Final Project Report

Introduction

TOR

The onion router is free and open-source software designed to enable anonymous communication over the internet by routing their traffic through multiple servers in the TOR network and encrypting it at each step. It is often used by individuals who prefer to protect their identity online and to have privacy. Due to the anonymity offered by TOR, it has become a medium for engaging in illegal activities like drug trafficking, money laundering and many more.

TOR Circuit

The TOR circuit consists of entry/guard, middle (one or more), and exit node. Entry node only knows the identity of the user and next node in the chain. Middle Node decrypts the traffic and forwards it to the next relay/destination and exit node which serves as the last node before destination in the circuit. This is pictorially represented in Fig 1.

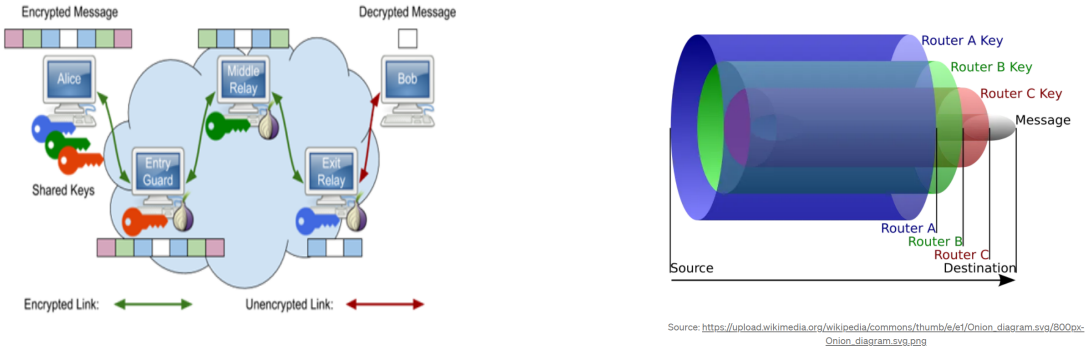
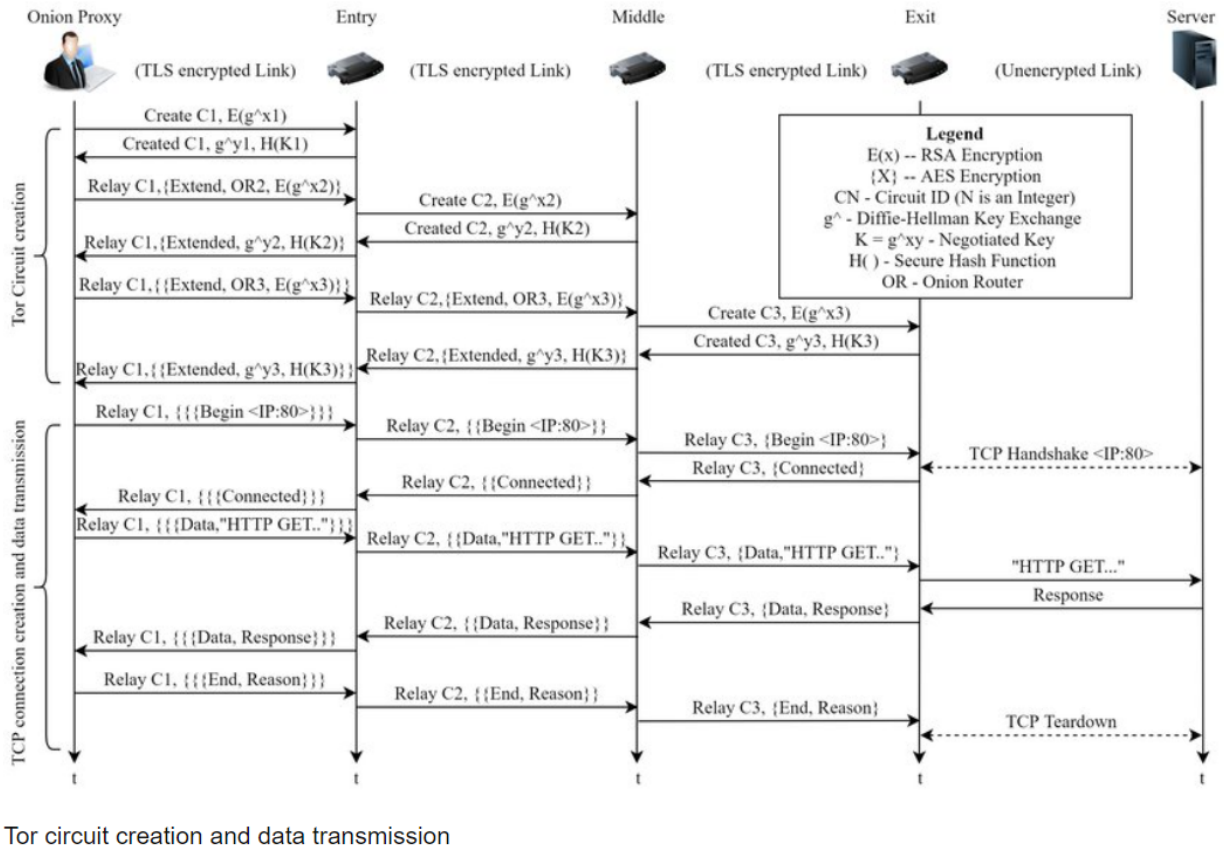


Fig 1: The nodes in TOR

The nodes are randomly selected from a list of publicly known relays to form the network. The client then negotiates a series of symmetric keys with each node in the circuit which is used to encrypt and decrypt the traffic through the network.



Tor circuit creation and data transmission

Fig 2: TOR circuit creation

Fig. 2 explains the step-by-step representation of circuit creation including establishment of symmetric keys with each node which is essentially done to decrypt then encrypt and send to the next node in the chain till the destination is reached.

TOR architecture and working

When a user wants to connect to the internet via TOR, they use the free TOR software and establish connection to the TOR network. The client software generates random symmetric encryption keys. The TOR software then encrypts the user data with three or more layers of encryption according to the nodes in the circuit. The data is sent to the entry node. It decrypts the outermost layer of the encryption to reveal the address of the next node in the circuit and forwards the remaining data to the next node. The second node receives the data and decrypts the layer of encryption and forwards to the next node in the chain. The process continues till the exit node and the data is stripped off the final layer of encryption before sending to the destination at the exit node. The destination then responds to the request it received and sends to the exit node where the data is encrypted with the levels of encryption in the reverse order.

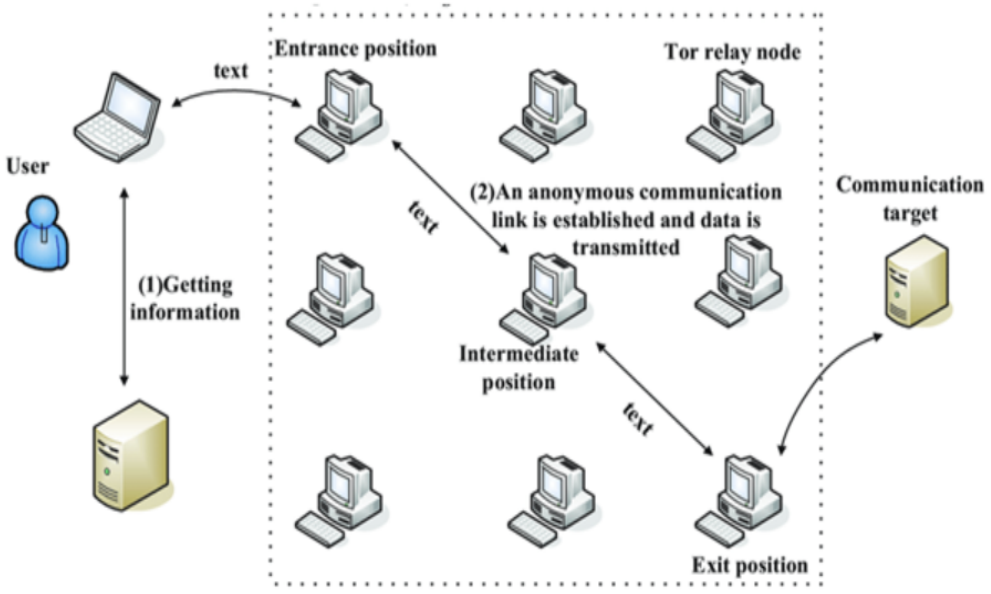


Fig 3: TOR architecture

The whole process ensures that each node only knows the identity of the previous and the next immediate node in the circuit. Therefore, no single node knows the entire route of the data and because of encryption, the data cannot be read by the attacker who intercepts it.

TOR traffic can be identified by couple of mechanisms namely network fingerprinting, deep packet inspection, traffic analysis and DNS requests. It has unique fingerprinting as the ports mostly used for communication are specific such as 9001, 9030 and 9050 (also custom ports). Analyzing the traffic pattern, frequencies, encryption type can shed some light on differentiating between TOR and nonTOR traffic. The DNS requests used in TOR will have domain .onion which is also a differentiating factor.

TOR can easily be leveraged by attackers to conceal their identity and to carry out malware operations. An attacker can use TOR to distribute malware by setting up command and control server over the TOR network because of the anonymity offered by TOR. Malware can be hosted on TOR hidden services. TOR enabled email and file sharing networks can also be used to distribute malware.

Malicious TOR traffic can be detected by noticing suspicious network connections. For example, malware often communicates with C&C servers using domain generation algorithms (DGAs), which generate random domain names that are difficult to block or identify.

Proposed Solution

"Exposing the Rat in the Tunnel: Using Traffic Analysis for Tor-based Malware Detection" proposed a solution of **training machine learning models for network packet analysis to classify the tor-malware versus benign traffic.**
Models trained for the purpose included the following:

- Random Forest
- KNN
- Extra Trees
- Logistic Regression
- XGboost
- Light Gradient Boosting Machine (LightGBM)
- Tabular Neural Network

The best performing model out of the above mentioned model was **LightGBM.**
Below are the results mentioned.

Experiment	Model	Precision(%)	Recall(%)	FPR(%)	AUC(%)
E1	XGBoost	86.13	63.37	1.53	93.62
E2	LightGBM	90.96	76.34	1.45	96.91
E3	LightGBM	93.33	81.60	0.88	98.56
E4	DF	75.51	60.76	3	91.24
E5	Var-CNN	85.65	35.59	0.9	86.27
E6	Var-CNN	91.48	55.13	0.78	89.23

Fig4. Results

Literature Review

Ways of DataML approaches prove effective in detecting malwares and other cyber-attacks. [4] shows that the cyber-attacks including tor malware classification are detected and classified using machine learning and neural network algorithms. [5], [8] and [9] shed light on how deep learning and neural networks is could be used to analyze network traffic to detect and classify attacks on tor as wells as non-tor

platforms. These works emphasize on how ML, neural networks and deep learning techniques could be leveraged to predict malwares and other cyber-attacks.
The proposed paper focuses on differentiating between benign and malicious encrypted TOR connections using machine learning approach and predicting the malware class using multilabel classification with AutoGluon (Light Gradient Boosting Machine (LightGBM), CatBoost, XGBoost, Random Forests, ExtraTrees, kNN, Logistic Regression, and Tabular Neural Network models). A similar work is done in [1] where the approach was to generate realistic attack samples to train machine learning models tor malware traffic detection. ML methods like DNN, SVM, LSTM, RNN, and GRU were used, and performance was compared. The AutoGluon model used in the proposed paper helps to achieve state-of-the-art performance on tasks such as text classification and prediction because of its multi-layer model ensemble, hence better results were achieved. [3] and [7] also analyzes network traffic to classify data but focused on classifying just tor and non-tor traffic with machine learning approach like logistic regression, decision tree, random forest, adaboost, SVM, KNN. [7] used ANN in the last layer to evaluate the data.
Similar to the proposed paper, [6] uses virusTotal platform for network traffic generation and focused on classification of tor malware traffic where ML algorithms like decision tree, K-nearest neighbors, Naive Bayes, and Random Forest are used for detection. [2] also focused on using data in the form of encrypted data payload to classify tor traffic. Algorithms like J48, random forest and KNN were used.

Malware Data Collection

collection

- Download Tor-based Malware binaries from **VirusTotal(VT)** to the virtual environment, upload those binaries to **Hybrid analysis Falcon sandbox**, for the analysis report and pcaps generated by the sandbox
- Download the already available tor based malware binaries from hybrid analysis Falcon sandbox, after going through a vetting process.
- Download publically available malware pcap files, **link**

Reason for Not collecting malware data

- The data collection required us to explore ways to mitigate risks which could have been raised from downloading the malware binary to the virtual environment. Those precautions varied from malware to malware, the short span of time did not allow us to explore the precautions as well as even after taking all

the precautionary measures enough data could not have been collected in a limited time.

- Another reason for not being able to collect the already available pcaps in the platform of Falcon sandbox, required vetting process, for safe usage for the data, which again due to shortage of time could not have been possible.
- At the very end of the semester we were able to find the sources which did not require any vetting process for downloading malware based pcap files.
- Handling malware on the personal computer can cause sensitive information stored on the device to be stolen and can install software which may track your activities.
- In the perspective of running malware on the virtual machine, if connected to the internet, it can become a medium for propagating malware as it would be communicating to it C&C center. In this case, the machine asks like a bot in a botnet network. Thus, the VM becomes a host to carry out malicious activities.

Our Work
Classification of Tor & Non-Tor traffic

Dataset

The Dataset used for the classification of Tor Non-Tor traffic is from UNB(University of New Brunswick) , canadian institute of cybersecurity **"Tor-nonTor dataset (ISCXTor2016)"**

Exploratory Data Analysis

The dataset used was skewed towards Non-Tor traffic.
"Pandas" library was used to visualize and then extract equal parts of Tor Non-Traffic from the dataset.
The dataset had 2 rows with NaN values, **the rows were removed completely.**

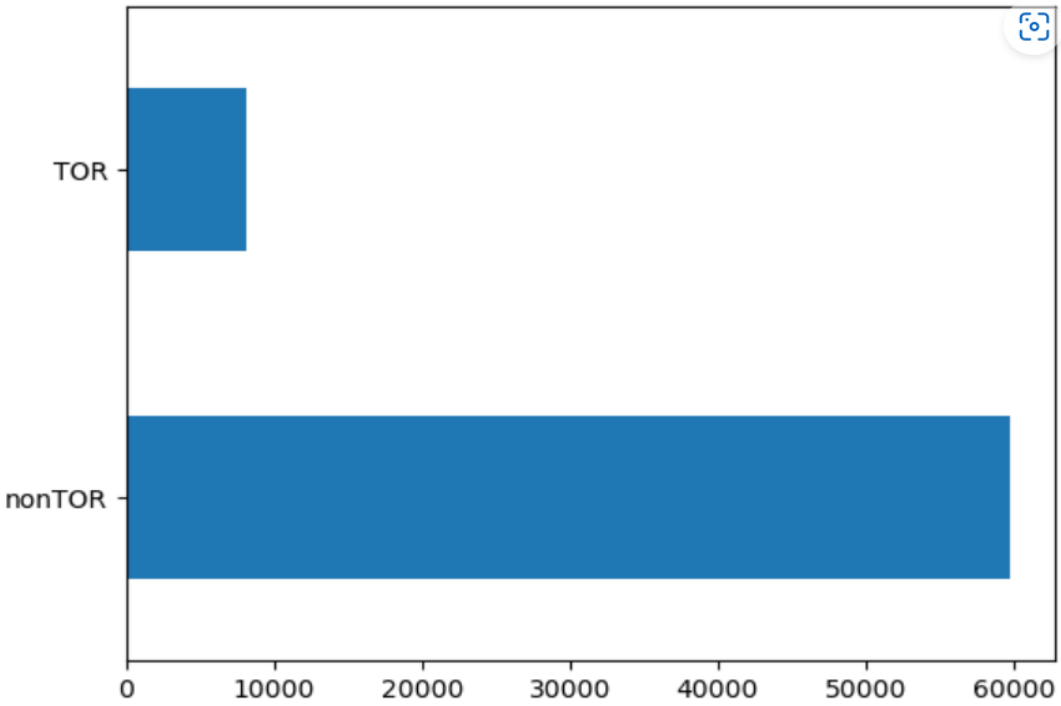


Fig5.Depiction of Tor Non-Tor traffic distribution of original dataset

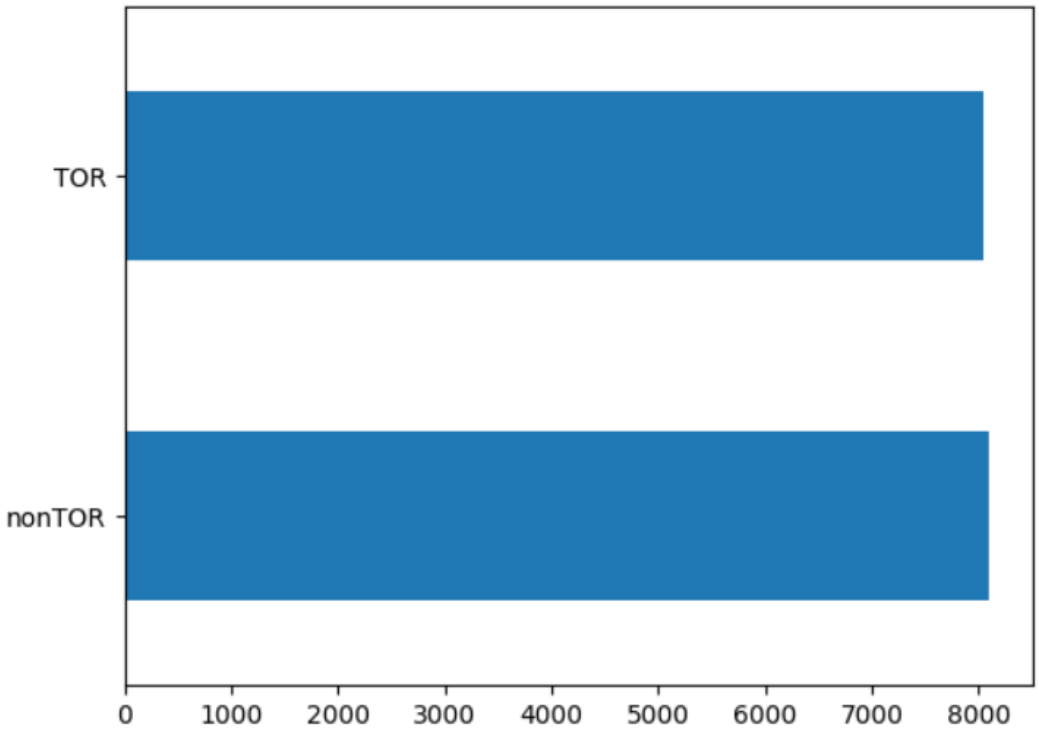


Fig6.Depiction of Tor Non-Tor traffic distribution of our version of dataset

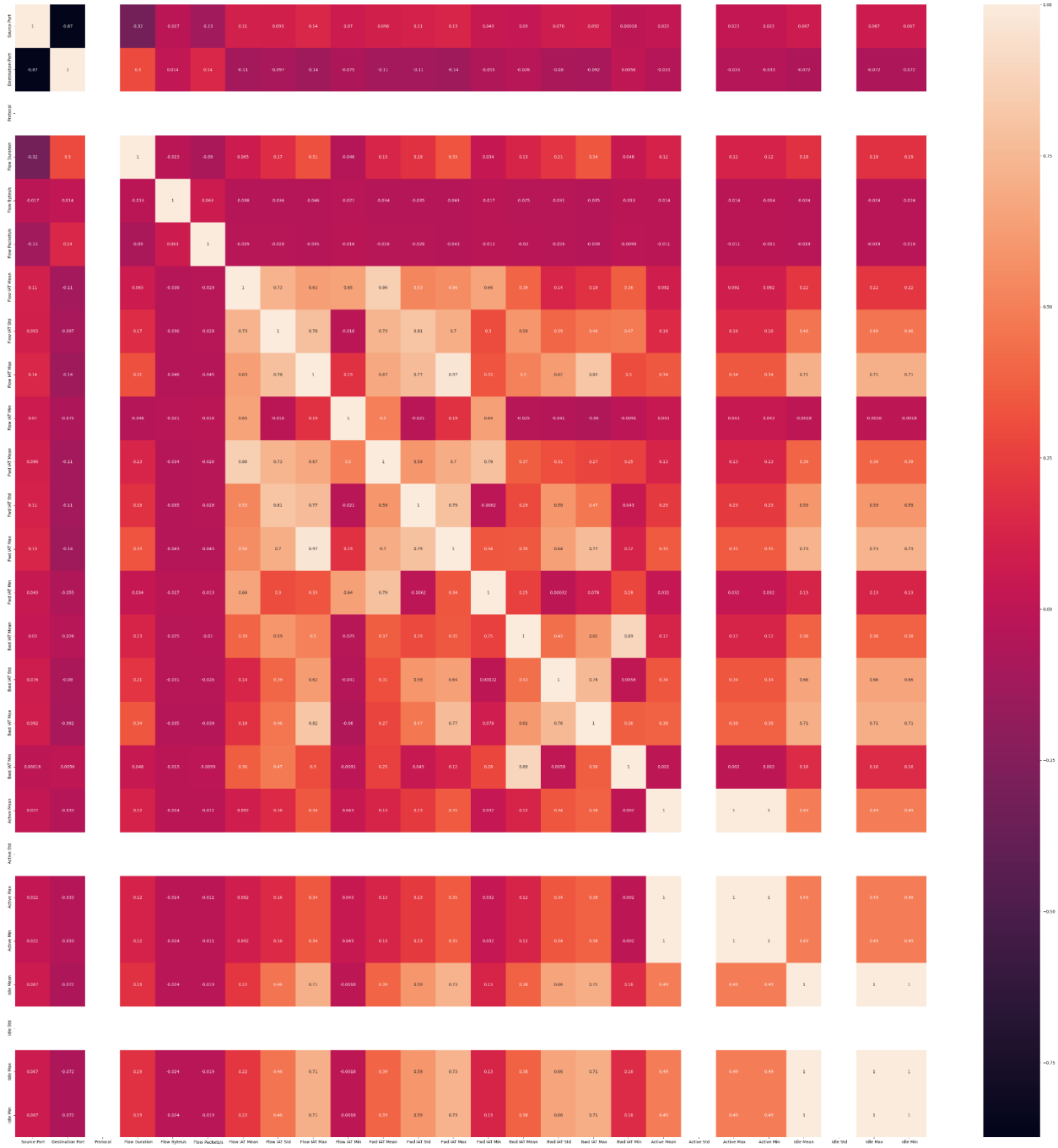


Fig7. Heatmap explaining correlation between different fields of dataset (zoom in to observe clearly)

Methodology

- Machine learning models were trained for the classification of Tor Non-Tor traffic.
- Libraries used: sklearn, keras, xgboost, lightgbm
- Models Trained: Random forest, KNN Classifier, Extra Trees, Logistic Regression, Lightgbm model, Tabular Neural Network, and Xgboost.

Result

Accuracy for **Random forest** model: 0.987611496531219
F1 score of the **random forest** model: 0.9876029453141222
Precision score of the **random forest** model: 0.9880237488897217
Recall score of the **random forest** model 0.987439284142354

Accuracy for **KNN model**: 0.9992566897918731
F1 score of the **KNN model**: 0.9992565415045161
Precision score of the **KNN model**: 0.9992466097438473
Recall score of the **KNN model** 0.999267578125

Accuracy for **Extra trees** model: 0.9960356788899901
F1 score of the **extra trees** model: 0.9960348600252046
Precision score of the **extra trees** model: 0.9960280385308753
Recall score of the **extra trees** model 0.9960421709947183

Accuracy for **logistic regression** model: 0.9883548067393458
F1 score of the **logistic regression** model: 0.9883505665628
Precision score of the **logistic regression** model: 0.9884466409425778
Recall score of the **logistic regression** model 0.9882896008865694
Accuracy for **light gbm** model: 0.9992566897918731
F1 score of the **light gbm** model: 0.9992565309135588
Precision score of the **light gbm** model: 0.9992529748988268
Recall score of the **light gbm** model 0.999260209695674

Accuracy of **xgboost**: 0.9992566897918731
F1 score of the **xgboost**: 0.9992565309135588
Precision score of the **xgboot**: 0.9992529748988268
Recall score of the **xgboost** 0.999260209695674

Accuracy of **Tabular Neural Network**: 0.9341317415237427

Precision of the **Tabular Neural Network**: 0.9132565309135588

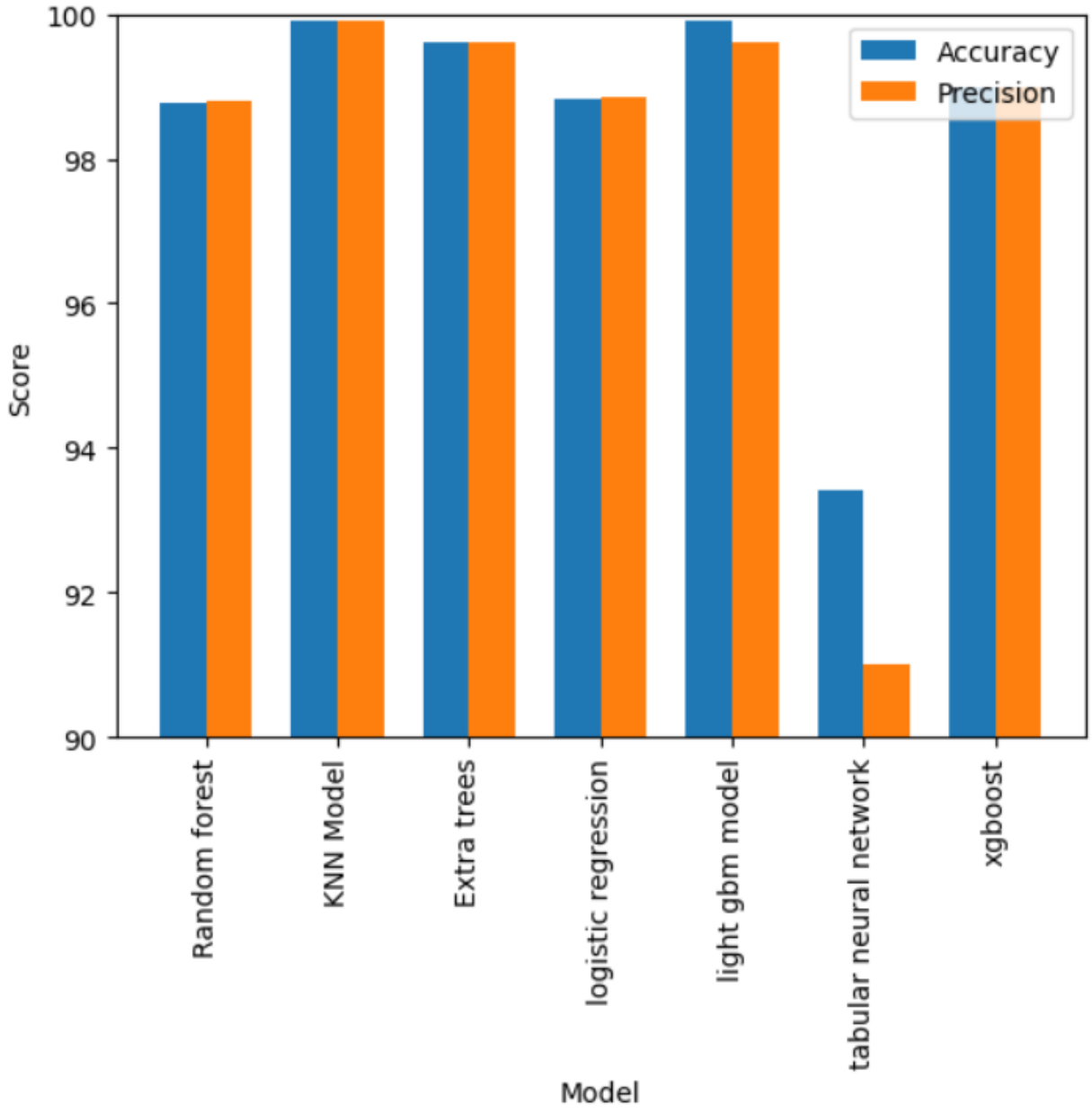


Fig8.Final Results

References

1. X. Hu, Y. Gao, G. Cheng, H. Wu and R. Li, "An Adversarial Learning-based Tor Malware Traffic Detection Model," GLOBECOM 2022 - 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 2022, pp. 74-79, doi: 10.1109/GLOBECOM48099.2022.10001131

2. P. Choorod and G. Weir, "Tor Traffic Classification Based on Encrypted Payload Characteristics," 2021 National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 2021, pp. 1-6, doi: 10.1109/NCCC49330.2021.9428874.

3. M. Soykan and P. S. Bölük, "Tor Network Detection By Using Machine Learning And Artificial Neural Network," 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates, 2021, pp. 1-4, doi: 10.1109/ISNCC52172.2021.9615730

4. Zhao, F., Zhang, H., Peng, J. et al. A semi-self-taught network intrusion detection system. Neural Comput & Applic 32, 17169–17179 (2020). https://doi.org/10.1007/s00521-020-04914-7

5. Thirumaran, M. "The LSTM-Based Automated Phishing Detection Driven Model for Detecting Multiple Attacks on Tor Hidden Services." (2022).

6. de Robles, Marie Betel B., Joseph Anthony C. Hermocilla, and Jaderick P. Pabico. "Characterization and classification of malware traffic over the tor network." In Proceedings of Philippine Computing Science Congress, pp. 78-88. 2020.

7. A. Gurunarayanan, A. Agrawal, A. Bhatia and D. K. Vishwakarma, "Improving the performance of Machine Learning Algorithms for TOR detection," 2021 International Conference on Information Networking (ICOIN), Jeju Island, Korea (South), 2021, pp. 439-444, doi: 10.1109/ICOIN50884.2021.9333989.

8. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

9. S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," in IEEE Communications Magazine, vol. 57, no. 5, pp. 76-81, May 2019, doi: 10.1109/MCOM.2019.1800819.