CS 584 - ML - Spring 24’

# Enhancing Network Security Through Machine Learning-Based Intrusion Detection

Presented By:

Mohan Babu Kunchala - A20524765

Sanjitha Reddy Pathuri - A20524383

Sirisha Gandham - A20544789

# Outline

1. Introduction
2. Dataset Overview
3. Methodology
4. Data Preprocessing
5. EDA & Feature Selection
6. Model Selection
7. Results
8. Conclusion & Future Study
9. References & Libraries

# Introduction

- ❑ Its more crucial than ever to safeguard our personal and professional information from cyber threats in the current digital era. Network Intrusion Detection is a crucial component of cybersecurity.

- ❑ Advanced attacks can no longer be prevented with only standard security measures. Using the UNSW-NB15 dataset, this project seeks to enhance network security through machine learning techniques. The project attempts to address the shortcomings of conventional security techniques by utilizing machine learning models such as neural networks including MLP classifiers, CNNs and RCNNs.

- ❑ It will also investigate feature selection techniques to improve the performance of these classifiers. The project aims at creating a strong intrusion detection system that can even recognize sophisticated attacks through extensive testing and comparison with traditional classifiers.

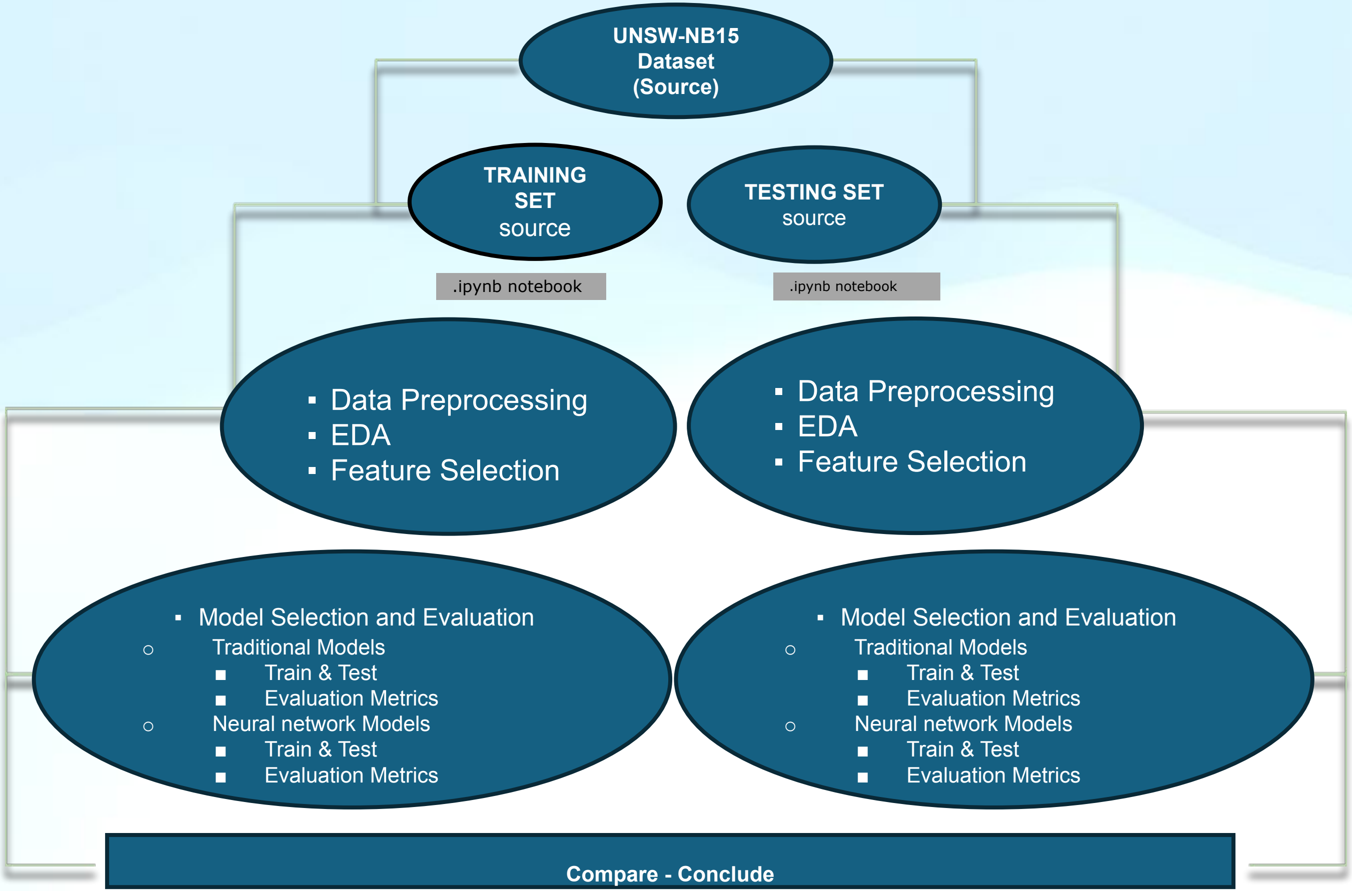
# Why UNSW-NB15 ?

- UNSW-NB15 dataset is popular and widely used for network intrusion detection research.
- Contains large and diverse set of network traffic data (normal and attack traffic) designed to simulate real-world traffic.
- Provides detailed features for feature engineering and model building.
- Ideal for creating and evaluating machine learning-based methods for detecting network intrusions.

# UNSW-NB15 Dataset - Overview

- The UNSW-NB15 dataset is a network traffic dataset that contains labeled traffic data generated by a network simulator. The data contains a total of 49 features, which include nominal, integer, float, timestamp, and binary types.
- The dataset used has 45 features out of 49 features extracted from each network connection and includes both benign and malicious traffic.
- A training set and a testing set were created from the UNSW-NB15 dataset.
    - A total of 175,341 records are taken from the training set and divided into 70-30 records for the test.
    - A total of 82,332 records from the testing set are extracted and divided into 70-30 records for training and testing.
- The dataset is labelled as 0 and 1, where 0 indicates Normal (No Attack) and 1 indicates Attack (Any of the 9 Categories).
- The attack_cat attribute lists the various types of attacks within the nine categories.
- There are nine categories in this data set: e. g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms

# Methodology



**UNSW-NB15 Dataset (Source)**

**TRAINING SET** source

.ipynb notebook

**TESTING SET** source

.ipynb notebook

- Data Preprocessing
- EDA
- Feature Selection

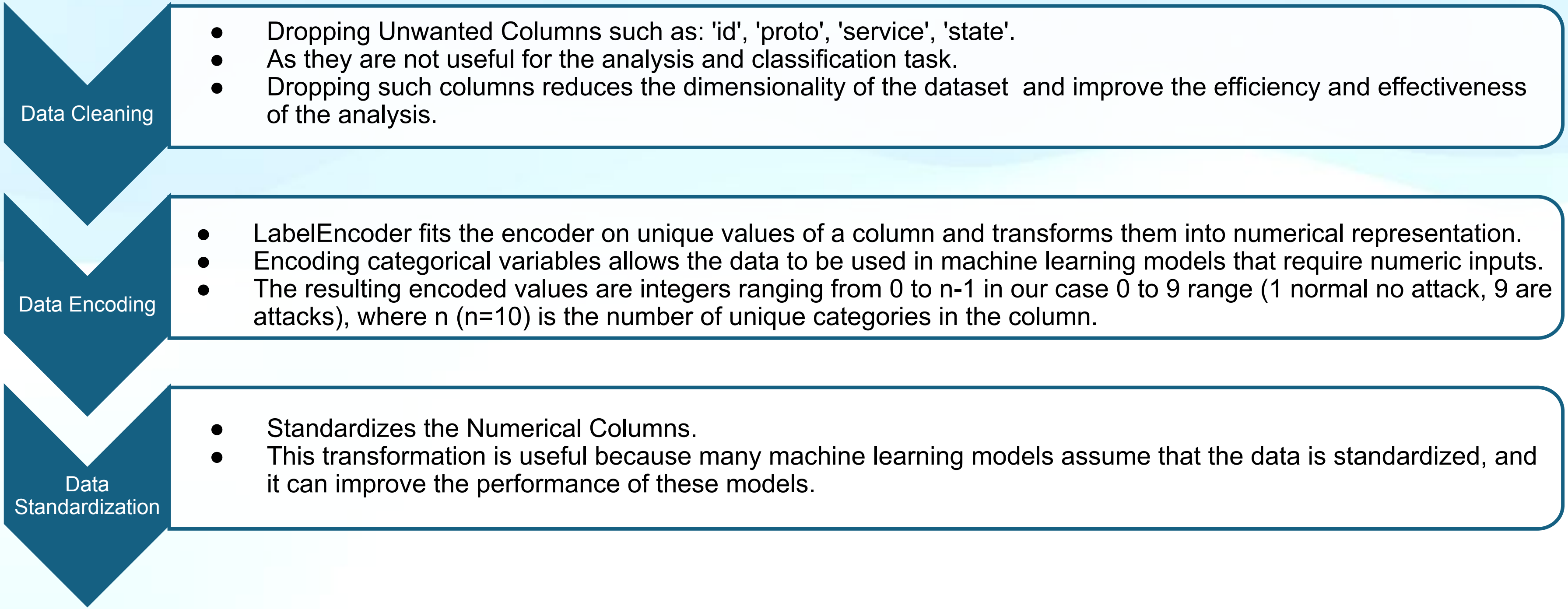
- Data Preprocessing
- EDA
- Feature Selection

- Model Selection and Evaluation
  - Traditional Models
    - Train & Test
    - Evaluation Metrics
  - Neural network Models
    - Train & Test
    - Evaluation Metrics

- Model Selection and Evaluation
  - Traditional Models
    - Train & Test
    - Evaluation Metrics
  - Neural network Models
    - Train & Test
    - Evaluation Metrics

**Compare - Conclude**

# Data Preprocessing



Data Cleaning

- Dropping Unwanted Columns such as: 'id', 'proto', 'service', 'state'.
- As they are not useful for the analysis and classification task.
- Dropping such columns reduces the dimensionality of the dataset  and improve the efficiency and effectiveness of the analysis.

Data Encoding

- LabelEncoder fits the encoder on unique values of a column and transforms them into numerical representation.
- Encoding categorical variables allows the data to be used in machine learning models that require numeric inputs.
- The resulting encoded values are integers ranging from 0 to n-1 in our case 0 to 9 range (1 normal no attack, 9 are attacks), where n (n=10) is the number of unique categories in the column.

Data Standardization

- Standardizes the Numerical Columns.
- This transformation is useful because many machine learning models assume that the data is standardized, and it can improve the performance of these models.

# EDA - Exploratory Data Analysis

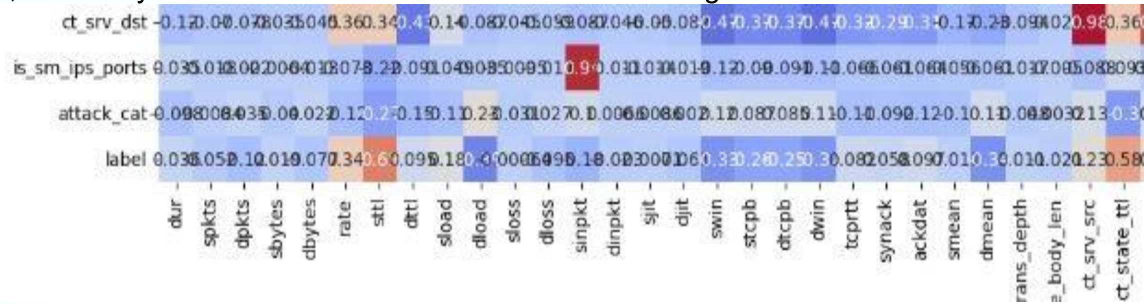
EDA is essential to understand the distribution of the features, correlation among the features, and to identify any outliers in the data.

In accordance to our plan we have performed EDA using visualizations such as histograms, heatmaps, and boxplots.

- Histograms of numerical variables: Shows the distribution of values for each numerical variable in the dataset.
- Heatmap of correlation matrix: Visualizes the degree of correlation between different variables in the dataset using a heatmap.

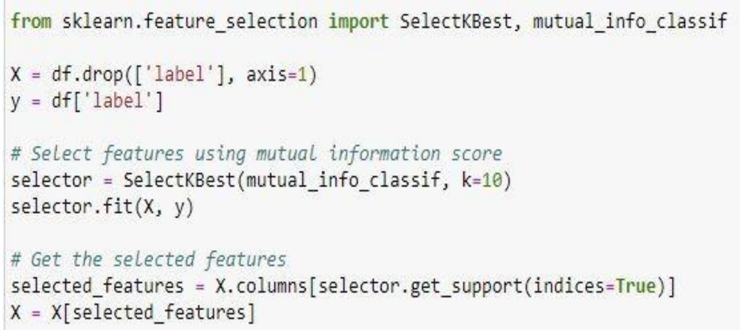
For Ex: The high correlation (0.98) between the features *ct_srv_src* and *ct_srv_dst* in the UNSW-NB15 dataset is due to their similar behavior - the number of connections with the same source is related to the number of connections with the same destination, resulting in a strong linear relationship

Box plots of numerical variables: Creates box plots for each numerical variable in the dataset, grouped by the different attack categories, to identify differences in the distributions across categories.



# Feature Selection

- We perform the feature selection on the dataset using the SelectKBest and mutual information score.
- During fitting, mutual information scores are calculated for each feature with respect to the target variable.
- The SelectKBest object selects the k features (k = 10) with the highest mutual information scores.
- The mutual information criterion measures the amount of information shared between the feature and target variables.
- This method is useful for feature selection when there is a non-linear relationship between the features and target variable or when there are complex interactions among the features.
- The selected features are assigned to the names of the selected features using the get_support() method of the selector object.



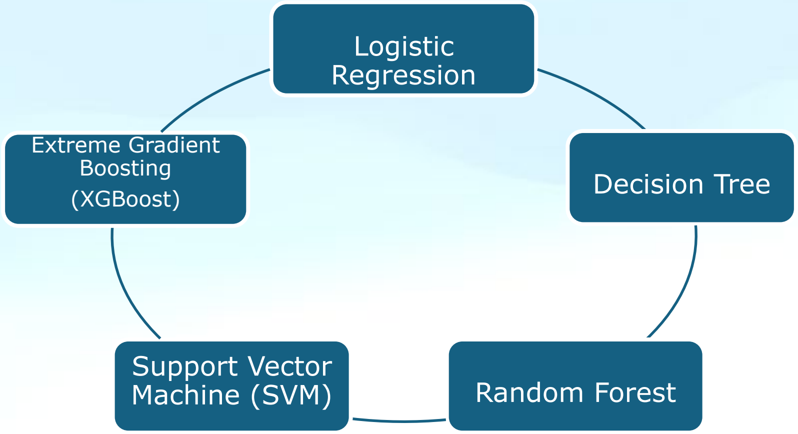
```
from sklearn.feature_selection import SelectKBest, mutual_info_classif

X = df.drop(['label'], axis=1)
y = df['label']

# Select features using mutual information score
selector = SelectKBest(mutual_info_classif, k=10)
selector.fit(X, y)

# Get the selected features
selected_features = X.columns[selector.get_support(indices=True)]
X = X[selected_features]
```

# Supervised Models / Traditional Classifiers



Logistic Regression

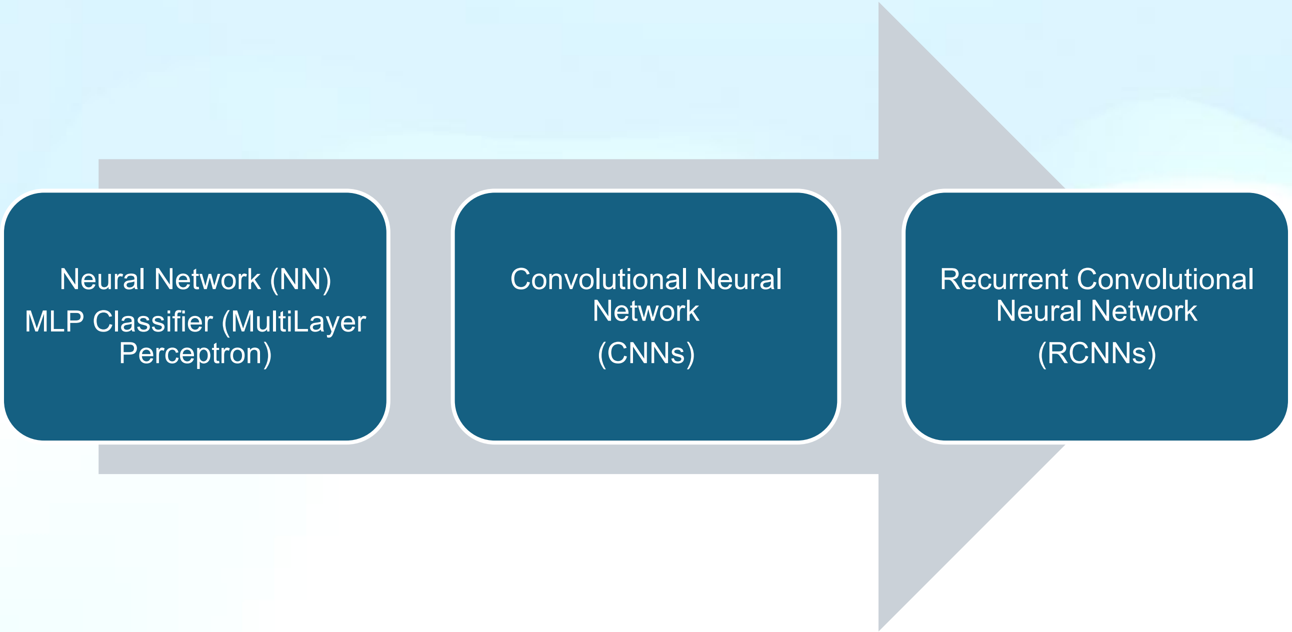
Decision Tree

Random Forest

Support Vector Machine (SVM)

Extreme Gradient Boosting (XGBoost)

# Neural Network Models



Neural Network (NN)

MLP Classifier (MultiLayer Perceptron)

Convolutional Neural Network

(CNNs)

Recurrent Convolutional Neural Network

(RCNNs)

# Results

| Traditional Models | | Logistic Regression | Decision Tree | SVM | XGBoost | Random Forest |
|---|---|---|---|---|---|---|
| TRAINING SET | Accuracy: | 0.89639 | 1.0 | 0.99925 | 1.0 | 1.0 |
| | F1-Score: | 0.90 | 1.00 | 1.00 | 1.00 | 1.00 |
| TESTING SET | Accuracy: | 0.93121 | 1.0 | 0.99902 | 1.0 | 1.0 |
| | F1-Score: | 0.93 | 1.00 | 1.00 | 1.00 | 1.00 |

| Neural Network Models | | NN (MLP Classifier) | CNN | RCNN |
|---|---|---|---|---|
| TRAINING SET | Accuracy: | 0.99998 | 0.99918 | 0.99929 |
| | F1-Score: | 1.00 | 1.00 | 1.00 |
| TESTING SET | Accuracy: | 0.99995 | 0.99740 | 0.99740 |
| | F1-Score: | 1.00 | 1.00 | 1.00 |

# Conclusion & Future Study

- Proposed approach for Network Intrusion Detection using Machine Learning was successful.
- Neural Networks, especially RCNNs, outperformed traditional classifiers in detecting network intrusions.
- Project's main contribution: Exploration of different types of Neural Networks and use of Feature Selection methods.
- Future Study: Explore more advanced deep learning models, such as Generative Adversarial Networks (GANs), etc.. and integrate proposed approach into a real-time network intrusion detection system.

# References & Libraries

1 Sourav Mukherjee, Ananda Roy Chowdhury, Shukla Das, Sayan Chakraborty, and Mita Nasipuri. A comparative study of deep learning approaches for network intrusion detection. Future Generation Computer Systems, 107:1063–1077, 2020.

2 Md Rafiul Islam and Kazi Mohammed Ahmed. Machine learning approaches for network intrusion detection: A comprehensive survey. IEEE Access, 7:27459–27484, 2019.

3 Ahmed Moustafa, Jill Slay, and Gregory Creech. Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In Military Communications and Information Systems Conference (MilCIS), pages 1–6. IEEE, 2015.

4 Nour Moustafa and Jill Slay. The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 dataset and the comparison with the kdd99 dataset. Information Security Journal: A Global Perspective, 25(1-3):1–14, 2016.

5 Mamoun Alazab, Michael Hobbs, and Jemal Abawajy. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 60:84–102, 2016.

*LIBRARIES* Used: *NUMPY, MATPLOTLIB, SEABORN, XGBOOST, SKLEARN, and* KERAS.

# THANK YOU