Trojan Detection Using Machine Learning :

APPROACHES:

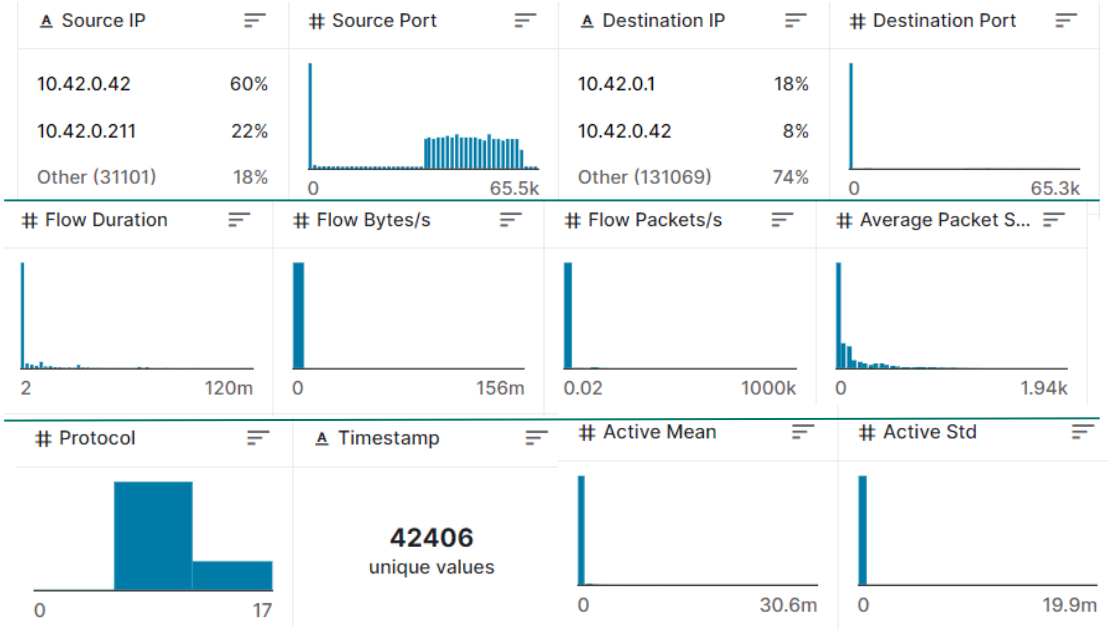
- **Text Data Mining:** Extracting valuable insights, patterns, and knowledge from textual data. It involves applying various techniques from natural language processing (NLP), machine learning, and statistical analysis to understand and analyze unstructured text data.
- **Performance Based Trojan Forensics:** Detecting and analyzing trojans by examining the performance metrics and behavior of a system. It involves monitoring various system-level performance indicators such as network traffic, disk activity, and other relevant metrics.

Approach to choose

- Text mining approach needs multiple trojan programs from all over the world.
- Our trojan program of .exe (for pc) and .apk (for Android) is unable to be decoded into even hexadecimal form on any platform since it is a malicious program.
- Due to the non-availability of resources for data mining, we have to drop this approach.
- Let’s go toward the second approach: Performance-Based Trojan Forensics.

Dataset chosen for Training

- **Dataset taken from Central Information Commission**



and many more……

Overview of Dataset

Here are some common features found in Trojan horse traffic datasets and their explanations:

- **Source IP Address:** The IP address of the device that initiated the network traffic. It helps identify the origin of the malicious activity.
- **Destination IP Address:** The IP address of the device to which the network traffic is directed. It provides information about the target of the malicious activity.
- **Source Port:** The port number used by the source device for network communication. It indicates the specific application or service initiating the traffic.
- **Destination Port**: The port number used by the destination device for the network communication. It indicates the specific application or service receiving the traffic.

Overview of Dataset .....

- **Protocol**: The network protocol used for the communication, such as TCP (Transmission Control Protocol) or UDP (User Datagram Protocol).

- **Packet Length**: The size of the network packets exchanged between the source and destination devices. It helps identify unusual packet sizes that may be indicative of malicious behavior.

- **Timestamp**: The timestamp of each network packet or event, indicating the time at which it occurred. It enables the analysis of the temporal patterns and sequencing of malicious activities.

- **Payload Data**: The actual data payload within the network packets. It may contain encrypted or encoded information used by the Trojan horse malware for its operations.

Overview of Dataset .....

- **Network Flow Features**: These features capture information about network flows, which are sequences of packets sharing common characteristics, such as source and destination IP addresses, ports, and protocols. Flow features can include the total number of packets in a flow, total bytes transferred, flow duration, and more.

- **Statistical Features**: Various statistical features can be extracted from the network traffic, such as mean, standard deviation, minimum, maximum, and entropy of packet lengths or inter-packet arrival times. These features provide insights into the statistical properties of the traffic associated with the Trojan horse activity.

- **Behavior-based Features**: Features based on behavioral patterns of network traffic can also be included. For example, the rate of outgoing connection attempts, connection durations, or the frequency of certain network protocol commands can be used to identify suspicious behavior.

Analyzing Dataset Features

- **Source IP Address**: Multiple distinct source IP addresses with suspicious or unusual activity patterns, indicating a potential botnet or distributed Trojan infection.
- **Destination IP Address**: Unusual or suspicious destination IP addresses that are known for hosting malicious content or acting as command-and-control servers for Trojans.
- **Source Port**: Use of non-standard or uncommon source ports for communication, potentially indicating an attempt to evade detection or bypass firewall rules.
- **Destination Port**: Connections to well-known Trojan-associated ports, such as ports commonly used by remote access Trojans (RATs) or backdoors.
- **Protocol**: Utilization of uncommon or non-standard protocols for communication, which may indicate covert communication channels or attempts to bypass network security measures.
- **Packet Length**: Abnormally large or small packet lengths compared to the normal network traffic patterns, suggesting data exfiltration or obfuscation techniques used by the Trojan.

Analyzing Dataset Features……

- **Timestamp**: Unusual traffic patterns occurring at specific times or intervals, indicating scheduled activities or periodic communication with command-and-control servers.
- **Payload Data**: Presence of suspicious or encrypted payload data that may contain malicious commands, instructions, or data exchanged between the Trojan and its control server.
- **Network Flow Features**: Unusual or high-frequency network flows involving the same or multiple source and destination IP addresses, suggesting coordinated or distributed malicious activities.
- **Statistical Features**: Abnormal statistical properties of packet lengths or inter-packet arrival times, indicating deviations from the normal network traffic behavior due to the presence of a Trojan.
- **Behavior-based Features**: Anomalies in connection rates, durations, or protocol command frequencies that deviate significantly from legitimate network traffic patterns, signaling the presence of a Trojan-related behavior.

Analyzing Dataset Features……

- **Timestamp**: Unusual traffic patterns occurring at specific times or intervals, indicating scheduled activities or periodic communication with command-and-control servers.
- **Payload Data**: Presence of suspicious or encrypted payload data that may contain malicious commands, instructions, or data exchanged between the Trojan and its control server.
- **Network Flow Features**: Unusual or high-frequency network flows involving the same or multiple source and destination IP addresses, suggesting coordinated or distributed malicious activities.
- **Statistical Features**: Abnormal statistical properties of packet lengths or inter-packet arrival times, indicating deviations from the normal network traffic behavior due to the presence of a Trojan.
- **Behavior-based Features**: Anomalies in connection rates, durations, or protocol command frequencies that deviate significantly from legitimate network traffic patterns, signaling the presence of a Trojan-related behavior.

Should we do Feature Selection?

- **RFE**: Recursive Feature Elimination can be a suitable choice for Trojan traffic datasets because it iteratively eliminates less important features based on their rankings. This can help identify the most relevant features associated with the Trojan activities. RFE provides explicit feature importance rankings, which can be valuable in understanding the importance and impact of different features on the presence or behavior of Trojans.

- **Sequential Feature Selection**: Sequential Feature Selection methods, such as Forward Selection or Backward Elimination, can also be effective for Trojan traffic datasets. These methods explore different feature subsets, evaluating the model's performance as features are added or removed. Sequential Feature Selection can be particularly useful when searching for an optimal or near-optimal feature subset that maximizes the detection or classification performance of Trojan activities.

Should we do Feature Selection?

- **Decision Taken:** We chose RFE, as we are not only concerned about the performance of our model but also analyze the importance of each feature for trojan detection.
- RFE ranks the features according to their importance.

- **NOTE:** Dimension Reduction causes data loss. Let's see….

Which model to work on?

There are three best-performing models to use:

- **Random Forest**: Suitable choice for Trojan traffic datasets due to its ability to handle high-dimensional data, capture complex relationships, and provide feature importance rankings. It can effectively handle a mix of categorical and numerical features and can handle imbalanced datasets. Random Forest is often considered robust against outliers and missing values. It can be a good choice if interpretability and feature importance analysis are desired.

- **Support Vector Machines (SVM):** SVM is a powerful algorithm known for its ability to handle complex and nonlinear classification problems. It can work well with small to medium-sized datasets and can handle high-dimensional feature spaces. SVM with appropriate kernel functions can effectively capture intricate decision boundaries. However, SVM might require careful tuning of hyperparameters, and the training process can be computationally expensive for large datasets.

Which model to work on?

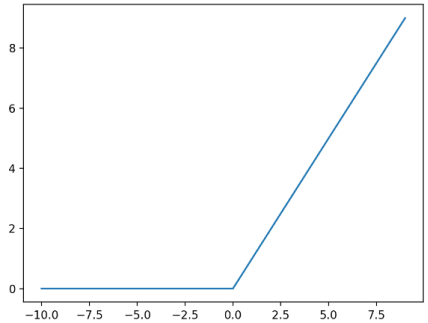
- **Neural Networks**: Neural networks, particularly deep learning models, have shown excellent performance in various domains, including network traffic analysis. They can effectively capture complex patterns and relationships in the data.

- **Let’s work on Neural Network…..**

Structure of our best neural model trained

- **We made two models:**
  - One upon complete feature dataset.
    - Number of features = 85
  - Another upon RFE applied dataset.
    - Number of features = 25
- **Activation Function:**
  - **ReLU:** widely used activation function in deep learning due to its simplicity and effectiveness.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               11008

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 64)                4160

 dense_3 (Dense)             (None, 64)                4160

 dense_4 (Dense)             (None, 1)                 65

=================================================================
Total params: 27,649
Trainable params: 27,649
Non-trainable params: 0
```



Optimizer chosen: Adam Vs SGD

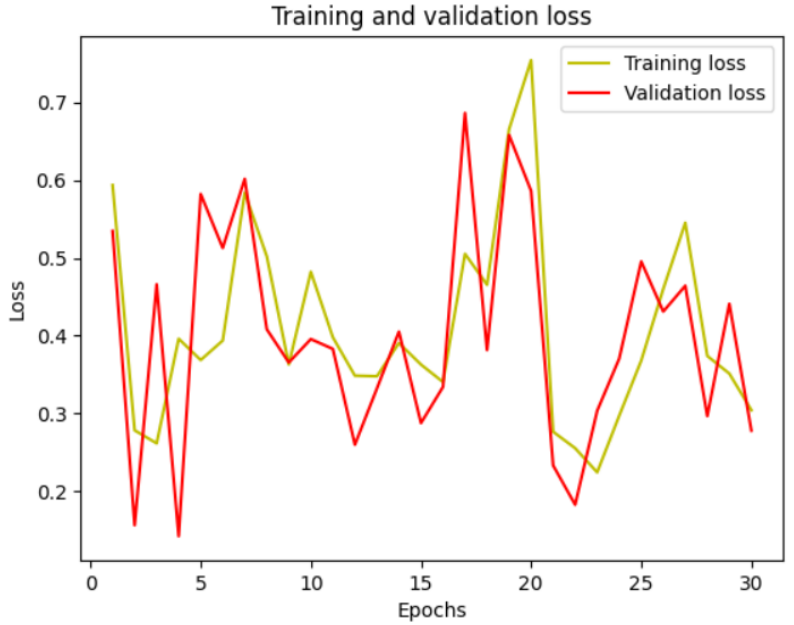
- In machine learning and deep learning, an optimizer is an algorithm or method used to adjust the parameters of a model during the training process. The goal of an optimizer is to minimize the loss or error of the model by finding the optimal set of parameter values.
- **Learning Rate**
  - Adams uses Adaptive Learning Rate
    - help Adam converge faster by taking larger steps when the gradients are large and smaller steps when the gradients are small.
  - SGD uses Fixed Learning Rate
- **Optimizer chosen:** Adams Optimizer

Loss Function

- **Binary Cross Entropy Loss**
- We chose Binary Cross Entropy Loss as we are working on the binary classification problem of identifying trojan or benign.
- Effective in training models to distinguish between the two classes by penalizing deviations from the true binary labels.

Loss Function

- **Binary Cross Entropy Loss**
- We chose Binary Cross Entropy Loss as we are working on the binary classification problem of identifying trojan or benign.
- Effective in training models to distinguish between the two classes by penalizing deviations from the true binary labels.



Experiments upon Hidden Layers

- We started with a single-layer neural network.
- Then started increasing the depth of the neural network.
- Further, increased the width of the neural network.
- With each step followed, our model started getting trained better.

Results of our final models

- As discussed earlier, we worked on two datasets two trains the model.
- **Model 1** with complete features and **Model 2** with reduced feature set.
- **Model 1:** Our model 1 is highly accurately trained with an accuracy of 99.12 %
- **Model 2:** Model 2 didn’t perform well due to feature reduction, there is data loss.

What next .....

- Our model is ready to use.
- One has to make a software application to measure all the features upon which our model is trained.
- Limitations: Might fail upon New Trojan Variants