Detection of VPN Network Traffic

Avnish Goel
CSE Department, PES University
Bangalore, India
avnishgoel99@gmail.com

Apoorv Kashyap
CSE Department, PES University
Bangalore, India
apookash55@gmail.com

B. Devesha Reddy
CSE Department, PES University
Bangalore, India
devesharedddy10@gmail.com

Rochak Kaushik
CSE Department, PES University
Bangalore, India
rochakk17@gmail.com

S Nagasundari
ISFCR, CSE Dept, PES University
Bangalore, India
snagasundari5@gmail.com

Prasad B Honnavali
ISFCR, CSE Dept, PES University
Bangalore, India
prasad.honnavalli@gmail.com

Abstract*—Today, in the world of the internet, the amount of internet activity is growing which generates a huge amount of data. Data has become the most powerful asset. There is a tremendous increase in the need for security and privacy for each and every personnel. There is a need for more cybersecurity and privacy preservation. One of the ways to prevent loss of data and breach in privacy is by using a Virtual Private Network. A Virtual Private Network(VPN) sends the data traffic through an encrypted 'tunnel', making it extremely difficult to decipher or intercept. Because of the inadequate security, public networks are ideal ground for hackers. Cybercriminals and internet service providers can eavesdrop such networks, which may result in theft of personal/financial data. When a VPN is not used on public networks, hackers may be able to steal personal data such as credit card information or passwords. When an individual is connected to a VPN, the original identity of the user is hidden. Though VPN is used to tunnel the traffic inorder to prevent data breach, it may also become a source for several attacks such as Ransomware. Most of the time, the perpetrator makes use of a VPN to remain anonymous during the attack. As a result, there is a need for VPN detection to prevent such malicious activities. This paper focuses on building a VPN based on a popular network security protocol and detecting VPN traffic using various Machine Learning models. The results are analyzed and accuracy for VPN detection seems to be better than the existing approaches.

Keywords*—VPN, IPSec, MLP, Random Forest, AWS, Real-Time VPN Detection

I. Introduction

The world has seen an immense amount of technological advancements in the past two decades which in turn is generating a need for security as well as safety and privacy. The Internet has become everything for most businesses online. If a person uses a public WiFi network, such as those which are found in airports and cafes, anyone connected to the same network can access the other users' private data flowing through the network. Therefore a Virtual Private Network(VPN) service is needed. It forms a private tunnel through which the data flows across the internet, thus ensuring that all of the communication over the internet remains hidden.

Nevertheless, there have been occasions when the use of VPN has proven to be a burden for organizations. On university networks, there is a growing amount of VPN usage. Most of the time, students use a VPN to bypass a web filter or a firewall to access a blocked service such as a torrenting website. A VPN can break through a firewall, opening the door for malware attacks like ransomware.

VPNs have existed for a long time. A VPN integrates a private network to a public network, enabling users to transmit and receive data as though their computers were directly connected to the private network.

The benefit of using a VPN is to ensure end-to-end encryption between the client and the server (even in public networks). Basically a VPN secures the network and hence stops any third-party entity from stealing the user's sensitive data. The benefit of VPN detection systems is to prevent people from accessing restricted websites or applications.

This paper focuses on exploring various VPN security protocols based on performance factors taking insights from "Application specific tunneling protocol selection for Virtual Private Networks"[1]. It uses the most suitable protocol to demonstrate the working of the same. It further emphasizes on detecting and classifying the network traffic as VPN/Non-VPN over the standardized Dataset using various Machine Learning algorithms. This paper compares and tests the different ML models such as the multilayer perceptron neural network model, random forest model based on their accuracy and performance. Further, a website has been hosted where the detection and classification of VPN/ non-VPN is done using the above models and the clients are directed to the respective webpage according to the prediction made by the models.

The content of the paper is arranged as follows: section II describes related research; section III describes the proposed methodology along with graphs; section IV shows the experimental results and section V is the conclusion.

II. Related Works

Research on different literatures regarding the impact of VPN on network performance have been conducted. Inferences drawn from literature are depicted as follows.

"Virtual Private Networks' Impact on Network Performance"[2] implies that the VPN has an impact on the throughput and packet delay, hindering performance of the network. The impact can be reduced with the use of small packet size and window size.

2022 IEEE Delhi Section Conference (DELCON) | 978-1-6654-5883-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/DELCON54057.2022.9753621

Different Network Topologies influence the VPN performance by affecting the factors such as throughput, network latency, jitter, and packet loss. The analysis of different network topologies have shown that a star or tree topology with not more than 3 layers is most preferable.[3]

The comparisons between the impact of different Layer 2 protocols on the network performance have been computed. Results from the comparisons suggest that OpenVPN provides the highest amount of security.[4]

From the paper [5], it can be inferred that IPSec VPN protocol is secure as it supports various encryption algorithms such as advanced encryption standard (AES), data encryption standard (DES), triple data encryption standard (3DES) and the use of public/private keys. The paper describes the use of various cryptographic encryption techniques like Rivest–Shamir–Adleman (RSA), DES, 3DES, AES, Blowfish etc. These techniques increase the computational time, memory usage and output bytes. The security protocol IPSec provides integrity, security and authentication using security services, 'Authentication Header' protocol and the 'Encapsulating Security Payload', thus encrypting and encapsulating the packets being transferred.

The objective of Case Study at the University of Namibia [6] is to design and implement a secured, reliable and affordable IPSec point-to-point VPN wireless area network. In comparison to Multiprotocol Label Switching(MPLS), which is handled by a third party, such as the internet service provider, the deployment of an IPSec-based VPN at University of Namibia substantially reduced the expenses. IPSec is a highly secure, cost effective VPN protocol which can be internally managed at a local level.

The paper describes how bandwidth-sensitive and time-sensitive software packages such as Voice over ip, video conferencing, and streaming which require high throughput and low latency. On the other hand applications such as file transfer, email, and web documents are not time-sensitive and have flexible throughput requirements. As a result, the IPSec VPN protocol is better suited to applications that require security. Layer Two Tunneling Protocol(L2TP) with IPSec is preferred over Point-to-Point Tunneling Protocol(PPTP) while using a remote access VPN. L2TP with IPSec is also ideal for bandwidth-constrained, time-constrained, and security-constrained applications.[7]

In the paper [8], the influence of time-related features are considered for the characterization of encrypted network traffic and detection of VPN network traffic. For this purpose of classification, common machine learning algorithms such as C4.5 and k-nearest-neighbors are used. It was also found that the classifiers show better results when shorter timeout values are used for generating the flows.

The paper[9] has analyzed, processed and combined datasets from various sources implemented and compared encrypted malicious traffic detection algorithms while paper [10] describes machine learning algorithm for malicious network traffic detection using only the bytes of the raw network traffic using a 1D-Convolutional Neural Network (1D-CNN) and Feed Forward Network.

III. Methodology

A. VPN set-up and Establishing connection

1) VPN Server :

A VPN Server has been set-up on the AWS cloud platform. This server can be used by a variety of user groups ranging from laptop, desktop or smartphone users to offices and home offices. The users can connect to this VPN service using the network devices like switches and routers. On the AWS platform, an EC2 Linux instance is created for setting up the server. Security and Authenticity is provided by the rules used to create the AWS instance. A VPN internet gateway is also created. This whole architecture sits under a Virtual Private Cloud(VPC) that provides a public IP/endpoint for it to be accessed through the internet. The user is provided with a pre-shared key(PSK), username and password. These credentials are essential to connect to the VPN service. After connecting to the VPN, the user can access the internet by hiding his original identity(IP, location) and device from the public Internet. The data which flows through this network is encrypted. Anyone trying to sniff the data being sent on the internet would only get the VPN server's location and IP. Hence the users' privacy is preserved.

The server that has been built currently supports only a single user to connect to a VPN using the IPSec VPN protocol. This server is successful in hiding the user's IP address from other organizations on the Internet.

2) Connecting to the VPN Instance :

The user must locate the VPN settings on the device in order to connect to the VPN server. The VPN server's IP address, username, password and pre-shared key must be correctly provided in the appropriate areas. For example in an Android device the user needs to navigate to the VPN profiles in the settings and create a new VPN profile by entering the correct credentials.

After entering all these credentials, the user successfully gets connected to the VPN. After a connection is established, the user can ensure that the VPN is working successfully by finding the device's public IP address. This can be found out by accessing the website www.whatismyip.com. In this way the user can confirm that his original IP address is hidden and the data that he is accessing on the Internet is being tunneled through the VPN service running in the Elastic Cloud Compute(EC2) instance.

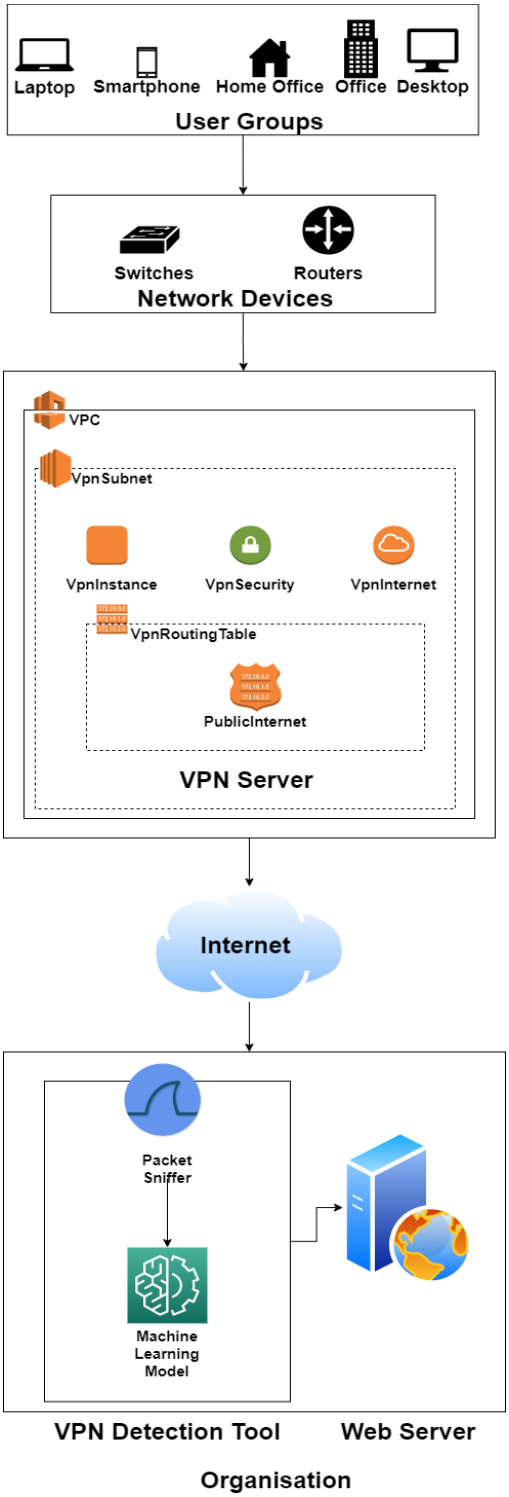

Fig. 1. Architectural Diagram

In fig. 1, the architecture of the project is depicted which consists of different components such as user groups, VPN server and the organization server. The user groups include laptops, smartphones, desktops to offices and home offices. They can connect to the VPN server through network devices like switches and routers. The VPN server is hosted on Amazon Web Services using the VPC which provides a public IP address to the users who connect to the server. A routing table is created so that the users can access the internet through this VPN server.

At the organization server side, a VPN detection tool is deployed. This tool consists of a packet sniffing tool, like wireshark or tshark, which captures all the incoming connection packets and sends them to the machine learning model, in appropriate format. The machine learning model further works on those packets to identify if the users are using the VPN or not. The VPN users are not allowed to access the organization content while the non-VPN users are redirected to the organization's main webpage.

B. Dataset

The dataset used in the project is a standardized dataset that has been taken from the official website of "Canadian Institute for Cybersecurity". The name of the dataset is "VPN-nonVPN Dataset". The dataset is rich in quality and diversity. It consists of the traffic that has been captured with and without using VPN. There are 14 different traffic categories like P2P, VoIP etc. The traffic generated is based on the different activities by the user such as browsing, Email, Chat Application, Streaming, File transfer etc.

The Wireshark and tcpdump are used to capture the traffic while creating the dataset. The total size of the dataset is around 28 GB. The network traffic in the dataset is labelled, it also includes full packet capture in the pcap format.

C. Data Preparation

Different methods have been considered for the purpose of converting the dataset obtained in the PCAP format to the usable CSV format. All of them have their own benefits as well as drawbacks. The methods are as follows:

1) Cisco Joy :

Cisco Joy is a software package that uses the libpcap library to extract data characteristics. It works on live internet traffic or saved pcap files. These data characteristics are represented in JavaScript Object Notation(JSON) format. It also includes data analysis tools that may be used to derive appropriate conclusions from the data. Deep analysis of packets can be done with this method. It can be used for exploring and analyzing threat relevant data.

It has few of the limitations such as it only considers the first 200 unique packets transferred. The whole process of data analysis and conversion is very time consuming. It considers both bidirectional and unidirectional flow of packets.

This method has not been adopted as only the first 200 unique packet captures of the pcap files are converted into JSON format. Moreover it is inefficient in terms of time taken to convert those packet captures.

2) Wireshark :

Wireshark is a packet analyzer used to capture packets flowing through the networks. It can convert all the packets directly to json, csv, xml, plain text formats.

Wireshark has few drawbacks such as the converted csv file does not contain all the required attributes and the converted JSON file contains the payload which takes up a lot of memory.

This method had been discarded because each pcap file had to be manually converted. Hence it was an inefficient method. It also requires to merge all the files into a single file at the end. It cannot be used to convert files simultaneously while capturing the traffic.

3) Nfstream :

NFStream is a Python framework for working with live network traffic that provides fast, versatile, and expressive data structures. NFStream also aids in the reproducibility and deployment of machine learning techniques for network traffic control. It converts all the packets to csv format efficiently in terms of both time and size. It only considers bidirectional packet flow. It also analyses the data statistically of different attributes(min, max, mean, stdv)

Therefore, Nfstream is selected for conversion of Dataset from PCAP to CSV. Also it is suitable for conversion of live network traffic as well.

D. Preprocessing

Nfstream is used to convert all the packets in PCAP format to CSV format efficiently in terms of both time and size as it only considers bidirectional flow of packets. It also provides the analyses of the data statistically in terms of min, max, mean and standard deviation of different attributes.

All the CSV files are merged together using pandas. The final column VPN which is binary classification of VPN and Non-VPN is added which signifies whether the VPN was used or not. 0 was used for Non-VPN connections and 1 was used for vpn connections.

The converted CSV dataset consisted of 383889 rows and 86 columns/attributes. Few of the attributes such as client_fingerprint, server_fingerprint, user_agent, content_type columns are mostly empty so they are dropped. Any other null values present in the rest of the attributes are handled in a similar way. Pearson's correlation coefficient of all the attributes are calculated with respect to the result column i.e. 'vpn'. Attributes having correlation coefficient { c } such that

$$c < -0.1 \text{ and } c > 0.1 \quad -(1)$$

Rows	383889
Columns	86

are selected to further process the detection of VPN. After calculating the Pearson's correlation coefficient, graphical analysis is performed on the selected attributes. The physical significance of the attributes is also considered for determining selection of attributes.

Based on the above selection criteria, 18 attributes, represented in the table I, along with their Correlation Coefficients, are selected out of 86 attributes for the process of classifying the network traffic into VPN/Non-VPN.

TABLE I CORRELATION COEFFICIENTS OF SELECTED ATTRIBUTES

Attributes	**Correlation Coefficient**
bidirectional_mean_piat_ms	-0.011164281
src2dst_bytes	-0.000410163
src2dst_mean_piat_ms	0.019052126
src2dst_duration_ms	0.031180924
bidirectional_duration_ms	0.031345471
dst2src_duration_ms	0.069250159
dst2src_stddev_piat_ms	0.08529663
dst2src_rst_packets	0.101003051
src2dst_fin_packets	0.101278065
src2dst_syn_packets	0.120710174
bidirectional_fin_packets	0.121087804
dst2src_syn_packets	0.125937516
dst2src_fin_packets	0.130759379
bidirectional_mean_ps	0.174280965
bidirectional_stddev_ps	0.200517862
dst2src_mean_ps	0.44087781

E. Data Visualization

Graphs were used in order to visualize the data related to the various attributes of VPN and Non-VPN traffic.

Fig 2 and 3 depicts the proportion of bytes transmitted from source to destination with and without the use of VPN. From fig. 2, it can be inferred that the 65 bytes transferred has the highest percentage when using the VPN. Whereas, fig. 3 shows that 128 and 168 bytes transferred that are greater than double of the former, have the highest percentage while not using the VPN.

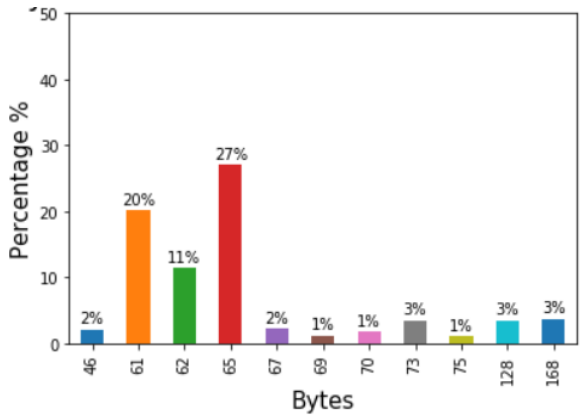

Fig. 2. Bytes transferred from source to destination with VPN

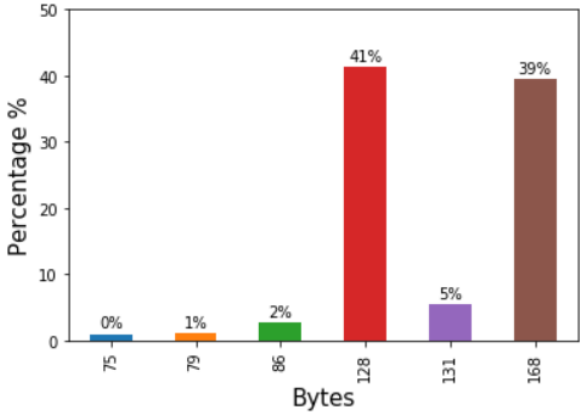

Fig. 3. Bytes transferred from source to destination without VPN

In fig. 4 and fig. 5, the graphs show the frequency of the mean packet size sent from source to destination with and without using the VPN. It can be inferred from fig. 4 that the packets having a mean size of 60 and 65 have the highest frequency while using the VPN. Whereas, fig. 5 shows that the packets having a mean size of 64 and 84, which is exactly half of the bytes transferred, have the highest frequency while not using the VPN.

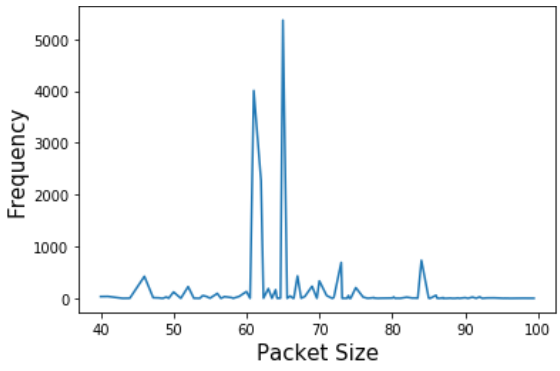

Fig. 4. Source to destination mean packet size with VPN

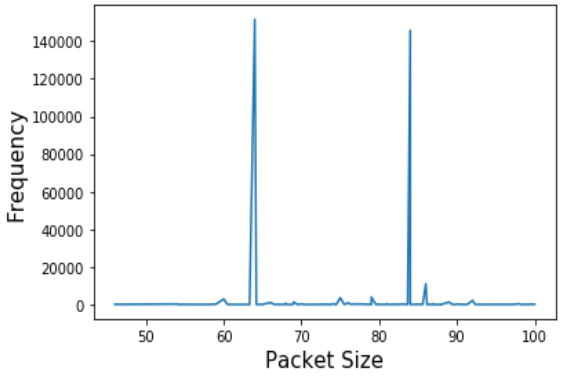

Fig. 5. Source to destination means packet without VPN

The fig. 6 and fig. 7 show the frequency of the mean packet size sent from destination to source with and without using the VPN. It can be inferred that the graphs show a similar trend in both the cases since there is no VPN present at the destination.

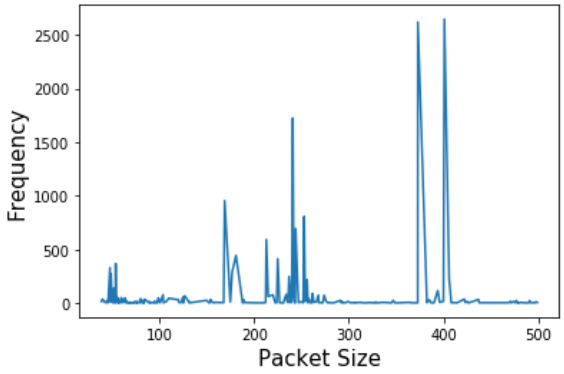

Fig. 6. Destination to source mean packet size with VPN

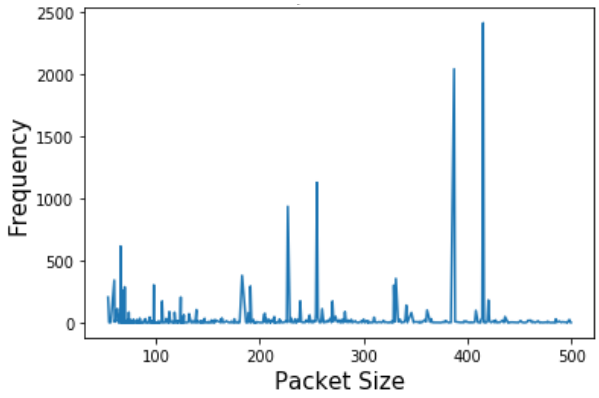

Fig. 7. Destination to source mean packet size without VPN

In fig. 8 and fig. 9, the graphs show the response time, number of packets and their time taken, with and without using the VPN. It can be inferred from the graphs that while using the VPN, a large number of packets took more time during the response .On the other hand, nearly all the packets have a quick response time while not using the VPN.

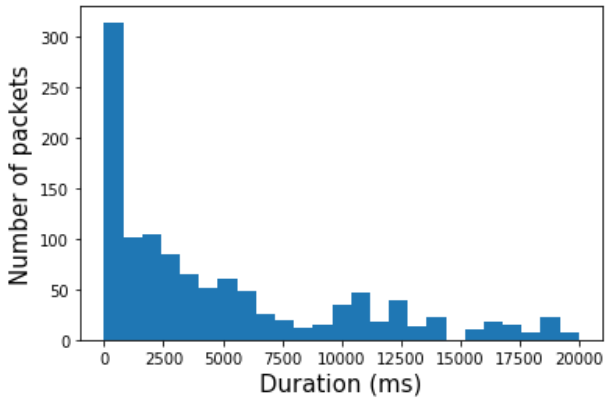

Fig. 8. Response Time with VPN

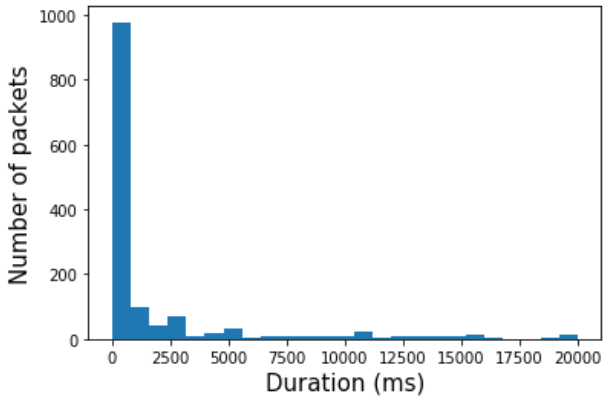

Fig. 9.Response time without VPN

F. Multilayer Perceptron (MLP) model :

Multilayer Perceptrons (MLPs) are a type of feedforward Artificial Neural Network (ANN) that consists of multiple layers of perceptrons (with threshold activation). There are at least three levels of nodes in an MLP: an input layer, a hidden layer, and an output layer. Each node represents a neuron that activates in a nonlinear manner. Backpropagation is a supervised learning technique used by MLP during training.

Neural networks are adaptable and may be used to solve issues in both regression and classification. As a neural network is a mathematical model with approximation functions, any numeric data can be utilized in the model. The predictions are rather quick once the model is trained. With more data points, neural networks function best.

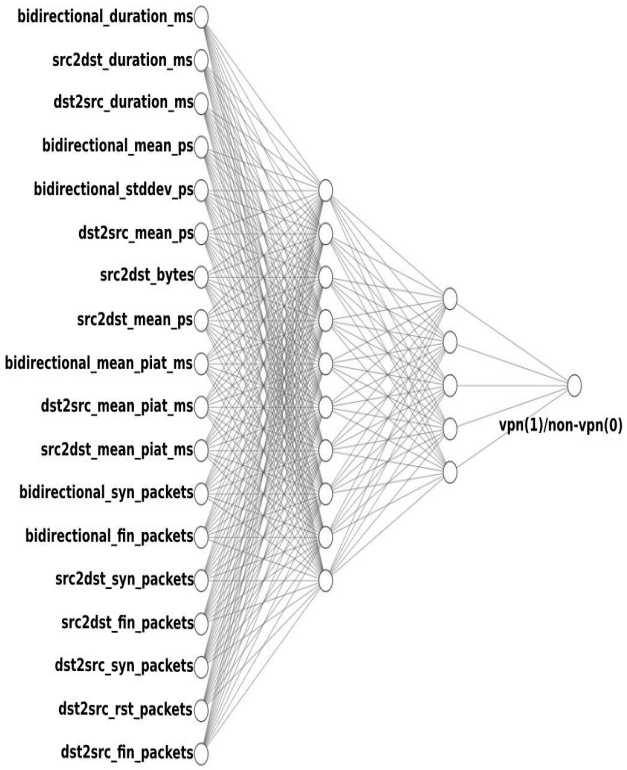

Fig. 10. Visual Representation of Neural Network

There are few drawbacks such as the influence of each independent variable on the dependent variables cannot be known. Training an MPL with conventional CPUs is computationally expensive and time intensive. Training data is extremely important for neural networks. Overfitting and generalization become an issue as a result of this. The model is more reliant on the training data and may be tailored to it.

G. Random Forest Model :

Random forests is a classification and regression ensemble learning approach. It entails building a large number of decision trees when the model is being trained.

For classification tasks, the random forest's output is the class chosen by the majority of trees. The mean or average forecast of the individual trees is returned for regression tasks. It employs a modified tree learning algorithm that picks a random subset of characteristics at each candidate split throughout the learning process.

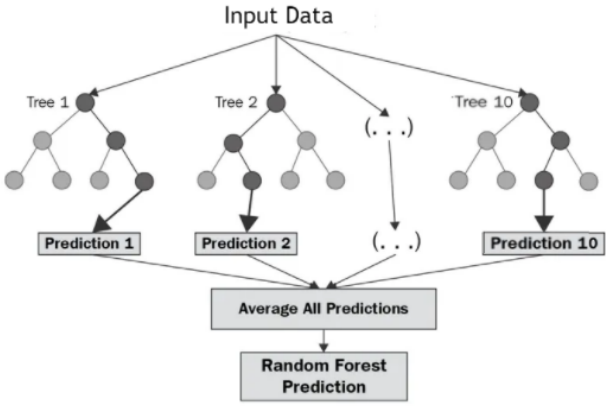

Fig. 11. Visual Representation of Random Forest

Some of the advantages of using a random forest model are that the ultimate conclusion of a random forest is dependent on all of the trees, as it only analyses a subset of characteristics. As a result, there is a greater degree of generality and less overfitting. It guarantees that individual tree faults are minimized, as well as overall variation and error. It can help deal with mistakes in data that is unbalanced (one class is majority and other class is minority)

Random forest also has some drawbacks such as the features need to have some predictive power otherwise they won't work. Predictions of the trees need to be uncorrelated. The results and performance may be affected by different parameters and random seeds.

H. Live Data Analysis :

An EC2 instance has been set up on the AWS with the following configuration:

Putty SSH client is used to connect to the EC2 instance in order to install/configure the apache/web server on the virtualized Ubuntu instance. The Apache server is configured on the instance using CLI with appropriate settings. Websites for restricted VPN users and genuine Non-VPN users are created and hosted using the apache2 server.

Tshark is a network analyzer. It has been used to capture the live network data packets over the eth0 interface. The packets captured are stored in the pcap file which is then used by the NFStreamer for analyzing and classification of the live data packets captured.

OS	Ubuntu 20.04 LTS
Instance Type	t2.medium
Inbound Rules	HTTP, HTTPS and SSH

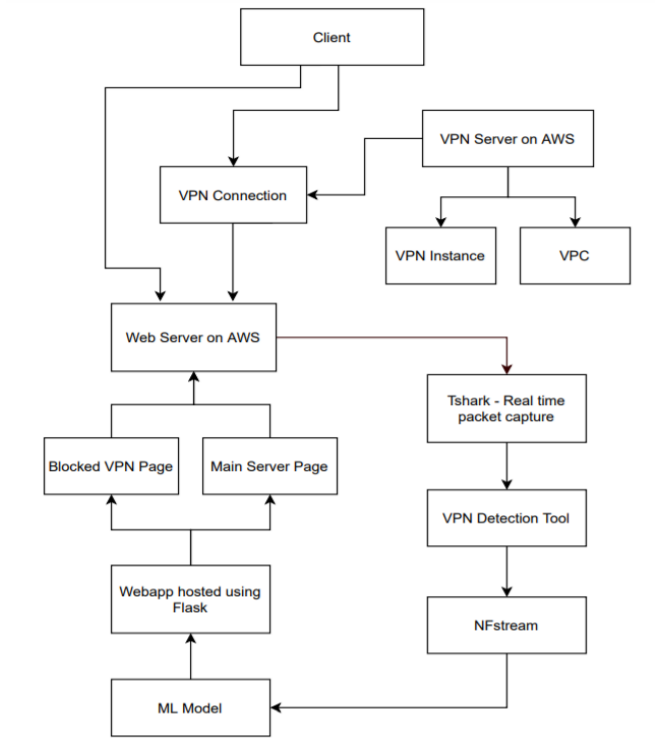

Fig. 12. Live Data Analysis Workflow

The validation of the model is done by using a pickle module in python to save the standard scalar model and trained model as an object so that it can be loaded in the server machine. Then the live data capture is loaded in the model to check for validation.

According to the results from the prediction made by the above used models, the client is redirected to the appropriate webpage. If the model detects the use of VPN, the client is restricted to access the main webpage and the following message is displayed to that client: "Forbidden you seem to be connected to a VPN. Disconnect the VPN and refresh the page to access the website". If the model does not detect the use of VPN then the client is redirected to the main server web page.

IV. Results

A. Multilayer Perceptron (MLP) Model

Multilayer perceptron model has been tested for the classification of vpn and non-vpn network traffic. The dataset has been divided into a test-train split of 20:80.

Fig. 13 and fig. 14 represent the confusion matrix for the testing set and training set respectively. The details such as number of False positives, False negatives, True positives and True negatives are represented in the confusion matrix. This is a measure of calculating the amount of correctly and incorrectly classified instances. The model has to be trained in such a way that the number of False positives and True negatives are minimized.

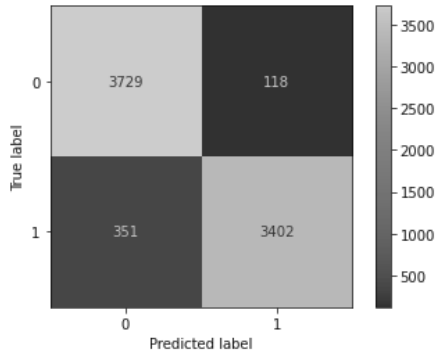

Fig. 13. Confusion Matrix of Test set.

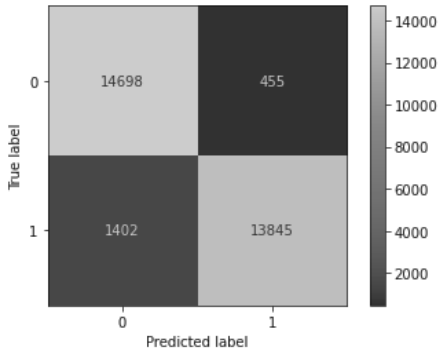

Fig. 14. Confusion Matrix of Training set

The model gave an accuracy of 93.83% for the test set. Hence 7131 were correctly classified out of a testing set of 7600. In other words, out of 3847 Non-VPN instances, 3729 had been correctly classified and out of 3753 VPN instances, 3402 had been correctly classified. The precision and recall for the same were 0.96647 and 0.90647 respectively. The F1-score was observed to be 0.93551. The same has been displayed in Table II.

TABLE II Test Set results

Accuracy	93.83%
Precision	96.64%
Recall	90.64%
F1 Score	93.55%
Average True Positive Rate	96.93%
Average True Negative Rate	90.64%

The model gave an accuracy of 93.89% for the train set. In other words 28543 were correctly classified out of a testing set of 30400. In other words, out of 15153 Non-VPN instances, 14698 had been correctly classified and out of 15247 VPN instances, 13845 had been correctly classified. The precision and recall for the same were 0.96818 and 0.90804 respectively. The F1-score was observed to be 0.93715. The same has been displayed in Table III.

TABLE III Training Set results

Accuracy	93.89%
Precision	96.81%
Recall	90.80%
F1 Score	93.71%
Average True Positive Rate	96.99%
Average True Negative Rate	90.80%

B. Random Forest Model

To classify the connections as VPN and Non-VPN, another algorithm used is Random Forest. The dataset has been divided into a test-train split of 20:80.

Fig. 15 and Fig. 16 represent the confusion matrix for the testing set and training set respectively. The details such as number of False positives, False negatives, True positives and True negatives are represented in the confusion matrix. This is a measure of calculating the amount of correctly and incorrectly classified instances. The model has to be trained in such a way that the number of False positives and True negatives are minimized.

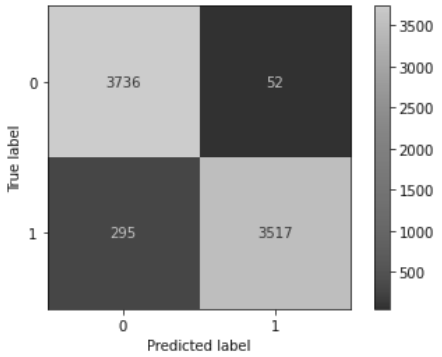

Fig. 15. Confusion Matrix of Test set.

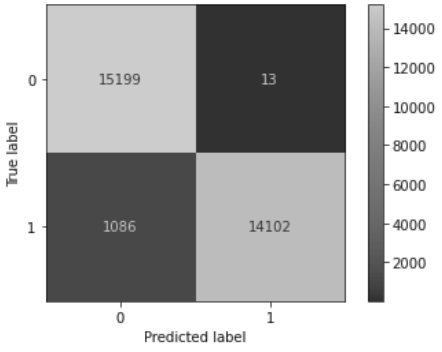

Fig. 16. Confusion Matrix of Training set

The model gave an accuracy of 95.43% for the test set. Hence 7253 instances were correctly classified out of a testing set of 7600. In other words, out of 3788 Non-VPN instances, 3736 had been correctly classified and out of 3812 VPN instances, 3517 had been correctly classified. The precision and recall for the same were 0.98543 and 0.92261 respectively. The F1-score was observed to be 0.95298. The same has been displayed in Table IV.

TABLE IV TEST SET RESULTS

Accuracy	95.43%
Precision :	98.54%
Recall :	92.26%
F1 Score :	95.29%
Average True Positive Rate	98.62%
Average True Negative Rate	92.26%

The model gave an accuracy of 96.38%s for the train set. In other words 29301 were correctly classified out of a testing set of 30400. In other words, out of 15212 Non-VPN instances, 15199 had been correctly classified and out of 15188 VPN instances, 14102 had been correctly classified. The precision and recall for the same were 0.99907 and 0.92849 respectively. The F1-score was observed to be 0.96249. The same has been displayed in Table V.

TABLE V TRAINING SET RESULTS

Accuracy :	96.38%
Precision :	99.90%
Recall :	92.84%
F1 Score :	96.24%
Average True Positive Rate	99.91%
Average True Negative Rate	92.84%

V. CONCLUSIONS

It can be advantageous to summarize the performance of each classifier into a single measure when comparing different classifiers. Calculating the area under the Receiver Operating Curve (ROC), also known as AUC, is the most popular method. The ROC curve depicts the trade-off between sensitivity and specificity (also known as TPR). In our scenario, the true positive rate is the percentage of VPN connections that are accurately identified as VPN. Similarly, the false positive rate is the percentage of VPN connections that are wrongly classified as non-VPN. Classifiers with curves closer to the top-left corner have greater performance than the others, i.e. a classifier curve with a higher AUC has better performance than the others.

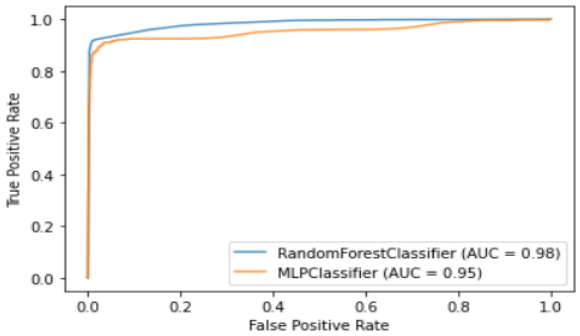

Fig. 17. Receiver Operator Characteristic(ROC) curve

Therefore performance of different classifiers can be compared by calculating the area under the ROC curve. As shown in the fig. 17, Random Forest gives better results as compared to MLP.

Live network traffic data validation was performed. The network traffic was successfully classified into VPN/Non-VPN based on the prediction from the trained machine learning models. The clients were redirected to the required web page according to the prediction, giving access to only the genuine/Non-VPN users.

REFERENCES

[1] S. Jahan, M. S. Rahman and S. Saha, "Application specific tunneling protocol selection for Virtual Private Networks," 2017 International Conference on Networking, Systems and Security (NSysS), 2017, pp. 39-44, doi: 10.1109/NSysS.2017.7885799.

[2] C. M. Nawej and S. Du, "Virtual Private Network's Impact on Network Performance," 2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC), Mon Tresor, Mauritius, 2018.

[3] Z. Wu and M. Xiao, "Performance Evaluation of VPN with Different Network Topologies," 2019 IEEE 2nd International Conference on Electronics Technology (ICET), Chengdu, China, 2019.

[4] S. T. Aung and T. Thein, "Comparative Analysis of Site-to-Site Layer 2 Virtual Private Networks," 2020 IEEE Conference on Computer Applications(ICCA), Yangon, Myanmar, 2020.

[5] D. Deshmukh and B. Iyer, "Design of IPSec virtual private network for remote access," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017

[6] V. Hashiyana, T. Haiduwa, N. Suresh, A. Bratha and F. K. Ouma, "Design and Implementation of an IPSec Virtual Private Network: A Case Study at the University of Namibia," 2020.

[7] Gerard Drapper Gil, Arash Habibi Lashkari, Mohammad Mamun, Ali A. Ghorbani, "Characterization of Encrypted and VPN Traffic Using Time-Related Features", In Proceedings of the 2nd International Conference on Information Systems Security and Privacy(ICISSP 2016) , pages 407-414, Rome, Italy.

[8] Sikha Bagui, Xingang Fang, Ezhil Kalaimannan, Subhash C. Bagui and Joseph Sheehan, "Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features"

[9] Michael J. De Lucia, Paul E. Maxwell, Nathaniel D. Bastian, Ananthram Swami, Brian Jalaian, Nandi Leslie, "Machine learning raw network traffic detection", 2021

[10] Zihao Wang, Kar Wai Fok, Vrizlynn L.L. Thing,"Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study", 2021