# Detecting Traffic and Vulnerability exploits against a target application using Network Intrusion Detection System

**Bonumaddi Ramya**
Department of CSE, Vignan’s Institute of Engineering for Women,
Visakhapatnam, India
Email: rakshithabonumaddi@gmail.com

**Dadhirao Jyothsna**
Department of CSE, Vignan’s Institute of Engineering for Women,
Visakhapatnam, India
Email: dadhiraojyothsna@gmail.com

**Abstract** This project addresses the pressing need for robust network security through the development of a Network Intrusion Detection System (NIDS) that seamlessly integrates network security principles with advanced machine learning techniques. Existing methods often fall short in adapting to dynamic threat landscapes. Existing Network Intrusion Detection (NID) methods exhibit notable drawbacks that hinder their efficacy in addressing the dynamic cybersecurity landscape. Signature-based detection, reliant on known attack signatures, proves inadequate against emerging threats lacking predefined signatures. Anomaly-based methods struggle with false positives and negatives due to challenges in distinguishing normal variations in network behaviour from malicious activities. Heuristic-based detection's reliance on predefined rules leads to limited adaptability to evolving attack techniques. Behaviour-based detection may generate false positives stemming from variations in legitimate network behaviour. Rule-based detection, prone to high false positives and negatives, faces challenges in adapting to diverse attack strategies. Additionally, packet filtering methods are constrained in their ability to detect complex attacks involving packet manipulation or encryption. These drawbacks highlight the need for more advanced and adaptive approaches, such as machine learning, to bolster the effectiveness of NID systems in proactive threat detection. The proposed method mitigates the limitations of existing approaches by leveraging machine learning, enabling the system to adapt and learn from evolving threats. It overcomes the drawbacks of static signature-based methods by dynamically analyzing network behavior. False positives and negatives are reduced through continuous learning and the ability to detect anomalies beyond predefined rules. The machine learning model enhances adaptability to changing attack techniques, improving overall detection capabilities in dynamic network environments. The proactive nature of the proposed method enables the identification of novel threats, addressing the shortcomings of signature-based and rule- based systems. Overall, the integration of machine learning mitigates the limitations of traditional NIDS methods, providing a more robust and effective solution for network intrusion detection.

**Keywords:** Network Intrusion Detection (NID), Signature-based detection, Anomaly-based methods, False positives and negatives, Malicious activities, Packet filtering, Novel threats, MachineLearning

## 1 Introduction

In the current digital era, we are aware that everything is moving online. Confidential meetings, online banking, or any other service we use online is at risk, and various types of intruders or attackers can attack in our network to get data from our node. As a result, it is important to give the user a model that allows them to recognize and detect the different types of attacks that could occur in their system. For many of their professional, social, and personal activities, many individuals rely on the Internet. The frequency of network attacks, including those carried out by hackers, crackers, and criminal businesses, has risen, which has an influence on the accessibility, confidentiality, and integrity of vital data. However, there are also others who make attempts to harm our machines that are connected to the Internet, invade our privacy, and disable Internet services. Any technique, procedure, or tool used to maliciously try to breach network security is referred to as a "network attack". An individual or individuals might want to attack corporate networks for a variety of reasons. Network attackers, hackers, or crackers are frequent names. The expansion of digital networks and the increasing reliance on interconnected systems have transformed the landscape of modern communication and business operations. However, this interconnectedness also brings with it a heightened risk of cyber threats and malicious activities aimed at compromising the integrity, confidentiality, and availability of sensitive information. In such a dynamic and evolving environment, the need for robust and effective network security measures is paramount.

## 2 Problem Definition

The problem of network attack classification involves developing a predictive model capable of accurately distinguishing between benign network traffic and malicious attacks within a given dataset. The task is to build a network intrusion detector, a predictive model capable of distinguishing between bad connections, called intrusions or attacks, and good normal connections. The objective is to classify network connections or events as either normal (non-malicious) or malicious based on their characteristics and attributes. This entails identifying patterns, anomalies, and indicators of compromise within network traffic data to enable proactive detection and mitigation of cyber threats. The ultimate goal is to enhance the security posture of network infrastructures by providing timely and reliable alerts for potential attacks, thereby minimizing the risk of unauthorized access, data breaches, and service disruptions.

# 3 Existing System

Before the advent of machine learning, network attack classification primarily relied on rule-based or signature-based methods. These approaches involved predefined rules or signatures to identify known attack patterns within network traffic. Some common existing systems for network attack classification before the widespread adoption of machine learning include: **Firewalls:** Firewalls are one of the oldest and most widely used network security tools. They act as a barrier between a trusted internal network and untrusted external networks, filtering incoming and outgoing traffic based on predefined rules or policies. While primarily used for access control and packet filtering, firewalls can also be configured to detect and block certain types of network attacks based on known signatures or patterns. **E+** IDS were developed to detect and respond to suspicious or malicious activities within computer networks. Traditional IDS typically operated in two modes: signature-based detection and anomaly-based detection. Signature-based IDS relied on predefined attack signatures or patterns to identify known attacks, while anomaly-based IDS monitored network traffic for deviations from normal behaviour, which could indicate potential attacks.

## 3.1 some of the existing systems

### 3.1.1 Rule based system:

Snort: Snort is an open-source network intrusion detection system (NIDS) that gained popularity for its signature-based detection capabilities. It utilizes a rule-based engine to analyse network packets and trigger alerts for traffic that matches predefined rules or signatures associated with known attacks.

### 3.1.2 Signature based system:

Packet Sniffers: Packet sniffers, such as Wireshark and tcpdump, are tools used to capture and analyse network traffic at the packet level. While not specifically designed for attack classification, packet sniffers can be used to inspect network packets for suspicious or anomalous behaviour, which may indicate the presence of network attacks.

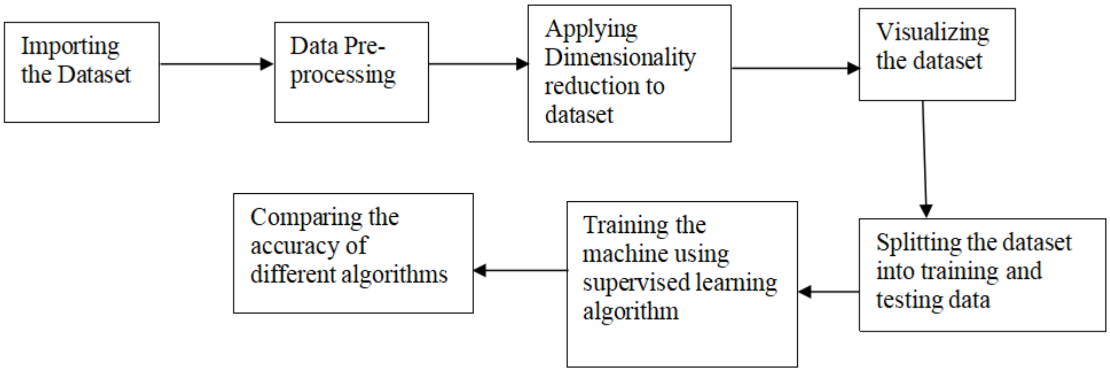
# 4 Proposed System

Our proposed system leverages a Random Forest algorithm, a well-suited choice for its robustness and ability to handle imbalanced datasets common in network security. The model is trained on a vast dataset encompassing diverse normal and attack traffic patterns, ensuring generalizability. We perform meticulous feature engineering to extract informative features from the data, empowering the model for accurate classification. To facilitate real-time monitoring and attack detection, a user-friendly graphical user interface (GUI) is developed. This interface seamlessly integrates with the model, displaying real-time traffic classification (normal or malicious). The system can be further enhanced by incorporating model explainability features to improve user trust and visualization tools to provide insights into network traffic patterns. Additionally, allowing for alert customization empowers users to tailor the system to their specific security needs.

In this project we classify different types of network attacks present in the dataset using supervised machine learning algorithms. The primary purpose of the project is to correctly classify the different types of network attacks. In this we have used supervised learning algorithms to classify the network attacks and visualized different attributes of data to see their relationship with attack using various python libraries.

First, we download the dataset. Because the size of data is very large, therefore we apply few dimensionality reduction techniques to reduce the dataset. In this way dataset would be reduced without reduction in the important features i.e. without affecting the efficiency of classification model. We visualize the dataset to see how different features are affecting the network attacks. Then we make a classification model for our dataset using supervised machine learning algorithms such as random forest, SVM and decision tree. We compare the accuracy given by the algorithms and find the most suitable algorithm for our dataset.

## 4.1Methodology

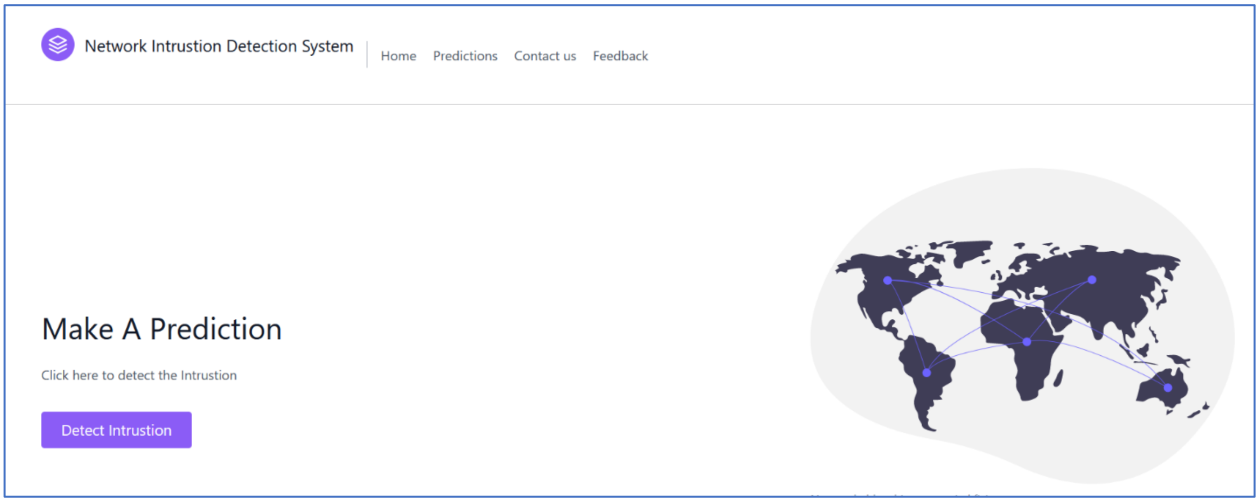

Importing the Dataset
Data Pre-processing
Applying Dimensionality reduction to dataset
Visualizing the dataset
Splitting the dataset into training and testing data
Training the machine using supervised learning algorithm
Comparing the accuracy of different algorithms

At first after importing the dataset, some data pre-processing is done. Then we visualize the data to see relationship between network attacks and different features of the data.

Then, we split the data into training and testing data in the ratio 80:20 and the apply supervised learning algorithms to classify the network attacks and compare between the algorithms to see algorithm giving most accurate classification for given dataset.

## 4.2User Interface

This user-friendly interface empowers anyone to assess network safety. You can simply enter the network's specific fields like packet size, source and destination values, for a more detailed analysis. Clicking a clear "Detect" button initiates the process. The results are then displayed prominently, providing a straightforward verdict like "Safe Network" or "Potential Attack Detected (DoS)" along with a confidence score indicating the model's certainty. This accessible interface empowers everyone to make informed decisions about network security, promoting a safer online environment.



Make A Prediction

Click here to detect the Intrustion

### 4.2.1. Network Data Input:

- Enter the IP address or domain name of the network you want to check.
- Optionally, upload a capture file (pcap) containing the network traffic for more detailed analysis.

### 4.2.2. Scan Button:

- Initiate the attack classification process by clicking a clearly labelled "Scan" button.

### 4.2.3. Result Display:

- Upon completion, the UI displays the classification result in a prominent location.
- Use clear, non-technical language like "Safe Network" or "Potential Attack Detected (DoS)".
- Include a confidence score (percentage) indicating the model's certainty in the classification.

### 4.2.4. Additional Information (Optional):

- For advanced users, provide the specific attack type identified (e.g., DoS, DDoS).
- Briefly explain the attack type in simple terms if a potential attack is detected.

# 5 Experimental Evaluation Analysis

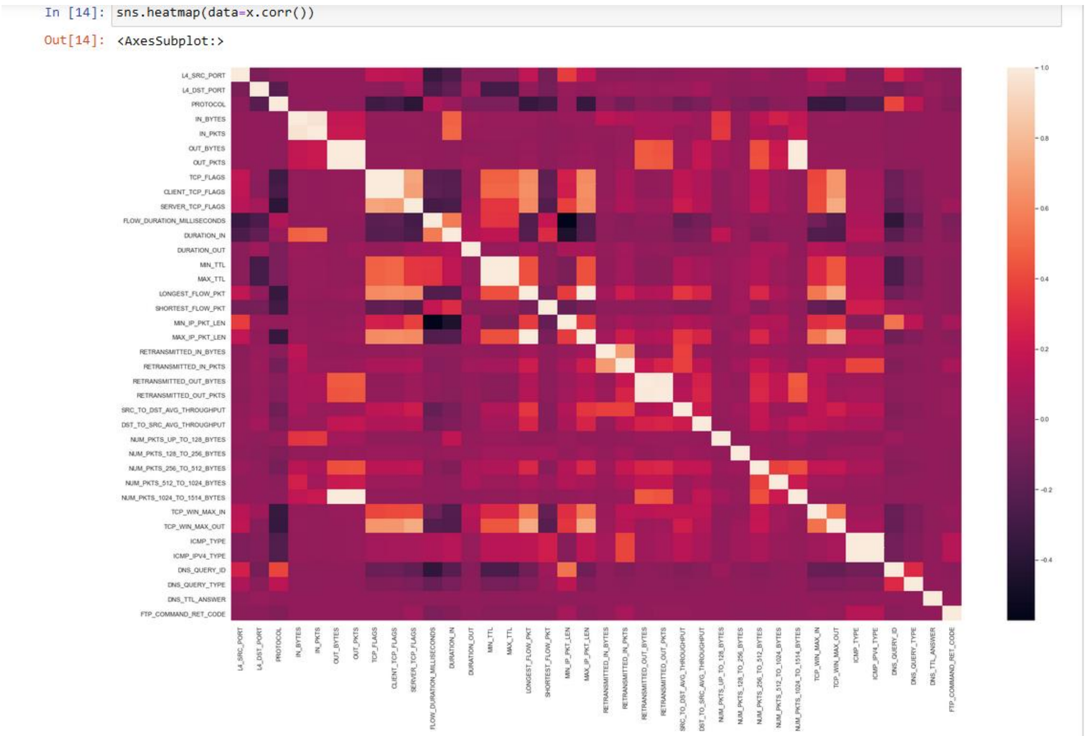
Our investigation into the optimal classification algorithm for this project involved the exploration of several established methods, including Support Vector Machines (SVM), Decision Trees, and Random Forests. Following a rigorous evaluation process, we opted to utilize Random Forests due to their inherent robustness, which significantly benefits our specific application.

### 5.1.1 Importance of Robustness

In the realm of machine learning, robustness refers to an algorithm's capacity to maintain performance in the presence of imperfections within the data. These imperfections can manifest as outliers, noise, or even missing values. For our project, achieving robustness was paramount because the data we employed might contain outliers that could significantly skew the results of more sensitive algorithms. The inherent noise often present in real-world data collection processes could negatively impact the performance of certain algorithms. Our data might have missing values, and a robust method is required to handle them effectively.

## 5.2 Contrasting Random Forest with Other Algorithms

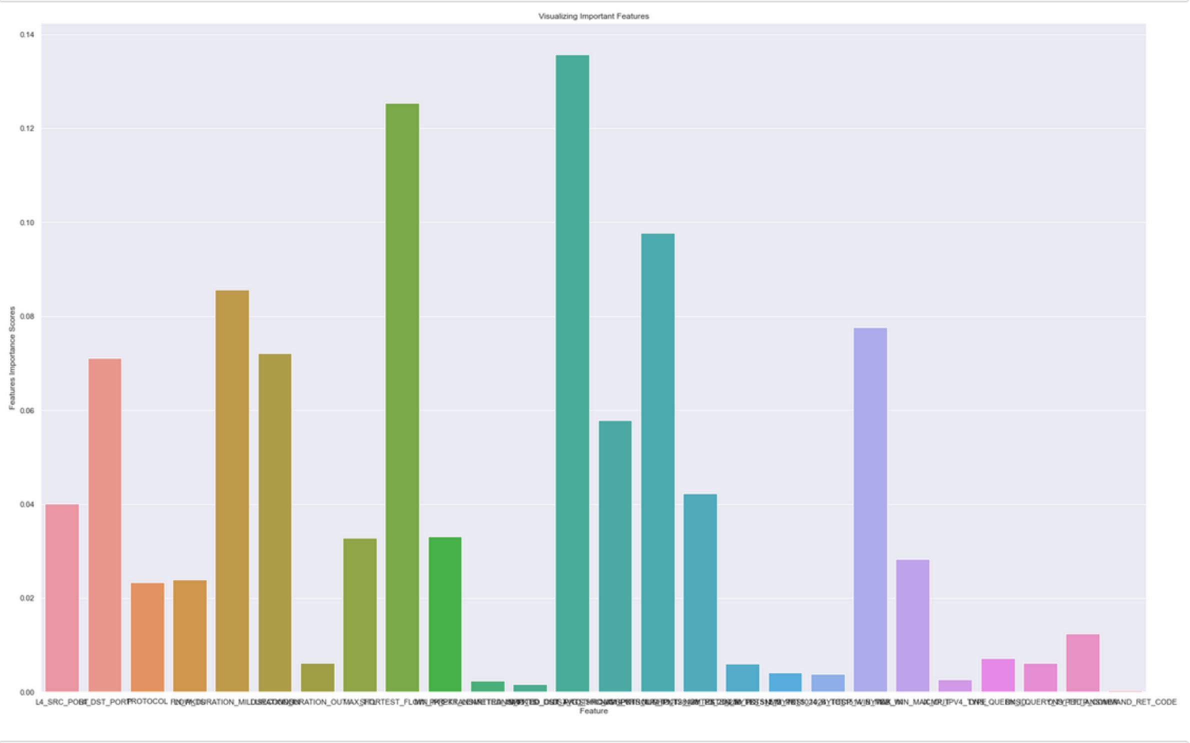
While both SVM and Decision Trees are powerful classification tools, they can be susceptible to outliers. A single outlier has the potential to considerably alter the decision boundary in an SVM or the branching structure of a Decision Tree. Random Forests, on the other hand, effectively alleviate this vulnerability through their averaging mechanism.



```
In [14]: sns.heatmap(data=x.corr())

Out[14]: <AxesSubplot:>
```

Random Forest stands out as a powerful tool for network attack classification due to its exceptional strengths. First, it achieves high accuracy in attack detection, effectively distinguishing between normal traffic and various attack types. This robustness stems from its ability to handle complex, high-dimensional network traffic data. Additionally, Random Forest's ensemble nature combats overfitting, a common pitfall in machine learning. By training multiple decision trees and combining their predictions, it avoids overspecializing on the training data and generalizes better to unseen traffic patterns. Furthermore, Random Forest offers valuable interpretability through feature importance scores. These scores highlight which features in the network traffic data are most critical for attack classification, providing insights to network security analysts. Finally, Random Forest boasts scalability and the ability to handle missing values, making it efficient for processing large datasets and less sensitive to data cleaning imperfections. In conclusion, the combination of high accuracy, robustness, interpretability, scalability, and tolerance for missing data makes Random Forest a compelling choice for network attack classification projects. Support Vector Machines (SVMs) require careful parameter tuning to achieve optimal performance, and their decision boundaries can be difficult to understand. Additionally, imbalanced datasets, a common challenge in network security, can trip up SVMs. Decision trees can overfit the training data if not carefully pruned, and their accuracy might fall short compared to Random Forests or SVMs on complex network traffic datasets. K-Nearest Neighbors (KNN) performance suffers in high-dimensional scenarios like network traffic analysis. They're also sensitive to outliers in the data, and classifying new data points requires comparing them to all training data, making it computationally expensive.



Therefore, due to the potential presence of imperfections in your network traffic data, Random Forests were chosen for their inherent robustness. This characteristic ensures the system maintains performance even with noisy or incomplete data, leading to more reliable network attack classification.

## CONCLUSION

This Project addressed the limitations of existing Network Intrusion Detection Systems (NIDS) by proposing a novel approach that leverages machine learning for robust and adaptive network attack classification. Traditional methods often struggle with evolving attack landscapes and limited ability to detect novel threats. The proposed system, built on a Random Forest algorithm, offers several advantages: **Adaptability:** Machine learning allows the system to continuously learn and improve with new data, effectively addressing dynamic threat patterns. **Robustness:** Random Forests handle noisy or imperfect data, a common challenge in network traffic analysis, leading to more reliable attack classification **Generalizability:** The model can effectively identify novel attacks not encountered before in the training data.

A user-friendly interface facilitates real-time traffic monitoring and attack detection. The system holds promise for enhancing network security posture by providing a more robust and effective solution for network intrusion detection.

## REFERENCES

DATASET:
https://www.kaggle.com/datasets/aryashah2k/nfuqnidsv2-network-intrusion-detection-dataset
Websites: -
https://seaborn.pydata.org/
https://matplotlib.org/
https://pypi.org/project/pydotplus/
https://analyticsindiamag.com/hands-on-guide-to-graphviz-python-tool-to-define-and-visualize-graphs/
https://www.javatpoint.com/dimensionality-reduction-technique
https://pypi.org/project/pydotplus/