate of roughly 6.5 percent.

Another study at [7] states the use of unsupervised machine learning to implement an intrusion detection system with high accuracy rate in detecting the DDoS attack by decreasing the false positive rate. The dataset used is NSL-KDD dataset and the model uses the five outlier detection classifiers which are SVM, Naïve Bayes, Logistic Regression, k-nearest neighbor, Random Forest. The model outperforms the classification, and the best accuracy detection is achieved only when used with Logistic Regression Classifier.

2.2 Classification of DDoS attack traffic using Supervised Machine Learning

Nevertheless, several supervised learning techniques are utilized to detect DDoS attacks; for illustration, in [8], the Naive Bayes machine learning method was employed to classify the attack data from the benign ones. It considered the significance of data pre-processing for various-sized training dataset and feature sets. In [9] Nguyen, et al., evaluates and analyzes the attack architecture at different stages to effectively determine the DDoS attack and minimize false positives. The analyzed data is also used to draw the variables based on the characteristics used in the KNN algorithm. Each aspect of the assault scenario is therefore established according to the specifications so that the attack can be identified at the initial stage.

By considering the usage of high Web traffic loads in an Application DDoS attack, the researchers at [10] presents a system architecture framework that incorporates three elements to identify the real-time attack traffic. This element includes filtering module, abnormal traffic detection module, and DDoS attack detection module which are built on Real-Time Frequency Vector Algorithm. The dataset used in this study was created using a traffic simulating tool and the web traffic generated by Sina Web Application. The proposed model significantly achieves the expected efficiency as it's based on traffic simulated using tool and real-time data. While assessing the model based on the dataset, major flaw identified in this research is that the response rate for the attack traffic is weak.

2.3 Application Layer DDoS attack detection using Deep Learning

One of several potential ways to control application layer DDoS mentioned by the researcher at [11] in is to use Deep Learning framework to appreciate and explore the characteristics of the attack traffic. The algorithm employs the neural network and the Auto - encoder modelling approach to create more than three stages of deep learning techniques. The primary purpose of this approach is to understand about the attack traffic by extracting the high-level features. The model is evaluated using two metrics, accuracy of traffic detection and false positive rate. The accuracy of detecting the attack traffic achieved by the model was 98.9% and roughly around 1.2 % of false positive rate is observed. The dataset used for this approach has the DDoS attack flavors of request flooding, asymmetric attacks, and session flooding.

2.4 Using Semi Supervised Machine learning for Classifying DDoS attack traffic

Another study carried out [12] to classify the DDoS attack traffic was by implementing Semi-Supervised ML. A Hybrid Feature Selection approach was combined with K-means Clustering to create the model. The feature selection technique is built utilizing Hadoop technology, in which the features are sorted in ascending order based on Key-Value pairs, and then the normalizing procedure is applied. The features are selected based on RSD (Ratio of average Sum of Squared Errors to cluster Distance) value lesser than Θ. The filtered dataset is supplied to K-means algorithm to cluster the data and applying algorithm of Radius on the labeled dataset. The dataset used for evaluation

was DARPA DDoS dataset, CAIDA dataset, CICIDS - DDoS attack 2017. The model achieves the detection rate of around 99.5% as proposed on the above said datasets.
DDoS attacks pose a substantial concern to data centers, and numerous security techniques have been implemented to identify them. A similar research was performed [13] by Xiao et al, which uses CKNN (KNN with correlation analysis) to identify the correlation and would then examine the stream of data in the data center before implementing efficient supervised machine learning technique to detect the DDoS attack. The journal [14] employs a Bayesian Classifier as well as other essential classifiers to recognize intrusions in aggregate while dealing with network modeling protocols such as TCP and UDP.

2.5 Determining the credit risk rating using Semi Supervised Learning

The procedure of applying Semi-Supervised machine learning is also seen in financial institutions to assess the credit risk rating of applicants, as the characteristics of a moderate applicant are identical with those of a poor applicant. The stated model uses the consensus and cluster-based models to define the rating of the risk [15]. To extract the features from the dataset, the K-means & Korhonen's self-organizing maps were utilized. Then, using a list of ML methods such as Logistic Regression, Decision Tree, Random Forest, artificial neural networks, and Support Vector Machines, Supervised Learning is implemented depending on the maximum classification accuracy achieved. The approach performs well to datasets examined, although it could be enhanced to reduce data noise.

3. Research Method and Specification

The research methodology used to perform this experiment is a product of the prior study [13] [15] stated in the above section. The approach utilizes the CICIDS 2017 Wednesday traffic dataset available at [5], the dataset comprises of web traffic features which are categorized under Benign and DDoS attack traffic. The dataset was encoded manually, after which a Correlation Coefficient test was used to determine the relationship between various features and to find the most essential ones. Following the identification of the essential features, a subset of the dataset is constructed based on those features, which is further used for creating clusters using K-means and finally using SVM for classifying whether the traffic belongs to Benign or DDoS attack. The next subsections will go over the entire method in great depth.

3.1 Data gathering/ selection

The attackers employ a range of bot-based technologies to execute DDoS attacks against several businesses, and these firms refuse to reveal the log files or evidence of the attack due to the company's reputation and the security of the data. And to conduct this research, a comprehensive examination of well-known DDoS attack simulation programs such as Spybot, SDBot, and others was considered to learn and construct the dataset.

However, the feasibility of launching a DDoS attack on a Web application requires a highly configured lab set up with web servers, data servers [16]. As such, one of the dataset used in [12] is utilized. The Dataset contains 79 features and 692703 records of traffic belonging to class - Benign, DDoS Hulk, DDoS Goldeneye, DDoS Slowloris, and DDoS Slow Http.

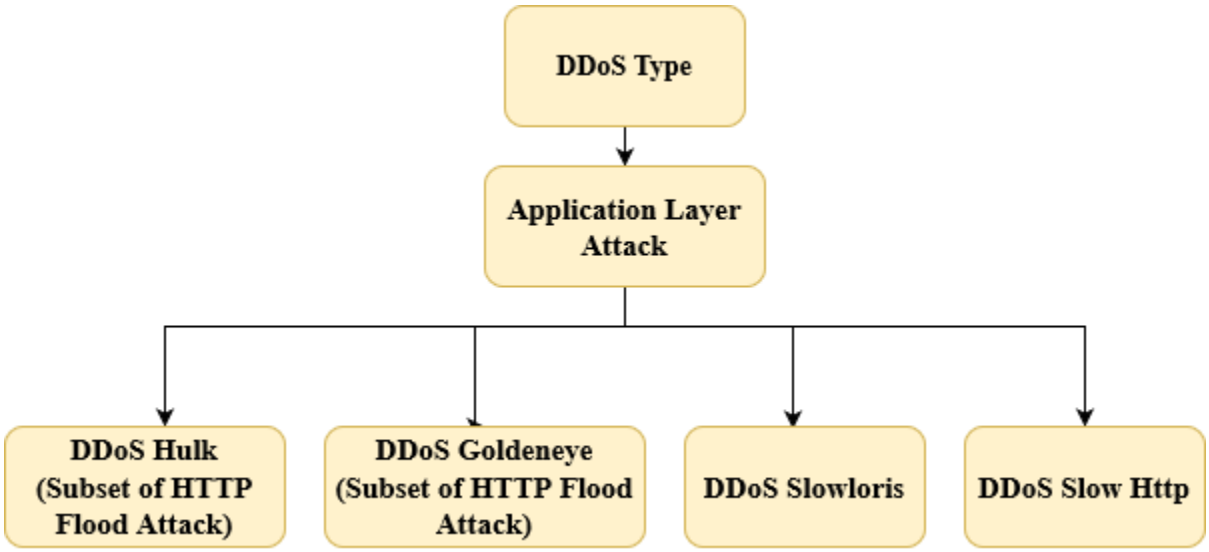

Fig.1 Dataset Tree

The number of records per DDoS attack type is stated in the below table:

Attack Type	Count of Traffic
DDoS Hulk	231073
DDoS Goldeneye	10293
DDoS Slowloris	5499
DDoS Slow Http	5796

Table 1

3.2 Data Pre-processing

The data-gathering phase was completed, and the final product generated a CSV file with both benign and attack traffic for each attack type. However, there were some rows in the dataset with infinite values and NaaN (Not a Number) values. This problem was solved by amending the Python code so that rows with infinite values are rounded to the maximum length of the data type of the variable and NaaN values are dropped.

3.3 Feature Extraction using Coefficient correlation

A correlation test was run using Python code to determine the relation between the various features as well as how the feature would complement the other [17], and the findings were represented as shown below.

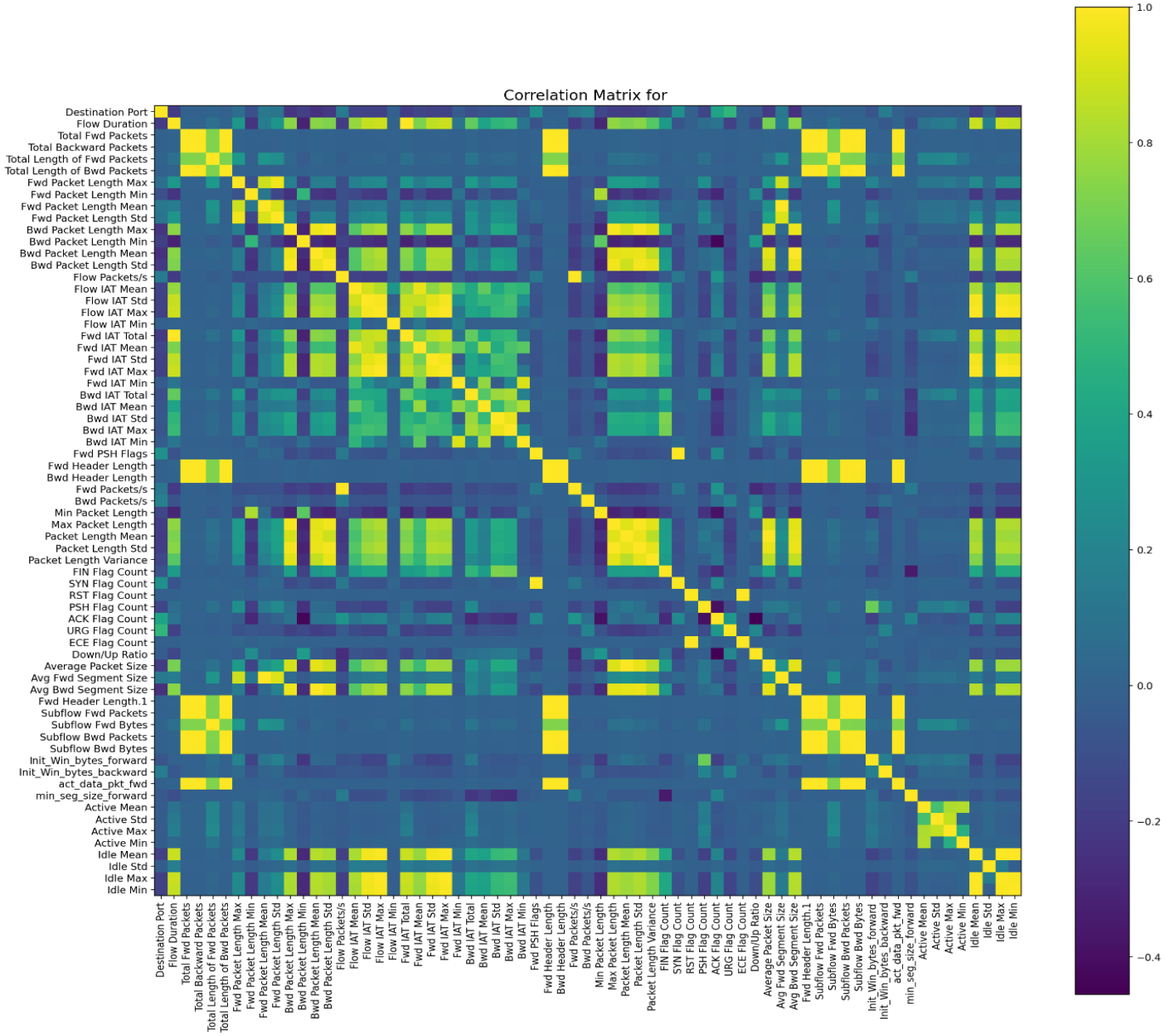

Fig.2 Correlation Test Plot

The outcome indicates the positive and negative correlations between the various variables, allowing us to better comprehend their relationship. Since, number of variables used in correlation analysis were high, a separate function was created in python to export the features which has strong relationship with target variable 'Label'. The threshold value for filtering these features was set to 0.99. Below table states the features extracted using correlation analysis:

Feature Names	Data Type
Average Packet Size	Float 64
Avg Bwd Segment Size	Float 64
Avg Fwd Segment Size	Float 64
Bwd Header Length	Int 64
Fwd Header Length	Int 64
Fwd IAT Max	Int 64
Idle Max	Int 64
Idle Min	Int 64
Subflow Bwd Bytes	Int 64
Subflow Bwd Packets	Int 64
Subflow Fwd Bytes	Int 64
Total Backward Packets	Int 64
Total Length of Bwd Packets	Int 64
act_data_pkt_fwd	Int 64
Fwd IAT Total	Int 64
Fwd Packets/s	Float 64
Idle Mean	Int 64
Subflow Fwd Packets	Int 64

Table 2. Extracted Features

3.4 Encoding of Categorical Data

It is essential to encode categorical variables in machine learning to ensure that the algorithm does not cluster identical results or entries that are adjacent to one another in one branch while being trained. Encoding is accomplished using One-Hot Encoding as for this model [18]. The One-Hot Encoding was chosen over Label Encoding because Label Encoding encodes the data and provides a rating system between the various values. The following table highlights the information about encoding values in the dataset.

Dataset Type	Encoded Variable	Values
Benign and DDoS hulk	Label	Benign - 0 DDoS Hulk - 1
Benign and DDoS Goldeneye	Label	Benign - 0 DDoS Goldeneye - 1
Benign and DDoS Slowloris	Label	Benign - 0 DDoS Slowloris - 1
Benign and DDoS Slow Http	Label	Benign - 0 DDoS Slow Http - 1

Table 3. Dataset Encoding Information

3.5 Training & Testing of data

The encoded subsets of the dataset are then trained using the below algorithms:

K-means: The K-means algorithm is one of the most widely used clustering methods is employed in this model. The algorithm splits encoded dataset into k mutually exclusive clusters, and the number clusters are created based on the number of unique data points (0's or 1') [19]. The efficiency of creating the clusters depends on the value of K. For our model, the value of K is identified using the elbow method [20], which is very effective in obtaining the optimal value of K.

The elbow method plots the various features of data against the changing values of K. The graph below was created with the help of python code. At K = 3, an elbow is formed, is a point after which distortion of values declines. Bearing this in mind, and to limit the likelihood of outliers, the value of k using the k-means algorithm is 2.

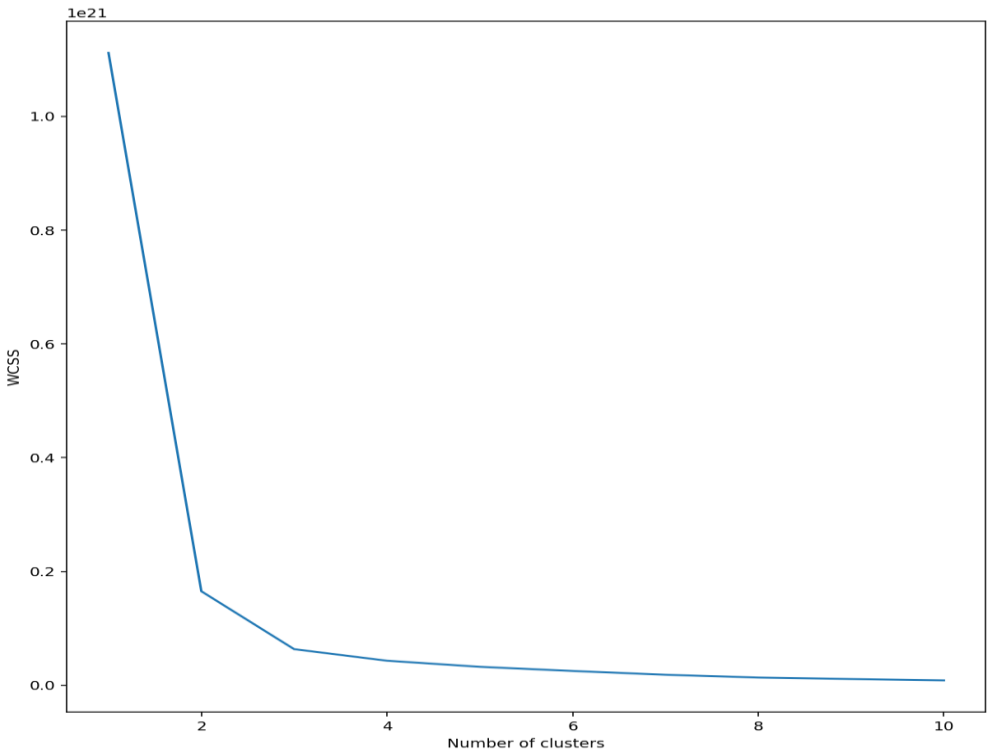

Fig. 3 – Elbow method graph

The dataset's labeling is removed before it is fed into the K-means algorithm, and the labeled data is then used as a benchmark to manually calculate the model's accuracy. Following the execution of the algorithm, an array of label values is formed, and the data values are clustered based on their properties. The data points in the plot overlap on each other resulting in the data distribution is not evenly distributed.

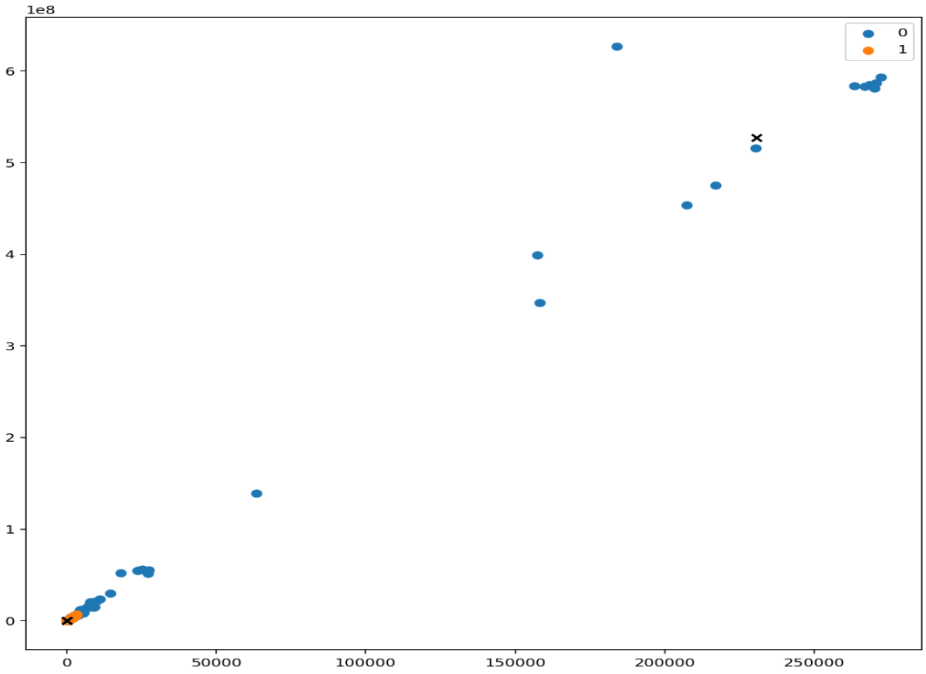

Fig. 4 Clusters=2

Support Vector machine: The dataset that was previously supplied to the K-means algorithm is appended to the array of 'Label' data obtained as the result of K-means Clustering. This yields a labeled dataset, which is then used to train the model along with labeled dataset. The model is trained to classify data traffic using the SVM algorithm.

Km-SVM Model: Consequently, a Km-SVM model has been developed and evaluated. This was accomplished by conducting SVM on the clustered outputs from K-Mean. The model is trained in batches of the dataset and then evaluated. The outcome of the model is generated in the form of a confusion matrix using which the accuracy of the model is calculated, and a comparison of actual and predicted values is made.

4. Design Specification

The structure of our developed model is discussed in this section. This section acts as a complete tool for classifying data traffic based on its nature. The model can take inputs in the form of CSV files, which are used further for data preprocessing, performing feature selection by applying correlation coefficient analysis to it. The extracted features are then encoded, and the subset of the dataset is created. The newly created dataset is sent to train the model (labeled data to SVM and unlabeled data to K-means) and then to perform classification on the test data.

Figure 5 shows the model's complete architecture. The dataset chosen is CICIDS 2017, which contains the web traffic captured in the form of a PCAP file and is converted into a CSV file. Before grouping the dataset into subsets, it is sent for feature extraction against the target variable 'Label', as all the attack groups share the same features of the web traffic. The features are extracted using correlation analysis, after which the dataset is divided into four subsets: 1) Benign & DDoS Hulk dataset 2) Benign & DDoS Goldeneye Dataset 3) Benign & DDoS Slowloris Dataset 4) Benign & Slow Http. The values of the target feature are encoded and then sent for training the model.

The K-means algorithm is utilized after the target feature is removed from the dataset. The program treats the dataset as unlabeled and groups data points together based on their characteristics. As a result of the k-means algorithm, an array of ‘Label' values is formed. The labeled set is generated by adding the array of ‘Label' values to the subset of the dataset that was previously submitted to K-means. SVM divides labeled datasets into X and y, with X containing all independent variables and y containing the dependent binary variable we want to predict. When splitting the train and test data, the size of the test split is set at 40% of the data, with the remaining 60% being used for training. The model's accuracy is calculated using a confusion matrix and an actual vs predicted matrix, which are generated as outputs.

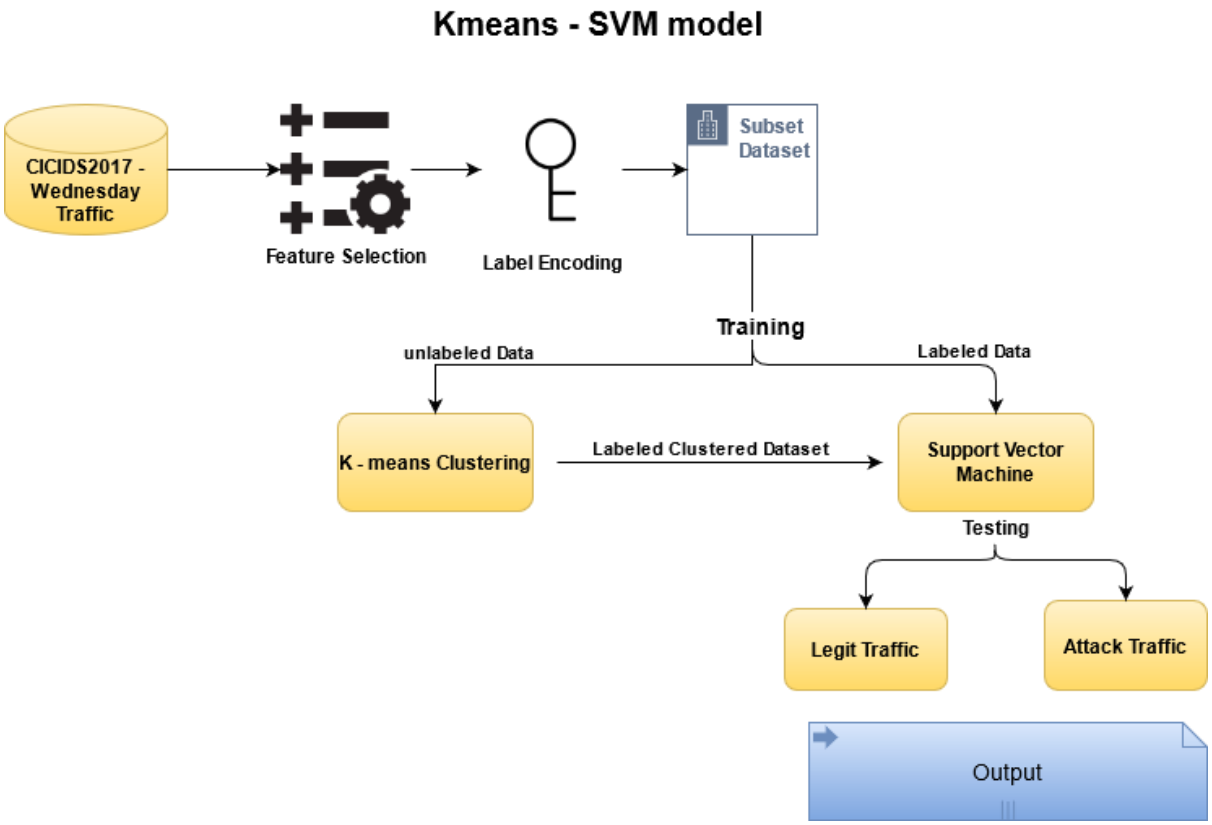

Fig. 5 Kmeans – SVM model Architecture

5. Implementation

This segment will go over the steps taken to put our proposed concept into action. We'll go over the hardware and software utilized, as well as the coding framework in depth.

5.1 Hardware

An HP laptop utilized for building this model with the following hardware specifications:

- CPU: Intel 6th Gen i7 Processor with 2.4 GHZ
- RAM: 16gb DDR4
- Storage: 1tb HDD with 256gb SSD
- GPU: AMD Radeon 2 GB

5.2 Software

Windows 10 (64 bit) is the host operating system on which the model was built. Also, the below list of software applications is used:

- Jupyter Notebook is used as development environment
- Python 3 is used as the coding language used for developing the model
- Libraries Included – Pandas, Numpy, Matplotlib, Seaborn, SVM, Kmeans, Sklearn

5.3 Data files

Final_Model_Kmeans_SVM.ipynb: This file includes the complete code used creating the model. The coding was done in juypter notebook using python 3.

Dataset: The dataset used for building this model varies at different stages, below is the details of the dataset along with their stages

Dataset File Name	Stage Description
Dataset_DDoS1.csv	Feature selection
Dataset_DDoS1_Benign_DDosHulk_18 Features_Unlabelled	Unlabeled Dataset fed to Kmeans Algorithm for Clustering
Kmeans_labelled.csv	Output Dataset derived from Kmeans Clustering, is fed to SVM Algorithm for training and testing
Dataset_DDoS1_Benign_DDosHulk_18 Features_labelled.csv	Along with Kmeans Dataset, the base dataset with labelled data is sent to SVM for training and testing.

6. Evaluation

To evaluate the efficiency of the model, we have created test scenarios against which the model will be tested with the datasets and the DDoS attack types. The results/observations are document along with the test scenarios.

6.1 Testing the model with untrained dataset of same DDoS attack type

- **State of the model**: The model is trained only using the attack type DDoS Hulk
- **Test scenario**: to verify the accuracy of the model when using untrained dataset of same DDoS type - HULK DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Hulk is used for testing
- **Dataset Name:** 2ndSet_Benign_DDosHulk_150L_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 150,000
- **Results:** The model was tested using web traffic containing both benign and DDoS Hulk traffic. Figure 8. depicts the Confusion matrix and shows us that **out of 150000**

web traffic, the model can accurately predict 130500 web traffic. And the model correctly predicts the DDoS Hulk traffic **with an accuracy of 89%.** Therefore, we obtained both an **accuracy and precision of 87%.** Figure 6 shows the subset of the dataset used for testing.

There are 150000 rows and 19 columns

Total Backward Packets	Total Length of Bwd Packets	Fwd IAT Total	Fwd IAT Max	Bwd Header Length	Fwd Packets/s	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes	act_data_pkt_fwd	Idle Mean
2	138	32703	32696	40	79.074825	57.166667	41.0	69.0	104	4	164	2	138	3	0.0
2	96	2	2	40	8230.452675	48.000000	32.0	48.0	40	2	64	2	96	1	0.0
2	108	4	4	64	12048.192770	55.500000	38.0	54.0	64	2	76	2	108	1	0.0
2	256	4	4	64	64.443370	99.250000	47.0	128.0	64	2	94	2	256	1	0.0
2	322	33405	33399	64	10.930007	96.166667	51.0	161.0	80	4	204	2	322	3	0.0

Fig. 6 TS01_Dataset

```
In [45]: accuracy_score(y4, y4_test_pred)
Out[45]: 0.87264

In [46]: df=pd.DataFrame({'Actual':y4, 'Predicted':y4_test_pred})
         df
```

Out[46]:

	Actual	Predicted
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
149995	1	1
149996	1	1
149997	1	1
149998	1	1
149999	1	1

150000 rows × 2 columns

Fig. 7 TS01_Actual vs Expected

	precision	recall	f1-score	support
0	0.87	0.95	0.91	100000
1	0.89	0.71	0.79	50000
accuracy			0.87	150000
macro avg	0.88	0.83	0.85	150000
weighted avg	0.87	0.87	0.87	150000

Fig.8 TS01_Confusion matrix

6.2 Testing the model with untrained dataset of DDoS attack type – Goldeneye

- **State of the model**: The model is trained only using the attack type DDoS Hulk
- **Test scenario**: To verify the accuracy of the model when using untrained dataset of DDoS type -Goldeneye DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Hulk is used for testing
- **Dataset Name:** 3rd_Set_Benign_DDosGoldenEye_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 30293
- **Results:** The model in this test scenario has not been trained or tested for DDoS Goldeneye attacks. The same classification model that was trained on DDoS Hulk attacks is used here. The collection includes 30293-web traffic, which consists of benign and DDoS Goldeneye traffic. Figure 11 illustrates the Confusion matrix, which shows that **the model can effectively predict 19690 web traffic out of 30293 web traffic.**

  Since **the model was never trained on DDoS Goldeneye attack traffic**, the prediction rate of **DDoS Goldeneye is low when compared to DDoS Hulk**. Even obtaining a 45% accuracy rate in detecting the attack traffic is a significant accomplishment for this approach. The model is trained on this attack type in the following test scenarios, and the prediction rate is calculated.
  The overall accuracy rate achieved by the model is 65%.

There are 30293 rows and 19 columns

	Total Backward Packets	Total Length of Bwd Packets	Fwd IAT Total	Fwd IAT Max	Bwd Header Length	Fwd Packets/s	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes	act_data_pkt_fw
0	1	6	0	0	20	26.104208	9.000000	6.000000	6.000000	20	1	6	1	6	
1	5	326	479	109	176	22964.509390	31.125000	15.636364	65.200000	368	11	172	5	326	
2	6	3150	1095	915	208	9132.420091	393.750000	315.000000	525.000000	336	10	3150	6	3150	
3	12	6660	15206	13391	388	1117.979745	348.689655	203.058823	555.000000	560	17	3452	12	6660	1
4	6	3152	1092	910	208	8241.758242	420.133333	350.000000	525.333333	304	9	3150	6	3152	

Fig.9 TS02_Dataset

```
df=pd.DataFrame({'Actual':y5, 'Predicted':y5_test_pred})
df
```

	Actual	Predicted
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
30288	1	0
30289	1	0
30290	1	0
30291	1	0
30292	1	0

30293 rows × 2 columns

Fig.10 TS02_Actual vs Expected

	precision	recall	f1-score	support
0	0.67	0.92	0.78	20000
1	0.45	0.13	0.20	10293
accuracy			0.65	30293
macro avg	0.56	0.52	0.49	30293
weighted avg	0.60	0.65	0.58	30293

Fig.11 TS02_Confusion Matrix

6.3 Testing the model with untrained dataset of DDoS attack type – Slow Http

- **State of the model**: The model is trained only using the attack type DDoS Hulk
- **Test scenario**: To verify the accuracy of the model when using untrained dataset of DDoS type -Slow Http DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Hulk is used for testing
- **Dataset Name:** 4th_Set_Benign_DDosSlowHttp_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 15499
- **Results:** The objective of this test scenario is similar to that of TS02, the model has not been trained or tested for DDoS Slow Http attacks. The same classification model that was trained on DDoS Hulk attacks is used here. The collection includes 15499-web traffic, which consists solely of benign and DDoS Slow Http traffic. Figure 14 illustrates the Confusion matrix, which shows that the model can effectively predict 11005 web traffic out of 15499 web traffic.
- In this test case, the model **accurately predicted the precision for attack traffic 95% of the time, which is a great perk**. However, **due to the poor rate of recall &**

F1 Score attained in detecting attack traffic, the model's total accuracy rate is reduced to 71%.

- The model is trained on this attack type in the following test scenarios, and the prediction rate is calculated.

There are 15499 rows and 19 columns

	Total Backward Packets	Total Length of Bwd Packets	Fwd IAT Total	Fwd IAT Max	Bwd Header Length	Fwd Packets/s	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes	a
0	2	264	4	4	40	23.847282	88.5	30.0	132.0	40	2	60	2	264	
1	2	162	3	3	40	87.958484	72.0	42.0	81.0	40	2	84	2	162	
2	1	0	0	0	32	17857.142860	0.0	0.0	0.0	32	1	0	1	0	
3	2	404	3	3	64	8888.888889	156.5	74.0	202.0	64	2	148	2	404	
4	2	284	46	46	80	8928.571429	105.5	46.0	142.0	80	2	92	2	284	

Fig 12. TS03_Dataset

```
df=pd.DataFrame({'Actual':y6, 'Predicted':y6_test_pred})
df
```

	Actual	Predicted
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
15494	1	0
15495	1	0
15496	1	0
15497	1	0
15498	1	0

15499 rows × 2 columns

Fig 13. TS03_Actual vs Predicted

	precision	recall	f1-score	support
0	0.69	1.00	0.81	10000
1	0.95	0.18	0.30	5499
accuracy			0.71	15499
macro avg	0.82	0.59	0.56	15499
weighted avg	0.78	0.71	0.63	15499

Fig 14. TS03_Confusion Matrix

6.4 Testing the model with untrained dataset of DDoS attack type – Slowloris

- **State of the model**: The model is trained only using the attack type DDoS Hulk
- **Test scenario**: To verify the accuracy of the model when using untrained dataset of DDoS type -Slowloris DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Hulk is used for testing
- **Dataset Name:** 5th_Set_Benign_DDosSlowloris_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 15795
- **Results:** The results from this test scenario achieves the similar accuracy rate that of TS03. Here the model has not been trained or tested for DDoS Slowloris attack and same classification model is used that was trained on DDoS Hulk attack. The collection includes 15795-web traffic, which consists solely of benign and DDoS Slowloris traffic. The Confusion matrix displayed under figure 17, depicts that the model can effectively predict 11214 web traffic out of 15499 web traffic.

  In this test case, the model reacts similarly to TS03 in **accurately predicting the Slowloris attack traffic, the precision score achieved is 96%**. But as we see that **the poor accuracy rate attained in recall & F1 score for attack traffic has reduced the model's total accuracy rate to 71%.**

There are 15795 rows and 19 columns

	Total Backward Packets	Total Length of Bwd Packets	Fwd IAT Total	Fwd IAT Max	Bwd Header Length	Fwd Packets/s	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes	act
0	1	0	0	0	32	10.554090	0.00	0.0	0.0	32	1	0	1	0	
1	1	6	0	0	20	8547.008547	9.00	6.0	6.0	20	1	6	1	6	
2	2	134	3	3	40	32.871489	62.75	39.0	67.0	40	2	78	2	134	
3	2	0	0	0	64	3.581623	0.00	0.0	0.0	32	1	0	2	0	
4	1	0	0	0	32	15873.015870	0.00	0.0	0.0	32	1	0	1	0	

Fig 15. TS04_Dataset

```
df=pd.DataFrame({'Actual':y7, 'Predicted':y7_test_pred})
df
```

	Actual	Predicted
0	0	0
1	0	0
2	0	0
3	0	0
4	0	0
...	...	...
15790	1	0
15791	1	0
15792	1	0
15793	1	0
15794	1	0

15795 rows × 2 columns

Fig 15. TS04_Actual vs predicted

	precision	recall	f1-score	support
0	0.69	0.99	0.81	9999
1	0.96	0.21	0.35	5796
accuracy			0.71	15795
macro avg	0.82	0.60	0.58	15795
weighted avg	0.79	0.71	0.64	15795

Fig 15. TS04_Confusion Matrix

6.5 Testing the model with trained dataset of DDoS attack type – Goldeneye

- **State of the model**: The model is trained for attack type DDoS Goldeneye
- **Test scenario**: To verify the accuracy of the model when using trained dataset of DDoS type -Goldeneye DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Goldeneye is used for testing
- **Dataset Name:** 3rd_Set_Benign_DDosGoldenEye_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 12118
- **Results:** The model in this test scenario has been trained on the dataset used for testing. The dataset contains 30293-web traffic out of which 40% of the data used for testing of the model and the rest for training. The newly created classification model that has trained on DDoS Goldeneye attack is used here. From figure 18, we can see that the model outperforms in detecting both benign and Goldeneye attack traffic, that's the reason why we obtain the accuracy score of 99.90%

```
accuracy_score(y8_test,y8_test_pred)
```

```
0.9990922594487539
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1.0	6.0	0.0	0.0	20.0	26.104208	9.000000	6.000000	6.000000	20.0	1.0	6.0	1.0	6.0	0.0	0.0	0.0
1	5.0	326.0	479.0	109.0	176.0	22964.509390	31.125000	15.636364	65.200000	368.0	11.0	172.0	5.0	326.0	4.0	0.0	0.0
2	6.0	3150.0	1095.0	915.0	208.0	9132.420091	393.750000	315.000000	525.000000	336.0	10.0	3150.0	6.0	3150.0	3.0	0.0	0.0
3	12.0	6660.0	15206.0	13391.0	388.0	1117.979745	348.689655	203.058823	555.000000	560.0	17.0	3452.0	12.0	6660.0	10.0	0.0	0.0
4	6.0	3152.0	1092.0	910.0	208.0	8241.758242	420.133333	350.000000	525.333333	304.0	9.0	3150.0	6.0	3152.0	2.0	0.0	0.0

Fig.16 TS05_Dataset

```
df=pd.DataFrame({'Actual':y8_test, 'Predicted':y8_test_pred}) #
df
```

	Actual	Predicted
17972	1	0
15549	1	0
1229	1	0
6227	1	0
22091	1	0
...	...	...
6373	0	0
18569	1	0
19423	1	0
27659	1	0
18245	1	0

12118 rows × 2 columns

Fig. 17 TS05_Actual vs Expected

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2223
1	1.00	1.00	1.00	9895
accuracy			1.00	12118
macro avg	1.00	1.00	1.00	12118
weighted avg	1.00	1.00	1.00	12118

Fig. 18 TS05_Confusion Matrix

6.6 Testing the model with trained dataset of DDoS attack type – Slow Http DDoS

- **State of the model**: The model is trained for attack type DDoS Slow http
- **Test scenario**: To verify the accuracy of the model when using trained dataset of DDoS type -Slow Http DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Slow Http is used for testing
- **Dataset Name:** 4th_Set_Benign_DDosSlowHttp_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 6200
- **Results:** The model in this test scenario has been trained and tested using DDoS Slow Http dataset. The dataset contains 15499-web traffic out of which 40% of the data used for testing of the model and the rest for training. The classification model used for testing the model was built using the trained dataset. Similar test results are achieved when compared with TS05, the model outperforms in predicting both benign and Slow Http attack traffic, that's the reason why we obtain the accuracy score of 99.90%.

```
accuracy_score(y9_test,y9_test_pred)
```

```
0.9996774193548387
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	2.0	264.0	4.0	4.0	40.0	23.847282	88.5	30.0	132.0	40.0	2.0	60.0	2.0	264.0	1.0	0.0	0.0	0.0
1	2.0	162.0	3.0	3.0	40.0	87.958484	72.0	42.0	81.0	40.0	2.0	84.0	2.0	162.0	1.0	0.0	0.0	0.0
2	1.0	0.0	0.0	0.0	32.0	17857.142860	0.0	0.0	0.0	32.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
3	2.0	404.0	3.0	3.0	64.0	8888.888889	156.5	74.0	202.0	64.0	2.0	148.0	2.0	404.0	1.0	0.0	0.0	0.0
4	2.0	284.0	46.0	46.0	80.0	8928.571429	105.5	46.0	142.0	80.0	2.0	92.0	2.0	284.0	1.0	0.0	0.0	0.0

Fig. 19 TS06_Dataset

```
df=pd.DataFrame({'Actual':y9_test, 'Predicted':y9_test_pred})
df
```

	Actual	Predicted
7800	0	0
3669	0	0
12126	1	1
13410	0	0
9329	0	0
...	...	...
12920	0	0
1551	0	0
5600	1	1
9610	0	0
4293	0	0

6200 rows × 2 columns

Fig. 20 TS06_Actual vs Predicted

	precision	recall	f1-score	support
0	1.00	1.00	1.00	5773
1	1.00	1.00	1.00	427
accuracy			1.00	6200
macro avg	1.00	1.00	1.00	6200
weighted avg	1.00	1.00	1.00	6200

Fig. 20 TS06_Confusion Matrix

6.7 Testing the model with trained dataset of DDoS attack type – Slowloris DDoS

- **State of the model**: The model is trained for attack type DDoS Slowloris
- **Test scenario**: To verify the accuracy of the model when using trained dataset of DDoS type -Slowloris DDoS
- **Classification model**: The SVM classification model trained using k-means dataset and some labeled dataset of attack type DDoS Slowloris is used for testing
- **Dataset Name:** 5th_Set_Benign_DDosSlowloris_Rows_18Features_Encoded_labelled.csv
- **Count of Web traffic used for testing:** 6318
- **Results:** The model in this test scenario has been trained and tested using DDoS Slowloris attack dataset. The dataset contains 15795-web traffic out of which 40% of the data used for testing of the model and the rest for training. The classification model used for testing the model was built using the trained dataset. The overall

accuracy of the model for predicting the DDoS Slowloris Attack is 88.99% which meets the threshold of the proposed model.

```
accuracy_score(y10_test,y10_test_pred)
```

```
0.8899968344412789
```

There are 15795 rows and 19 columns

Total Backward Packets	Total Length of Bwd Packets	Fwd IAT Total	Fwd IAT Max	Bwd Header Length	Fwd Packets/s	Average Packet Size	Avg Fwd Segment Size	Avg Bwd Segment Size	Fwd Header Length	Subflow Fwd Packets	Subflow Fwd Bytes	Subflow Bwd Packets	Subflow Bwd Bytes
1	0	0	0	32	10.554090	0.00	0.0	0.0	32	1	0	1	0
1	6	0	0	20	8547.008547	9.00	6.0	6.0	20	1	6	1	6
2	134	3	3	40	32.871489	62.75	39.0	67.0	40	2	78	2	134
2	0	0	0	64	3.581623	0.00	0.0	0.0	32	1	0	2	0
1	0	0	0	32	15873.015870	0.00	0.0	0.0	32	1	0	1	0

Fig. 21 TS07_Dataset

```
df=pd.DataFrame({'Actual':y10_test, 'Predicted':y10_test_pred})
df
```

	Actual	Predicted
13259	1	0
8222	0	0
5440	0	0
1201	0	0
3371	0	0
...	...	...
5521	0	0
6373	0	0
3479	0	0
6268	0	0
3877	0	0

6318 rows × 2 columns

Fig. 22 TS07_Actual vs Predicted

	precision	recall	f1-score	support
0	0.88	1.00	0.94	5076
1	1.00	0.44	0.61	1242
accuracy			0.89	6318
macro avg	0.94	0.72	0.77	6318
weighted avg	0.90	0.89	0.87	6318

Fig. 23 TS07_Confusion Matrix

6.5 Summary of the Evaluation

Upon training our model with several datasets and evaluating its efficiency, we discovered that our model had an average prediction accuracy of 83 percent throughout all seven test scenarios, also with least accuracy rate of 65 percent and the highest prediction accuracy of 99.9 percent. The initial conclusion drawn from TS05 and TS06 was that the model achieves the highest accuracy with small datasets. However, after analyzing the TS07, we discovered that while the size of the dataset has little influence on accuracy, the type of attack dataset employed does.

Test Scenario	Model trained on Dataset	Dataset Used for Testing	Accuracy Obtained	Number of Data traffic used for Testing
TS01	DDoS Hulk	DDoS Hulk	87%	150000
TS02	DDoS Hulk	DDoS Goldeneye	65%	30293
TS03	DDoS Hulk	DDoS Slow Http	71%	15499
TS04	DDoS Hulk	DDoS Slowloris	71%	15795
TS05	DDoS Goldeneye	DDoS Goldeneye	99.90%	12118
TS06	DDoS Slow Http	DDoS Slow Http	99.90%	6200
TS07	DDoS Slowloris	DDoS Slowloris	88.90%	6318

7. Conclusion and Future Work

Our research and model evaluation illustrate that the stated hypothesis of employing K-means and SVM to distinguish DDoS attack traffic from benign data at application layer is accurate. The proposed model achieves the maximum accuracy of 99.9% on the variety of attack datasets used in testing. The study's findings suggest that the model's accuracy prediction is not influenced by the size of the dataset. It's also important to consider the nature and type of attack dataset provided to the model. Finally, using Correlation for feature selection with K-means and SVM to predict DDoS attack traffic is highly effective.

We were also unable to construct a dataset for DDoS attacks by simulating the attack traffic using tools, on a web application and collecting the traffic using Wireshark due to time restrictions and hardware limitations. As a result, with a newly constructed dataset in a real-time environment, we are dubious of the model's prediction strength. In future, we'd like to test our model's capabilities against different cyber-attacks, such as Malware or phishing attempts, by employing larger, real-time datasets with a larger number of attributes.

References

[1] Cloudflare.inc, “What is a distributed denial-of-service (DDoS) attack?,” *Cloudflare*. https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/ (accessed Aug. 14, 2021).

[2] “DDoS Attack Types & Mitigation Methods | Imperva,” *Learning Center*. https://www.imperva.com/learn/ddos/ddos-attacks/ (accessed Aug. 14, 2021).

[3] “What is an Application Layer DDoS Attack?,” *NETSCOUT*. https://www.netscout.com/what-is-ddos/application-layer-attacks (accessed Aug. 14, 2021).

[4] “Reasons Why Every Business Need DDoS Protection | Indusface Blog,” *Indusface*, Jul. 26, 2019. https://www.indusface.com/blog/reasons-why-business-need-ddos-protection/ (accessed Aug. 14, 2021).

[5] “IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB.” https://www.unb.ca/cic/datasets/ids-2017.html (accessed Aug. 15, 2021).

[6] A. Kumar, W. B. Glisson, and R. Benton, “Network Attack Detection using an Unsupervised Machine Learning Algorithm,” *Hawaii Int. Conf. Syst. Sci. 2020 HICSS-53*, p. 10, Accessed: Feb. 08, 2021. [Online]. Available: https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1762&context=hicss-53

[7] S. Das, D. Venugopal, and S. Shiva, “A Holistic Approach for Detecting DDoS Attacks by Using Ensemble Unsupervised Machine Learning,” in *Advances in Information and Communication*, Cham, 2020, pp. 721–738. doi: 10.1007/978-3-030-39442-4_53.

[8] R. Vijayasarathy, S. V. Raghavan, and B. Ravindran, “A system approach to network modeling for DDoS detection using a Naïve Bayesian classifier,” in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, Jan. 2011, pp. 1–10. doi: 10.1109/COMSNETS.2011.5716474.

[9] H.-V. Nguyen and Y. Choi, “Proactive Detection of DDoS Attacks Utilizing k-NN Classifier in an Anti-DDos Framework,” *World Acad. Sci. Eng. Technol. 39 2010*, p. 6, 2010.

[10] W. Zhou, W. Jia, S. Wen, Y. Xiang, and W. Zhou, “Detection and defense of application-layer DDoS attacks in backbone web traffic,” *Future Gener. Comput. Syst.*, vol. 38, pp. 36–46, Sep. 2014, doi: 10.1016/j.future.2013.08.002.

[11] S. Yadav and S. Subramanian, “Detection of Application Layer DDoS attack by feature learning using Stacked AutoEncoder,” in *2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT)*, Mar. 2016, pp. 361–366. doi: 10.1109/ICCTICT.2016.7514608.

[12] Y. Gu, K. Li, Z. Guo, and Y. Wang, “Semi-Supervised K-Means DDoS Detection Method Using Hybrid Feature Selection Algorithm,” *IEEE Access*, vol. 7, pp. 64351–64365, 2019, doi: 10.1109/ACCESS.2019.2917532.

[13] P. Xiao, W. Qu, H. Qi, and Z. Li, “Detecting DDoS attacks against data center with correlation analysis,” *Comput. Commun.*, vol. 67, pp. 66–74, Aug. 2015, doi: 10.1016/j.comcom.2015.06.012.

[14] R. Vijayasarathy, S. V. Raghavan, and B. Ravindran, “A system approach to network modeling for DDoS detection using a Naïve Bayesian classifier,” in *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, Jan. 2011, pp. 1–10. doi: 10.1109/COMSNETS.2011.5716474.

[15] W. Bao, N. Lianju, and K. Yue, “Integration of unsupervised and supervised machine learning algorithms for credit risk assessment,” *Expert Syst. Appl.*, vol. 128, pp. 301–315, Aug. 2019, doi: 10.1016/j.eswa.2019.02.033.

[16] “Quick Guide: Simulating a DDoS Attack in Your Own Lab,” p. 9.

[17] J. Fern and O, “Correlation Coefficient Definition,” *Investopedia*. https://www.investopedia.com/terms/c/correlationcoefficient.asp (accessed Aug. 15, 2021).

[18] J. Brownlee, “Ordinal and One-Hot Encodings for Categorical Data,” *Machine Learning Mastery*, Jun. 11, 2020. https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/ (accessed Aug. 15, 2021).

[19] K. Alsabti, S. Ranka, and V. Singh, “An Efficient K-Means Clustering Algorithm,” p. 6.

[20] “Determining The Optimal Number Of Clusters: 3 Must Know Methods,” *Datanovia*. https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/ (accessed Aug. 15, 2021).