Abstract

We aim to create a project that helps detect potential threats and attacks on a web application with the help of visualization techniques and machine learning. A thorough literature survey was conducted to find the best fit machine learning algorithm, most malicious threats, dataset and visualization techniques. Machine learning was practically implemented with the K-nearest neighbour (KNN) algorithm in USB-IDS-1 and CIC-IDS-17 datasets to compare results. It was done in 4 steps: preprocessing, statistics, feature selection and ML implementation. Visualization was done by plotting all datasets individually, with their benign and attack rows rendered on different graphs to avoid overlapping.

Table of Content

1. Abbreviations

ML	-	Machine Learning
DDoS	-	Distributed denial of service
KNN	-	K-nearest neighbour
IoT	-	Internet of Things
TCP	-	Transmission Control Protocol
IDS	-	Intrusion detection system

2. Introduction

2.1 What

We aim to create a project that helps detect potential threats and attacks on a web application with the help of visualization techniques and machine learning; thus protecting the website's security.

2.2 Why

Predicting potential attack attempts will help a website be more secure and respond to it immediately. Moreover, machine learning provides an advantage in detecting zero-day and DDoS attacks where signature based methods fail. The anomaly based approach provides another benefit - it scans general properties such as size, connection time, and number of packets. This is advantageous where an encrypted internet stream is involved. We also intend to study visualization of network traffic because it is effective in investigating anomalies when machine learning gives false positive alerts.

2.3 How

We did a thorough literature survey and built upon the research to find the best fit machine learning algorithm and dataset suitable for popular malwares like Hulk DDoS, TCP SYN flood, Slow HTTP attacks and Slowloris. We also investigated visualization tools and techniques in recent research papers to represent network traffic.

It would be beneficial to select the latest dataset for training the model. This dataset will provide us with the latest attack samples and also cover pitfalls of previous datasets. Thus we have selected University of Sannio's 2021 dataset: *USB-IDS-1*. As concluded in our literature survey, KNN is the most accurate algorithm for detecting anomalies, hence we implemented it in our project.

3. Literature Survey

We did a comprehensive study on the research already done in this field.

3.1 Machine Learning

We studied 10+ machine learning algorithms and 5+ datasets used to train intrusion detection systems. From Kostas' thesis [1], we also concluded that K-Nearest Neighbor(KNN) Algorithm provided best performance over multidimensional data and is a relatively fast algorithm during the training phase, with a tradeoff with execution time.

With the points we gathered in our literature survey, we decided to use KNN classifier.

K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. It stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.

3.2 Attacks and dataset analysis

In a research paper on using ML for IoT [2], we studied various kinds and effects of cyber attacks. These attacks were DoS, spoofing, jamming, man-in-the-middle, selective forwarding, malicious input and data tampering attacks. This helped us define the scope of our attack surface that we would be covering in our project.

Another research paper [3] suggested that the major challenges in the field of malicious network activity detection are a huge volume of network traffic and diversity due to new attacks.

Most of the malware can be detected by signature based detection, but not Distributed Denial of Service (DDoS) attacks. That is why we chose a ready-to-use and specially crafted modern dataset, USB-IDS-1, which focused on a range of DDoS attack types.

• **Hulk**: it is conceived as an HTTP flood, which spawns a large volume of obfuscated and unique requests. It generates numerous distinct requests in order to prevent the server from recognizing a pattern and filtering the attack. This makes the requests difficult to be detected by means of signatures. The main goal of the tool is to overwhelm a given web server with randomly generated header and URL parameter values. We use one of the most popular Hulk implementations for our experiments6.

• **TCPFlood**: it is another well-known DoS attack tool. The attacker sends TCP connection requests, locking the available ports on the server and causing incapability to accept TCP connections from legitimate clients. It can be considered a flooding attack. For our experiments we used a GitHub TCPFlood script7. It can launch a TCPFlood attack against the victim host in seconds.

• **Slowloris**: it is a tool that implements DoS attacks by sending slow HTTP requests (slow DoS attacks) against a victim server. This category of attacks uses low bandwidth approaches, which exploit a weakness in the management of TCP fragmentation of the HTTP protocol. We launched this attack by means of a well-known Python attack script. In particular, it implements a slow header attack by sending incomplete HTTP requests (i.e., without ever ending the header). If the server closes a malicious connection, this is re-established by keeping constant the total number of open connections.

• **Slowhttptest**: it is a tool that allows to launch slow DoS application-layer attacks. The tool can prolong HTTP connections in different ways. For our experiments we used Slowhttptest in the "Slowloris" mode, which allows to send incomplete HTTP requests to the victim server.

Each of these attacks have a feature which can help us identify it. This has been summarized in table-1, with the help of results obtained from feature selection in Kostes' thesis [1].

Attack	**Most Important Feature**	**Importance Weight**
HULK	Bwd Packet Length Std	0.514306
SlowHTTPTest	Flow IAT Mean	0.64206
DoS Slowloris	Flow IAT Mean	0.465561
TCPFlood	Bwd Packet Length Std	0.514306

Table 1: Most important feature of attacks in USB-IDS-1

Finally, the official paper on the novel dataset USB-IDS-1 [4] highlighted key advantages it offers from other publicly available datasets. Existing public datasets tend to neglect the multilayer perspective, which allows to establish whether the attacks they provide caused just marginal traffic fluctuations at the network-level or “real” damage to the victim applications.

USB-IDS-1 makes a step ahead of other common intrusion detection datasets. All the attacks in this dataset are proven to be effective against the victim. It was created with its use in training machine learning models and IDS kept in mind. Denial of Service (DoS) attacks from the widely-used CICIDS2017 dataset are negligible at application-level. USB-IDS-1 takes into account both network traffic and application-level facets.

3.3 Visualization of dataset

Keith Fligg and Genevieve Max[5] in their research paper talk about the psychology of visualization. The visualizations constructed allow important information to be more visible than less important ones and this concept is called pre-attentive. Shapes such as circles, squares, lines etc that make up an image are called objects. Pre-attentive objects are effortlessly distinguishable from other objects.

A detailed research paper by Z. Ruan [6] extensively described various types of multidimensional data visualization techniques, colour design for distinguishing classes, and dimensionality reduction methods. By weighing their properties and effectiveness against the project’s requirements, the paper concluded that out of scatter plot, hyper graph, force graph and parallel coordinate; scatter plot will yield the optimal result. It uses dots to represent values and is used to observe relationships between variables.

Thus we used a scatter plot on the USB-IDS-1 itself to observe any noticeable feature that could help us detect an anomaly in the network.

4. Detailed Design

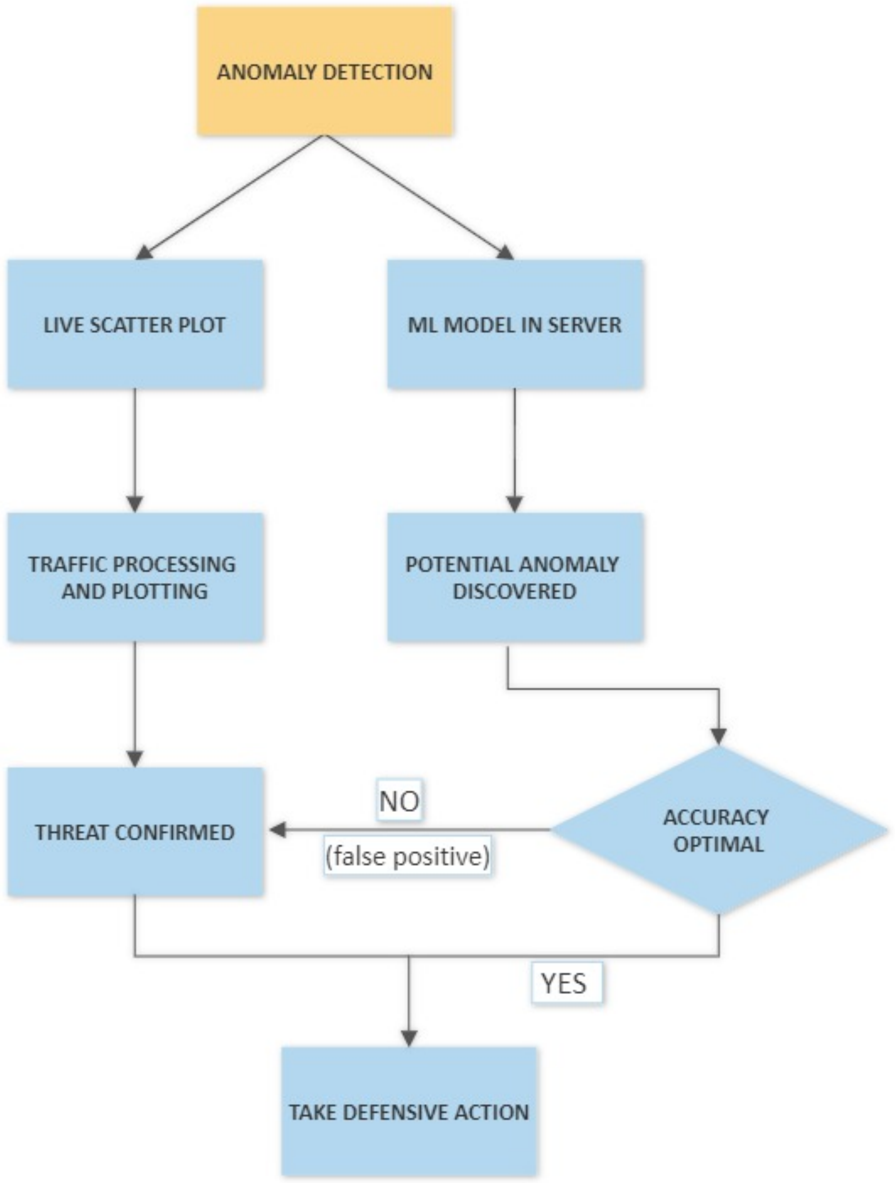


Figure-1: flow chart of potential project implementation

5. Implementation

The implementation was carried out in two parts: machine learning and visualization.

5.1 Machine Learning

This step was accomplished in 4 steps:

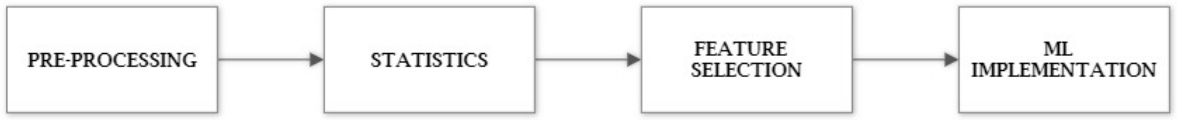
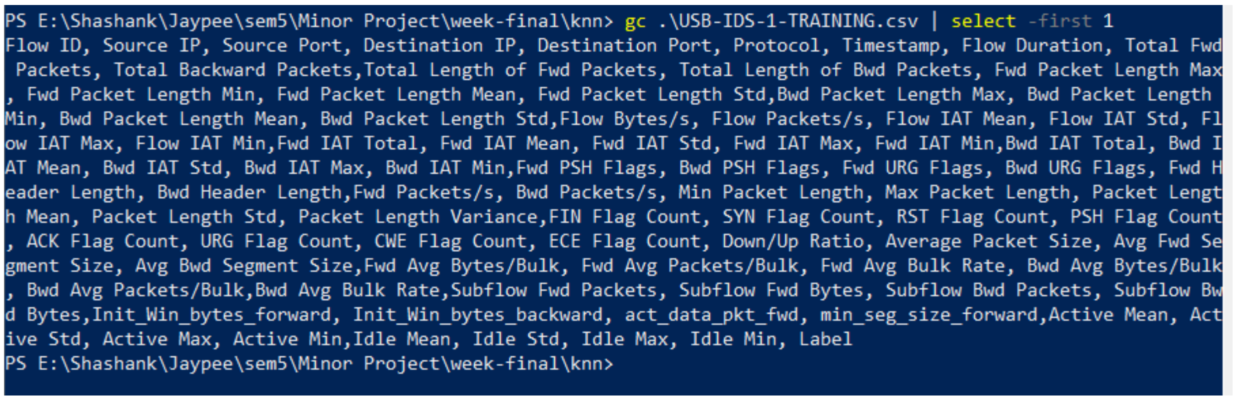


Figure-2: flow chart of machine learning implementation

1. Preprocessing: This step involves the cleaning of the dataset i.e removing missing values and converting the non - numerical values into numerical values. We were already provided with a machine-learning ready training and test dataset, so we needn't worry much about cleaning the dataset. In this step, we had to change the column names to remove trailing whitespaces.

2. Statistics: This program examines the file "USB-IDS-1-TRAINING.csv" and prints the statistics of attack and benign registry on this screen. It is not a prerequisite for any file. It only gives information.

3. Feature Selection: The aim of this program is to determine which features are important for each attack. For this purpose, Random Forest Regressor algorithm is used to calculate the importance weights of the features in the dataset. We will directly use the results from a project previously done [1]. These acquired features will be used in the next section.



```
PS E:\Shashank\Jaypee\sem5\Minor Project\week-final\knn> gc .\USB-IDS-1-TRAINING.csv | select -first 1
Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, Flow Duration, Total Fwd
 Packets, Total Backward Packets,Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max
, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std,Bwd Packet Length Max, Bwd Packet Length
Min, Bwd Packet Length Mean, Bwd Packet Length Std,Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Fl
ow IAT Max, Flow IAT Min,Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min,Bwd IAT Total, Bwd I
AT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min,Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd H
eader Length, Bwd Header Length,Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Lengt
h Mean, Packet Length Std, Packet Length Variance,FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count
, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Se
gment Size, Avg Bwd Segment Size,Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk
, Bwd Avg Packets/Bulk,Bwd Avg Bulk Rate,Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bw
d Bytes,Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward,Active Mean, Act
ive Std, Active Max, Active Min,Idle Mean, Idle Std, Idle Max, Idle Min, Label
PS E:\Shashank\Jaypee\sem5\Minor Project\week-final\knn>
```

Figure-3: List of columns in the dataset for feature selection

4. ML implementation: We have used the sklearn library to import the KNeighborsClassifier module. This is our main code. It trains the model with USB-IDS-1-TRAINING.csv dataset and prints the results of these operations on the screen and in the file "./results/results_Final.csv". It also creates a PDF file with the f-measure of our model. F-measure (F_1) is calculated as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

Where, TP = no. of true positives

FP = no. of false positives

FN = no. of false negatives

5.2 Visualization

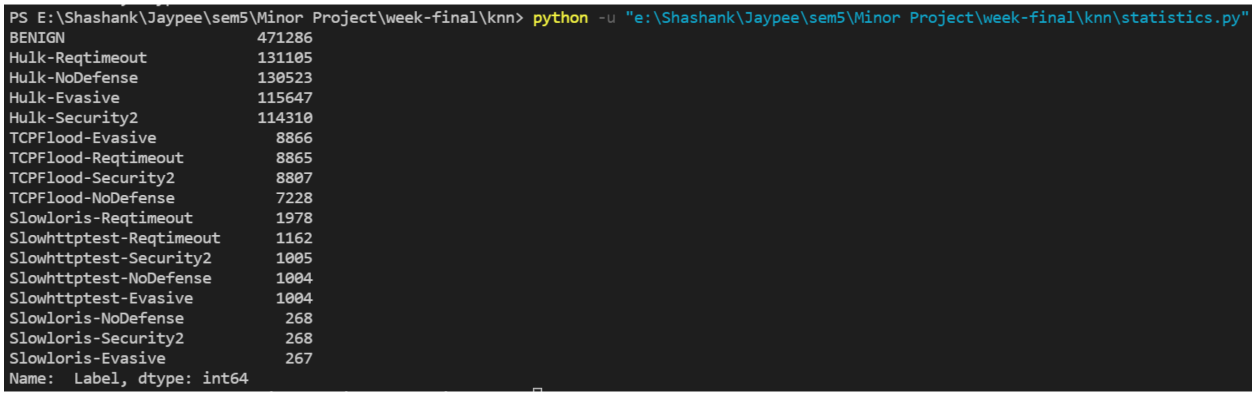
University of Sannio also provided separate datasets for each kind of DDoS attack. We use the following files for our purpose of visualization:

- Hulk-NoDefense.csv
- TCPFlood-NoDefense.csv
- Slowloris-NoDefense.csv
- Slowloris-NoDefense.csv

NoDefence provides the flows obtained by executing the attack with no defense in place. Thus we used it to obtain unbiased plots. We separated benign and the attack rows by filtering the label and mapped separate scatter plots for each attack. Although, in practise, both the points will appear in the same plot, we separated them here to aid us in studying their differences. Plotting was done with the aid of Microsoft Excel 2007.

6. Experimental Results and Analysis

These are the results obtained from dataset's statistics python code:



```
PS E:\Shashank\Jaypee\sem5\Minor Project\week-final\knn> python -u "e:\Shashank\Jaypee\sem5\Minor Project\week-final\knn\statistics.py"
BENIGN                     471286
Hulk-Reqtimeout            131105
Hulk-NoDefense             130523
Hulk-Evasive               115647
Hulk-Security2             114310
TCPFlood-Evasive             8866
TCPFlood-Reqtimeout          8865
TCPFlood-Security2           8807
TCPFlood-NoDefense           7228
Slowloris-Reqtimeout         1978
Slowhttptest-Reqtimeout      1162
Slowhttptest-Security2       1005
Slowhttptest-NoDefense       1004
Slowhttptest-Evasive         1004
Slowloris-NoDefense           268
Slowloris-Security2           268
Slowloris-Evasive             267
Name:  Label, dtype: int64
```

Figure-4: Attack type vs count in USB-IDS-1

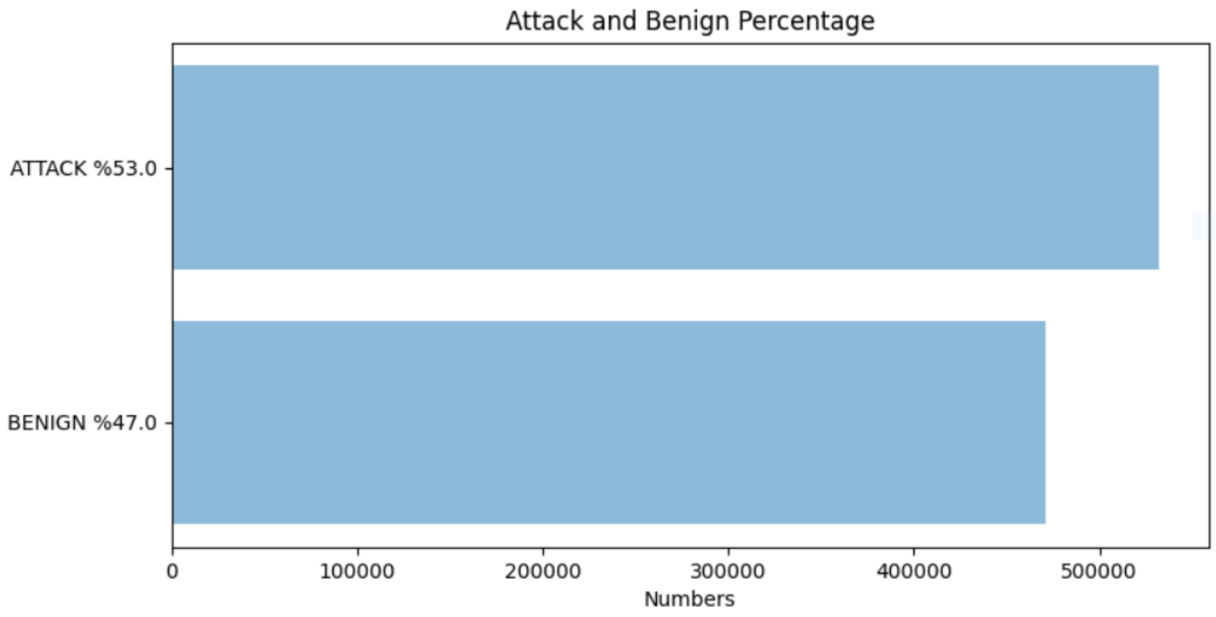


Figure-5: Total attack and benign percentage in USB-IDS-1

From Kostes' paper, the results of feature selection indicated that for KNN, the most suitable features are:

```
[" Bwd Packet Length Std", "Flow Bytes/s", "Total Length of Fwd Packets", " Fwd Packet Length Std", " Flow IAT Std", " Flow IAT Min", "Fwd IAT Total", " Label"]
```

After executing knn.py code on both the CIC-IDS-17 and USB-IDS-1 dataset, we get the following result:

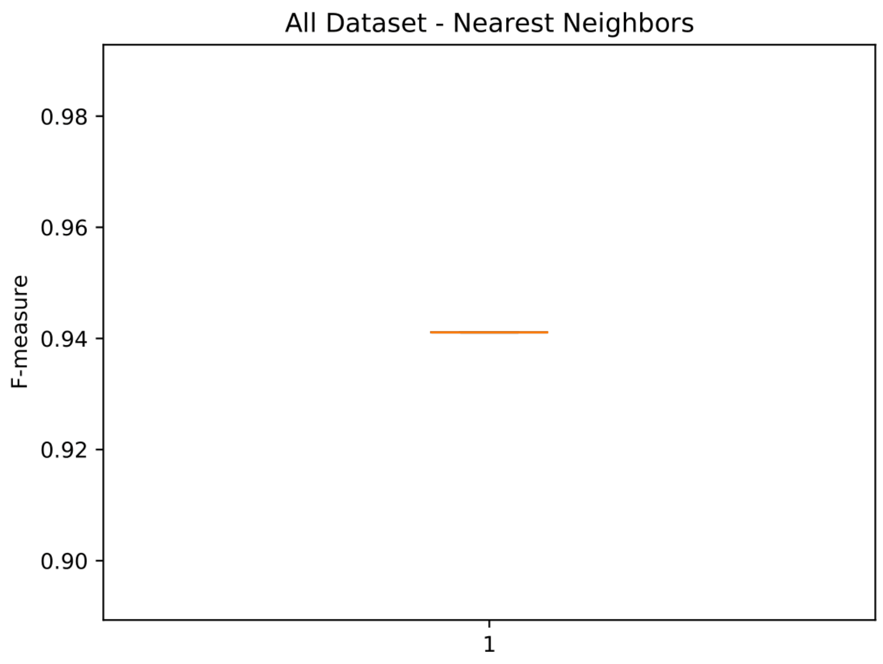


Figure-6: F-measure with USB-IDS-1 dataset.

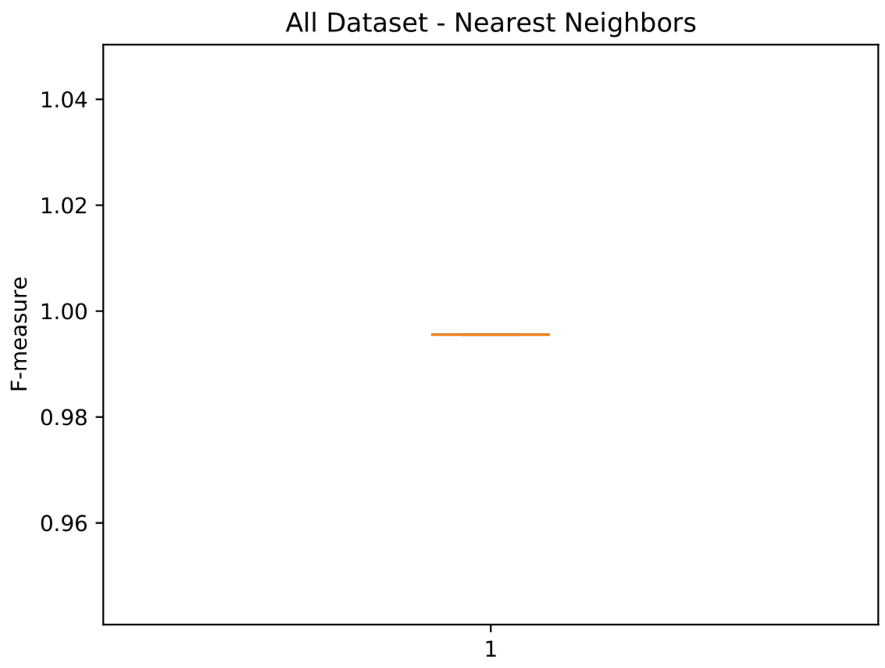


Figure-7: F-measure with USB-IDS-1 dataset.

File	**ML algo**	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Time (sec)**
USB-IDS-1	KNN	0.995579123	0.99554014	0.995586276	0.995563058	3660.66358
CIC-IDS-17	KNN	0.967628663	0.94750993	0.934962048	0.941088764	743.181707

Table-2: result from knn.py

With the novel 2021's USB-IDS-1 dataset, f-score of detecting a DDoS anomaly was 0.995563058, which is significantly better than the 0.941089 with CIC-IDS-17 dataset.

These are the results of scatter plotting the aforementioned datasets. In each of the following plots, orange dots represent benign traffic and blue dots represent malicious traffic. With the patterns of attack visually obvious, it is now easy to identify these future threats if they occur.

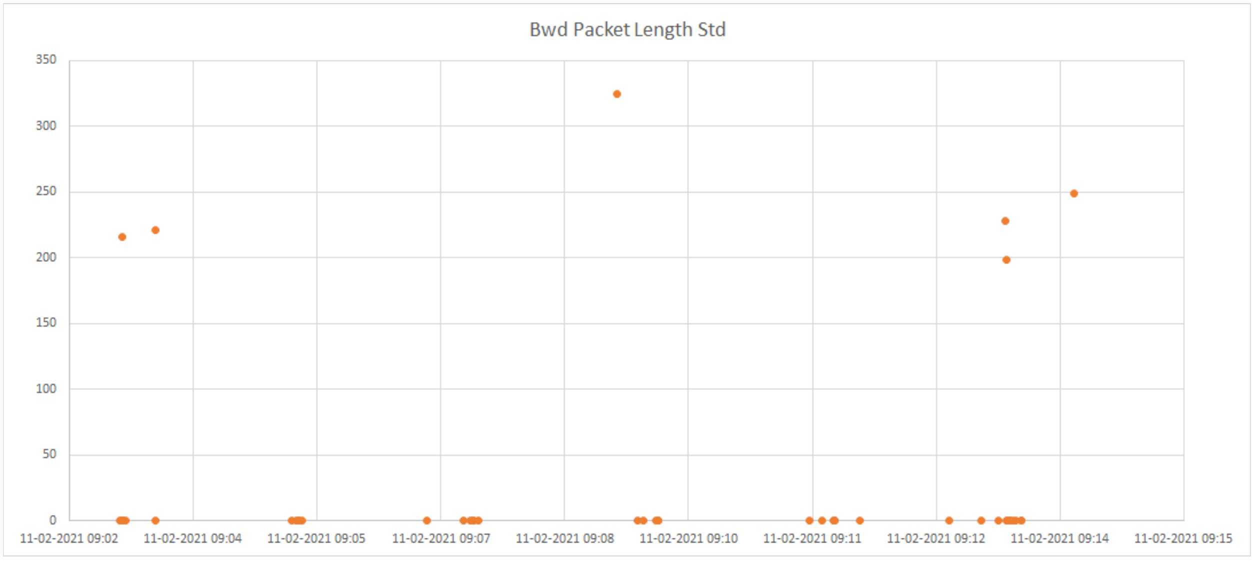


Figure-8: Hulk (benign)

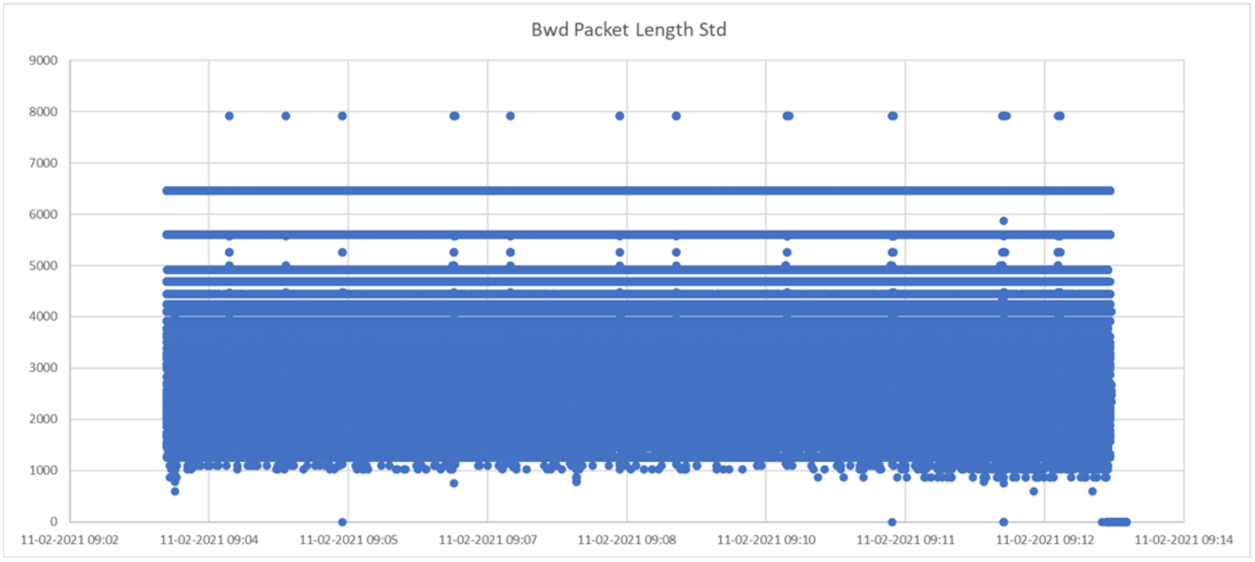


Figure-9: Hulk (attack)

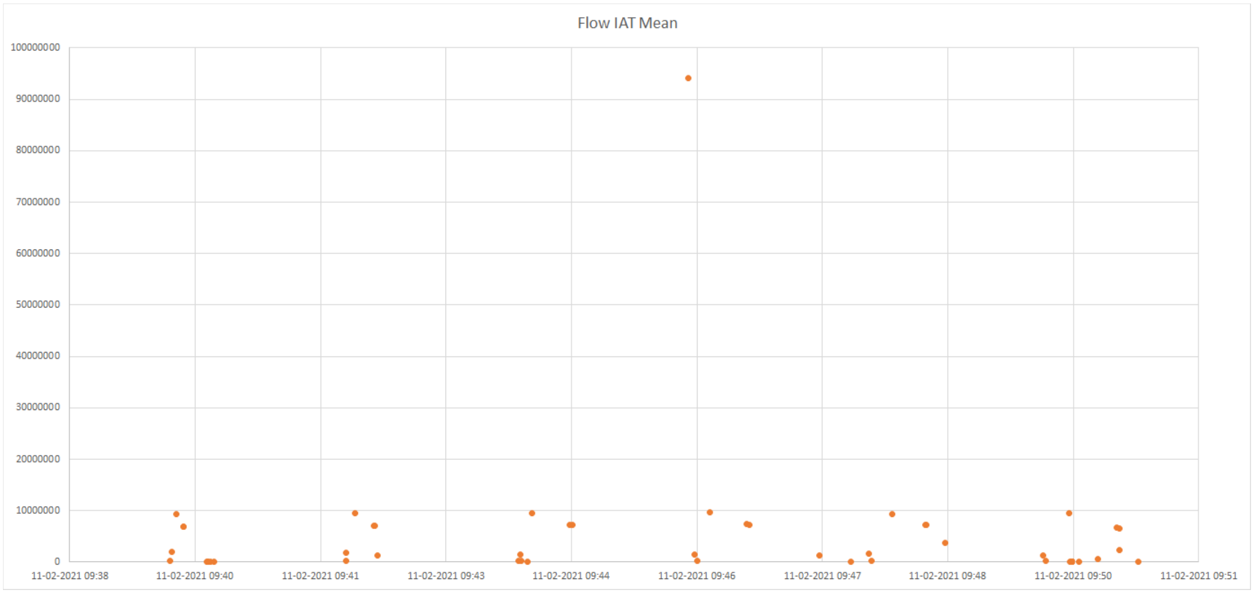


Figure-10: SlowHTTPtest (benign)

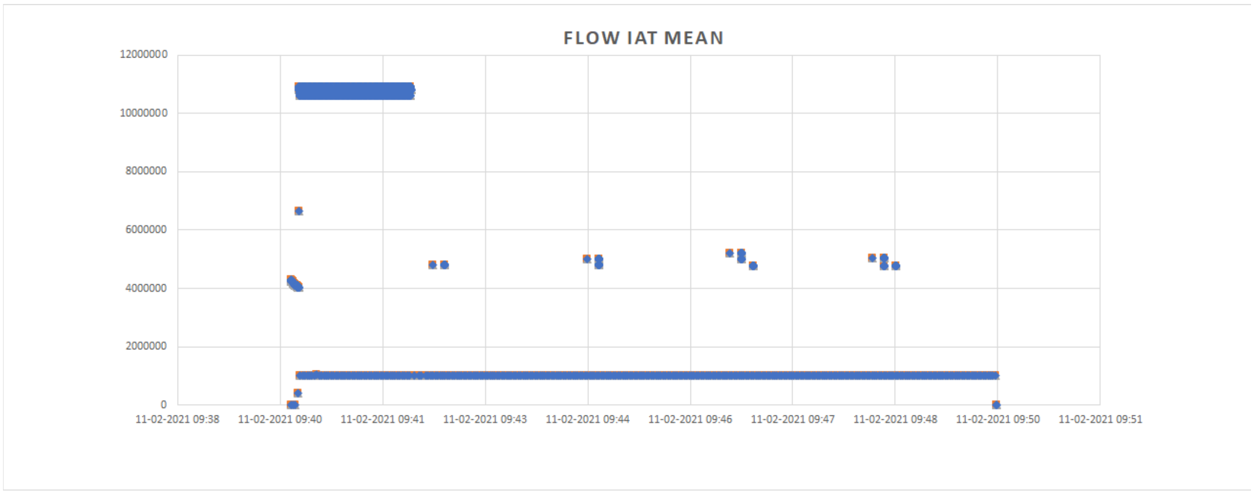


Figure-11: SlowHTTPtest (attack)

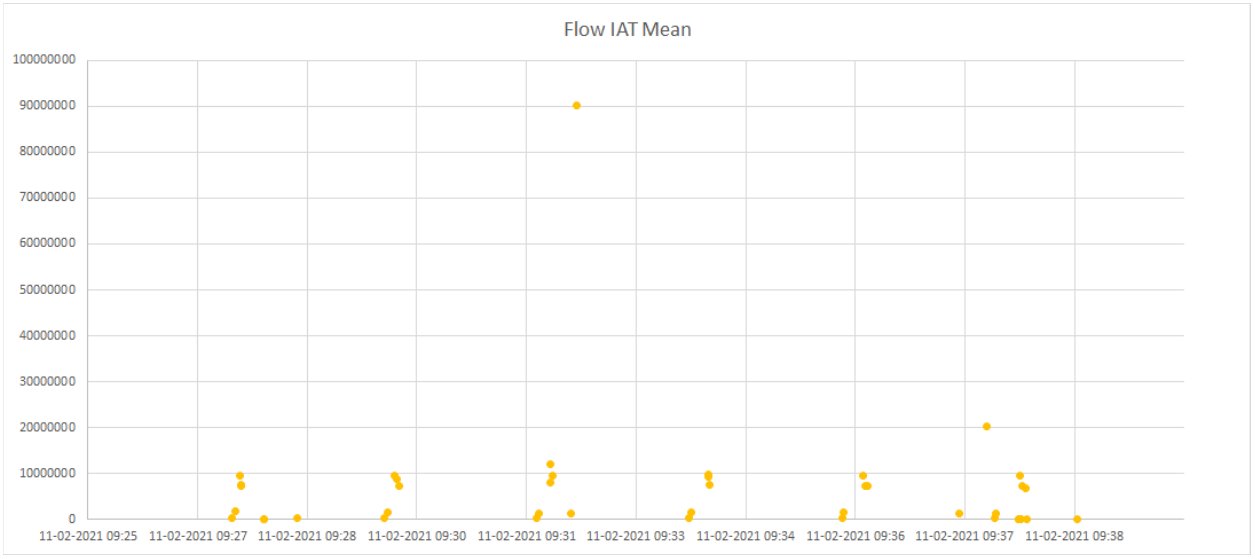


Figure-12: Slowloris (benign)

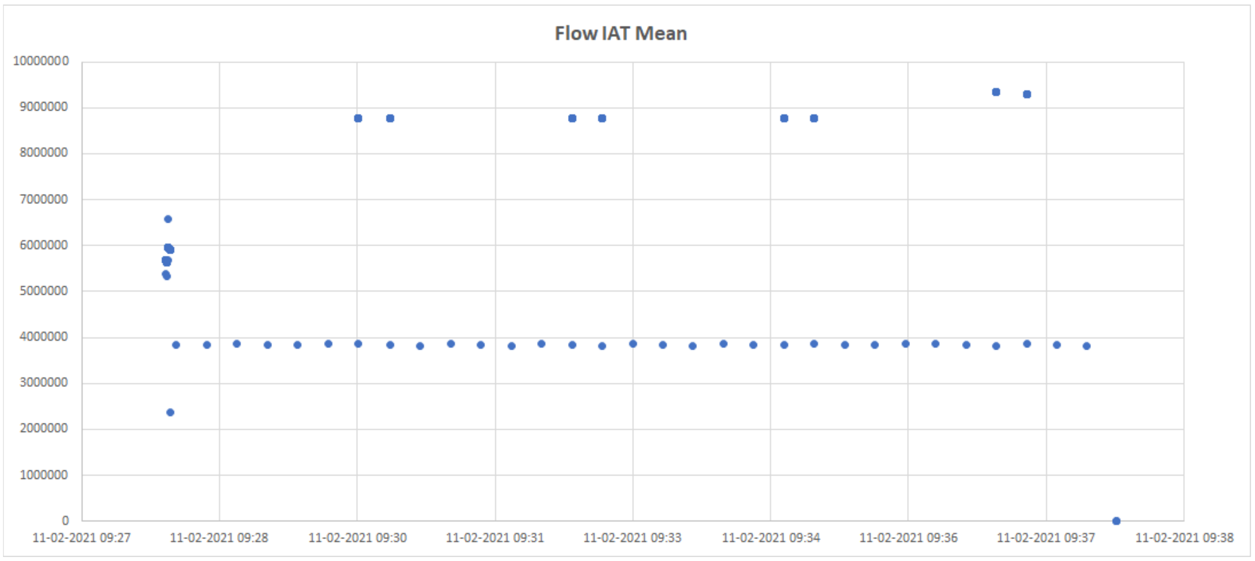


Figure-13: Slowloris (attack)

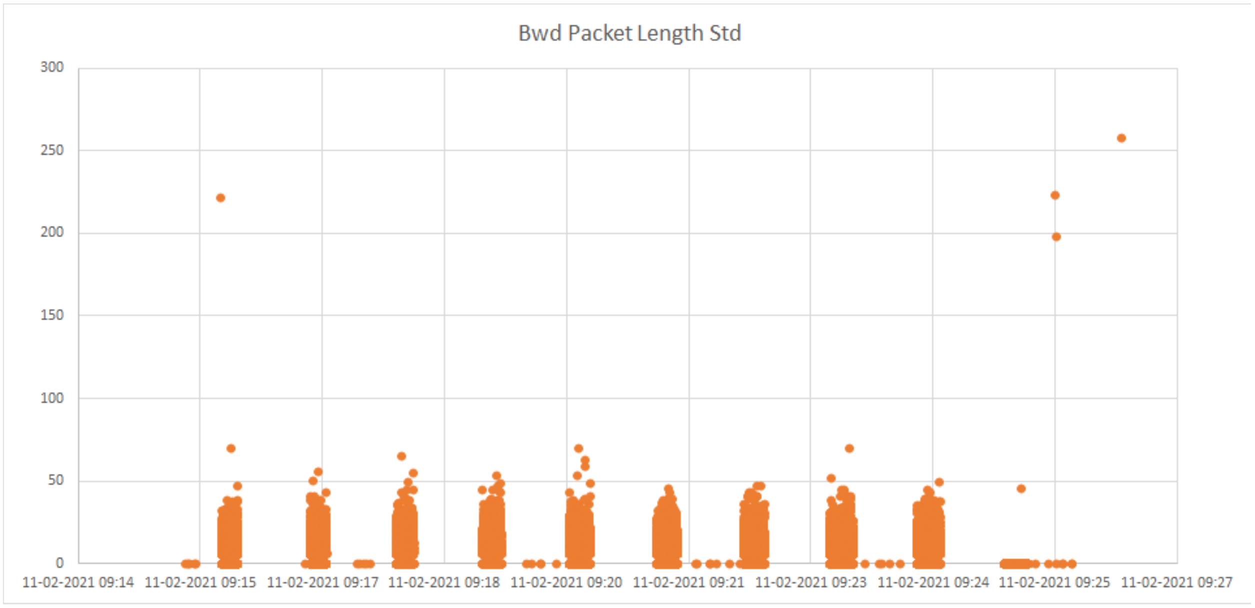


Figure-14: TCPflood (benign)

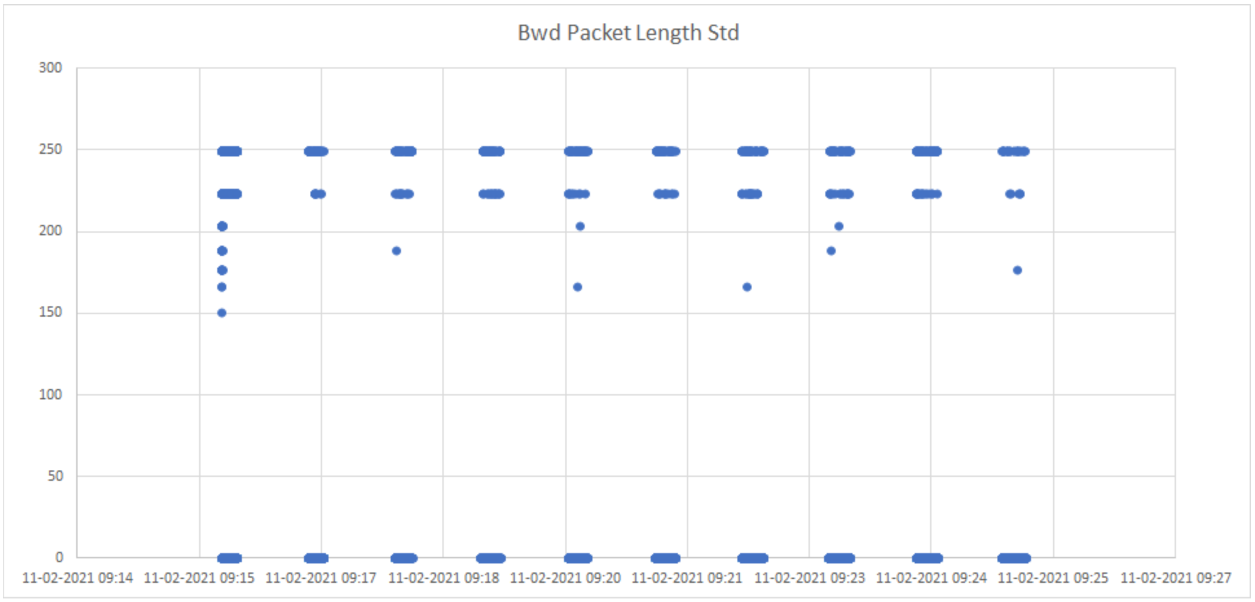


Figure-15: TCPflood (attack)

7. Conclusion and Future Scope

The experimental results were satisfactory. Both visualization and machine learning were successfully implemented. Moreover, USB-IDS-1, a dataset of the year 2021 was used to train the model, which contributed to a higher accuracy in detecting DDoS attacks. Not much research has been conducted around this new dataset. We've made the first step to practically implement the dataset for its intended purpose. Regarding visualization, there was a clear perceptible difference in benign and attack scatter plots of every attack. In case of the model's inaccuracy in detecting an anomaly (e.g. false positives and false negatives), the scatter plots will prove to be extremely useful.

The project can be expanded to be used as a scalable intrusion detection system. Moreover, the scope of attacks can be improved, as new types of zero-days are emerging every day. As more new datasets develop and old algorithms improve, we can continue enhancing the capabilities of the machine learning model.

8. References

[1] Kostas, Kahraman. (2018). Anomaly Detection in Networks Using Machine Learning.

[2] S. Tahsien, H. Karimipour, P. Spachos, “Machine learning based solutions for security of Internet of Things (IoT): A survey”, Journal of Network and Computer Applications, Volume 161, 2020, 102630, ISSN 1084-8045

[3] Madhankumar Y, “Malicious Attack Detection by Convolutional Deep Learning Model for Web Applications”, International Journal of Engineering Trends and Applications (IJETA) – Volume 7 Issue 4, Aug-Jul 2020

[4] Catillo, M., {Del Vecchio}, A., Ocone, L., Pecchia , A., & Villano, U. (n.d.). *USB-IDS-1: a Public Multilayer Dataset of Labeled Network Flows for IDS Evaluation*. Retrieved December 6, 2021, from http://people.ding.unisannio.it/villano/papers/DCDS2021.pdf.

[5] Fligg, Keith, and Genevieve Max. "Network security visualization." IEEE Network Special Issue on Recent Developments in Network Intrusion Detection (2012): 466-566.

[6] Ruan, Z., Miao, Y., Pan, L. et al. Big network traffic data visualization. Multimed Tools Appl 77, 11459–11487 (2018). https://doi.org/10.1007/s11042-017-5495-y