**Performance evaluation of secured network traffic classification using a machine learning approach**

**INTRODUCTION:**

Our existence has been completely upended by the Internet. It has completely changed how we communicate, to the point where it is now our go-to method. We use the Internet for virtually everything. Purchasing anything, ordering food, spending time with a friend, and sharing information are some examples.

Due to the vast number of users accessing the Internet for daily activities, there is an excessive volume of network traffic produced, necessitating its effective management.

**MOTIVATION:**

Monitoring network traffic and providing QoS is getting more and more challenging and sophisticated due to the exponential growth in the number of users and the diversity of apps accessing the Internet. Network traffic classification, therefore, becomes an essential component for providing QoS.

Anomaly detection and network analysis are becoming challenging tasks for ISPs to handle in light of the volume of encrypted traffic in use today.

Therefore, it is imperative to adopt a practical strategy that addresses the identification of encrypted network traffic while enhancing network management and QoS.

**RESEARCH PROBLEM:**

The given research paper is concerned with detecting and classifying network traffic as VPN and Non-VPN using various ML algorithms and analyzing the performance of the same.

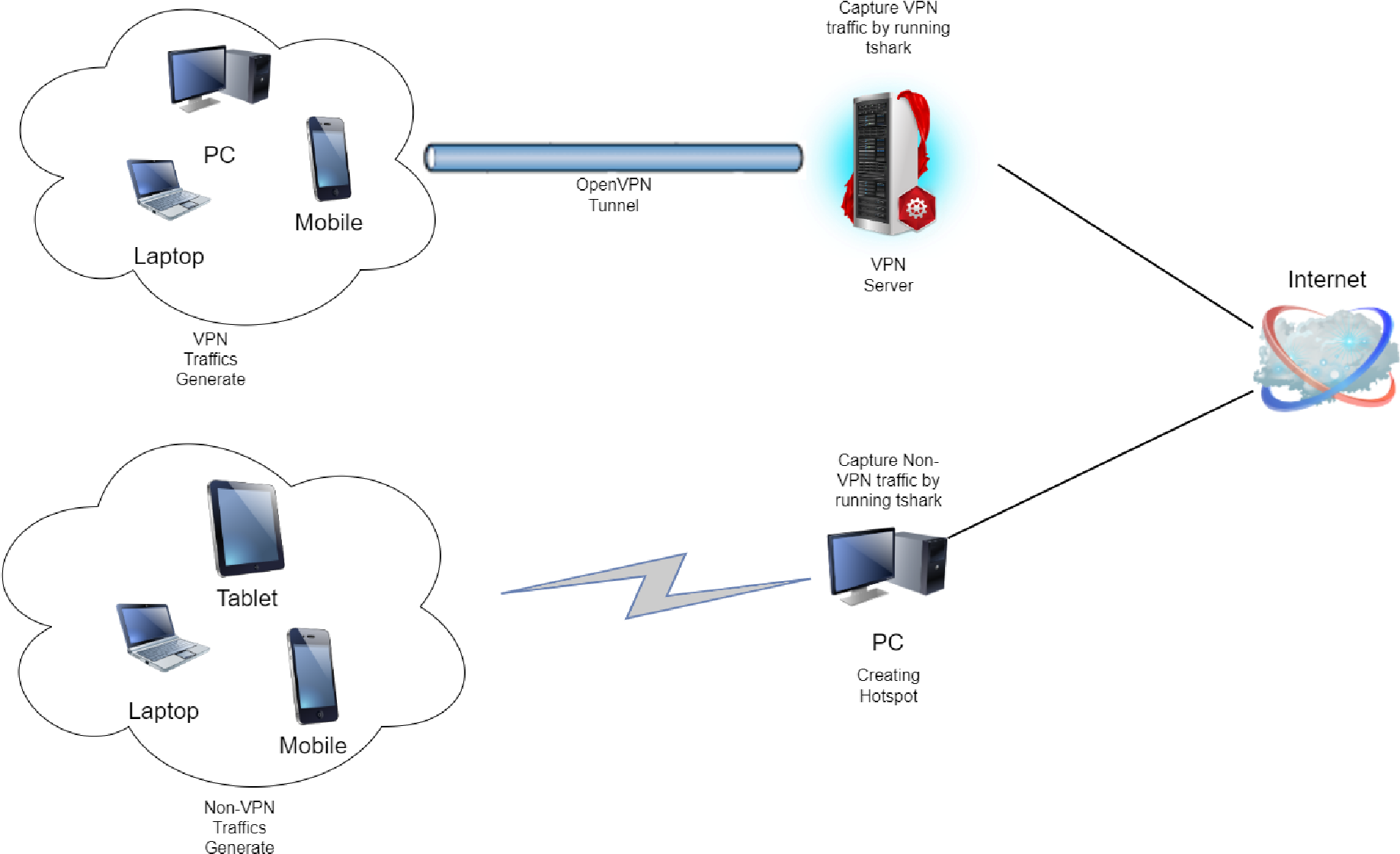
**CONTRIBUTIONS:**

* How to import a dataset, pre-processing, feature analysis, normalization, scaling
* How to train various ML models
* Performance analysis of various ML models
* How to solve the problem of underfitting and overfitting of an ML model
* Exporting an ML model
* Creating own environment to collect VPN/Non-VPN traffic dataset
* Testing various ML models on our own dataset
* Testing the model on real-time traffic and analyzing the performance
* Testing the model on live traffic and analyzing the performance

**Objectives:**

* Classify network traffic as VPN and Non-VPN
* Performance Analysis of different ML algorithms.

**Environment Setup for Real Time Traffic Capture:**



We set up our own environment for capturing VPN and Non-VPN real-time traffic.

**For capturing Non-VPN traffic -** Used a Local Machine as a wireless hotspot to connect various devices to it. Used tshark on the local machine to capture all the traffic generated between client and server. Captured the packets in pcap format. Converted pcap file format into csv file format we used wireshark.

**For capturing VPN traffic -** First we created our own VPN server. For creating our own vpn server we used the linode platform. Online platform Linode offers services to host our own system on their infrastructure. We installed OpenVPN on a Linux server that was hosted on Linode in order to construct a VPN server. After creating the VPN server, clients used the OpenVPN app on their respective devices to connect to the VPN server. The VPN server encrypts the packets that the clients produce as they pass through it on their way to the requested server.We were able to collect VPN traffic from clients by running tshark on the VPN server and saving intercepted packets in pcap format on the server.. To convert pcap file format to csv file format we used wireshark.

**Dataset Description:**

The dataset used in this project includes VPN and non-VPN traffic from browsing, chat, Email, Streaming, File Transfer, VoIP, TraP2P records and was generated using Wireshark and tshark.

The dataset has the following features:

* **Source Ip** : The IP packet field containing the IP address of the workstation from which it came.
* **Destination Ip :** The IP packet field containing the IP address of the workstation to which it is addressed.
* **Protocol :** The networking protocol used for data communication between different devices in the network.
* **Time :** Time of generation of packet.
* **Length :** Length of the generated packet.

**Implementation:**

* Imported the given dataset from the research paper.
  1. Using the pandas python library,
  2. Imported the dataset and visualized the same in the form of dataframes for pre-processing.
* Did pre-processing, feature analysis and scaling.
  1. Dropped NaN values.
  2. Converted object type features to numeric type using Label Encoder.
  3. Scaled the data using normalization technique Standard Scaler.
  4. Analyzed features using Chi-Square, Information Gain method to obtain the significance of features.
* Training and Testing.
  1. For training splitted the dataset into 80/20 ratio.
  2. Trained Decision Tree, Random Forest, Adaboost, K-nearest Neighbors, Xgboost.
  3. Applied PCA to resolve overfitting.
  4. Tested the above mentioned Models.
* Performance analysis of various ML algorithms.
  1. Analyzed performance of the above mentioned models using Accuracy, F1 score, Recall and Precision.
* Created our own environment for creating a new dataset.
  1. Created our own VPN server to collect VPN traffic.
  2. Collected VPN traffic of surfing, streaming, and file download.
  3. Collected Non-VPN surfing, streaming, and file download.
  4. Trained our model on the above dataset.
  5. Exported our model.
  6. Created another dataset at a different time and tested our model on that. ● Performance analysis of various ML algorithms on real-time traffic.

a. Analyzed performance of the above mentioned models using Accuracy, F1 score, Recall and Precision.

**Result Analysis:**

1. **Performance Analysis on Given Dataset**

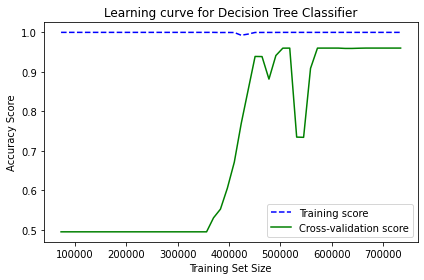
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Class** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| **Decision tree** | 0 | 0.99 | 0.99 | 0.99 | 0.9886 |
| 1 | 0.99 | 0.99 | 0.99 |
| **Random Forest** | 0 | 0.99 | 0.99 | 0.99 | 0.9922 |
| 1 | 0.99 | 0.99 | 0.99 |
| **Adaboost** | 0 | 0.92 | 0.85 | 0.88 | 0.8875 |
| 1 | 0.86 | 0.92 | 0.89 |
| **K-nearest neighbors** | 0 | 0.99 | 0.99 | 0.99 | 0.9906 |
| 1 | 0.99 | 0.99 | 0.99 |
| **Xgboost** | 0 | 0.98 | 0.98 | 0.98 | 0.9774 |
| 1 | 0.98 | 0.98 | 0.98 |

1. **Performance Analysis on Generated Dataset**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Class** | **Precision** | **Recall** | **F1-Score** | **Accuracy** |
| **Decision tree** | 0 | 0.72 | 0.93 | 0.81 | 0.8260 |
| 1 | 0.94 | 0.75 | 0.84 |
| **Random Forest** | 0 | 0.78 | 0.93 | 0.85 | 0.8661 |
| 1 | 0.94 | 0.83 | 0.88 |
| **AdaBoost** | 0 | 0.63 | 0.93 | 0.76 | 0.7551 |
| 1 | 0.93 | 0.63 | 0.75 |

1. **Learning and validation curve visualization of different ML models (Trained on RealTime Traffic):**

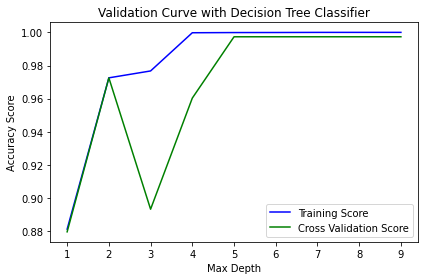
**a. Decision Tree model:**



**Fig-1)** Learning Curve of Decision Tree Model.

From the above curve, we can clearly infer that up to 3.8 lakhs inputs in the training dataset cross-validation score curve is close to zero accuracy. As the size of the training dataset increases above 3.8 lakhs cross-validation score increases exponentially till 4.6 lakhs inputs. We observe slight fluctuations in cross validation score for slight increase in inputs in the training dataset between 4.2 lakhs and 5.5. The cross validation curve becomes constant after

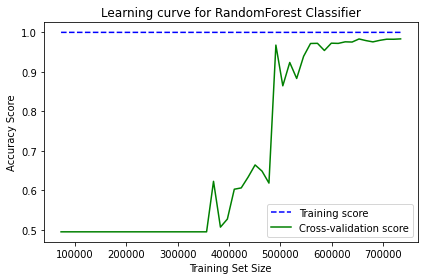
5.5 lakhs units. Therefore increasing the size of the training dataset leads to a better approximation of the cross validation curve. The high accuracy of the training score indicates low bias and high variance. Because the cross-validation score is relatively lower and rises very slowly as the size of the training set increases, this model also starts to overfit the data.



**Fig-2)** Validation Curve of Decision Tree Model.

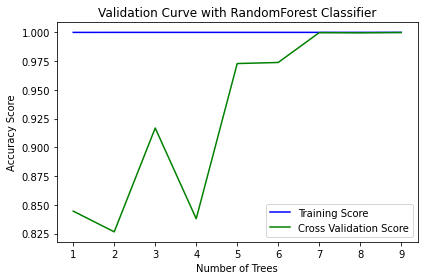
From the above curve, we can clearly infer that as we increase the max\_depth value from 1 to 2 in the Decision Tree algorithm ,the accuracy score of the model increases 88% to 97.4%. After that for max\_depth value of 3, accuracy score decreases to 89%( approx). By increasing max\_depth value above 3 accuracy score increases to 99.7%. and then remains constant, which is an indicator that above max\_depth value of 5, the model starts to overfit.

**b. RandomForest model:**



**Fig-3)** Learning Curve of RandomForest Model.

From the above graph, we can clearly infer that up to 3.5 lakhs inputs in the training dataset cross-validation score curve is close to zero accuracy. We observe slight fluctuations in cross validation score for slight increase in inputs in the training dataset between 3.5 lakhs and 4.9 lakhs.A very steep increase in the cross validation score is observed as the size of the training dataset increases from 4.9 lakhs to 5 lakhs . The cross validation continues to rise very slowly after 5 lakhs units. Therefore increasing the size of the training dataset leads to a better approximation of the cross validation curve. The high accuracy of the training score indicates low bias and high variance. Because the cross-validation score is relatively lower and rises very slowly as the size of the training set increases, this model also starts to overfit the data.

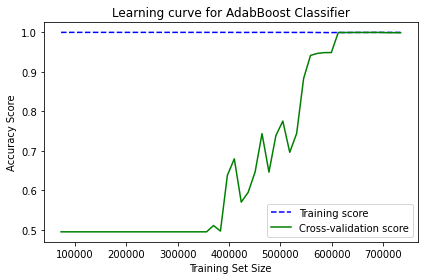


**Fig-4)** Validation Curve of RandomForest Model.

From the above curve, we can clearly see that as we increase the no. of estimators value from 1 to 4 in the RandomForest algorithm, the accuracy score of the model fluctuates between 85% and 92% , 92% and 85%. For no. of estimators between 4 and 5 the accuracy score increases to 97% (approx). As max\_depth value goes above 5 accuracy score increases to

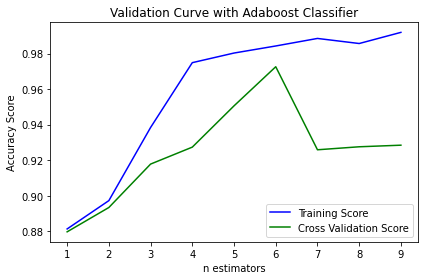
99.9%. and then remains constant which indicates that the model is overfitting.

**c. AdaBoost model:**



**Fig-5)** Learning Curve of AdaBoost Model.

From the above curve, we can clearly infer that up to 4 lakhs inputs in the training dataset cross-validation score curve is close to zero accuracy. We observe a steep rise with fluctuations in cross validation score between 4 lakhs and 6 lakhs. A very steep increase in the cross validation score is observed as the size of the training dataset increases from 4.9 lakhs to 5 lakhs . The cross validation becomes constant after 6 lakhs inputs. Therefore increasing the size of the training dataset leads to a better approximation of the cross validation curve. The high accuracy of the training score indicates low bias and high variance. Because the cross-validation score is relatively lower and rises very slowly as the size of the training set increases, this model also starts to overfit the data.



**Fig-6)** Validation Curve of AdaBoost Model.

From the above curve, we can clearly see that as we increase the no. of estimators from 1 to 4 in the AdaBoost algorithm, the accuracy score of the model increases from 88% to 96% and then decreases from 97% to 92 for estimator value between 6 - 7. For estimators value above 7 accuracy score remains constant to 92.5% (approx). Which shows that the estimator value of 6 gives high accuracy up to 96.5%.

**Conclusion:**

Among the ML algorithms that we trained and tested , Random Forest performed the best in terms of performance analysis parameters for classifying VPN and Non-VPN traffic.

**Future Work**:

We can deploy our model in a gateway and classify live traffic into vpn and non-vpn. And on the basis of that classification we can assign bandwidth, security , priority ..etc to the classified traffic classes.

**Learning outcomes:**

* Core python basics
* Working with python modules like pandas, NumPy, scikit learn, etc
* Machine Learning basics
* Supervised learning and Unsupervised learning
* Classification Machine Learning algorithms
* Basics of Computer Networking
* VPN and Non-VPN traffic
* Basics of Wireshark
* Capturing real-time traffic using Wireshark and tshark
* Creating own environment for VPN server
* Generating our own Dataset

**References:**

1. Afuwape, A. A., Xu, Y., Anajemba, J. H., & Srivastava, G. (2021). Performance evaluation of secured network traffic classification using a machine learning approach. Computer Standards & Interfaces, 78, 103545. doi:10.1016/j.csi.2021.103545

**Given Dataset:** <https://www.unb.ca/cic/datasets/vpn.html>

**Used Dataset**:

https://drive.google.com/file/d/1nrWozfjc0GW0TOnxnuxoEyKtuwPcseyh/view?usp= sharing

1. Source code and dataset of Real Time Traffic

**Generated Dataset (Real Time Traffic):**

[https://drive.google.com/drive/folders/1MqQKPdDNecNlQmVQM2ZuDc3j2DQDipt O?usp=share\_link](https://drive.google.com/drive/folders/1MqQKPdDNecNlQmVQM2ZuDc3j2DQDiptO?usp=share_link)

**Source code:** https://github.com/Tarunkashyap6665/vpn\_non-vpn\_classsification.git