

Network Intrusion Detection System Using ML

Group 16

Kosmika Saratkar - 22B2163
Anurag Deshpande - 21D110001
Mallekhedi Vijaya Krishna - 210040087
Pratham Tarjule - 20D110029
K S Varun - 200100088

Objective

We aim to build a Machine Learning based hybrid **Network Intrusion Detection System** (NIDS) model for businesses and firms which rely on computer networks. Our hybrid model will use attack pattern recognition for early detection and mitigation while anomaly detection can identify new attacks. The goal is to detect and classify different types of network attacks based on the input features provided in the dataset. The dataset used for training and testing contains various network-related features such as duration, protocol type, service, and flag.



Model Details

- **Training Dataset used** - KDD Cup 1999
<https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data>
- **Size of data:**
 - **494021** rows, **43** columns (features)
- **Train-test split:**
 - Training data: 80%
 - Testing data: 20%
- Features are separated into **numerical** and **categorical** subsets.
Numerical features are standardized using the *StandardScaler* function in *scikit-learn* to ensure faster convergence during training.
- **Target encoding:** The target variable is one-hot encoded to create a binary matrix representing the attack types.



Model Details

- **Architecture:**

A neural network model is designed using the *Keras* library. The model includes an input layer for numerical features and embedding layers for categorical features. Two hidden layers with 64 and 32 neurons, respectively, are followed by the output layer with softmax activation for multi-class classification.

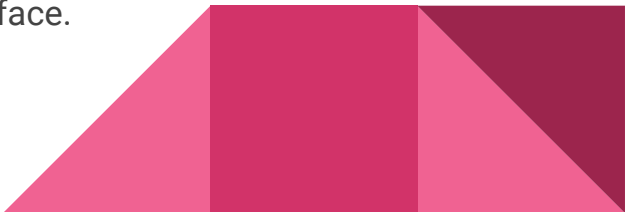
- **Training:**

The model is compiled using the *Adam* optimizer and binary cross-entropy loss function. Training is performed for 10 epochs with a batch size of 128.

- **Evaluation:**

The training history, which includes training loss and validation loss for each epoch, is visualized using Matplotlib, to monitor the model's performance.

- Additionally, a **Gradio interface** is implemented for real-time predictions using CSV files. The *predict_csv()* function preprocesses the uploaded CSV file, prepares the data for prediction, and utilizes the trained model to generate predictions. The results are then displayed in the Gradio interface.



Model Training and Testing

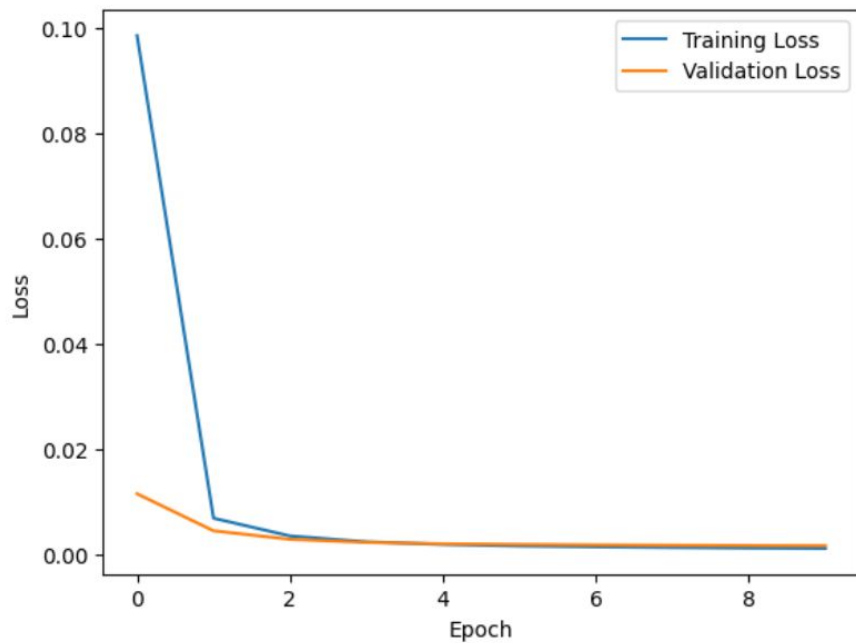


Figure: Loss vs Epochs

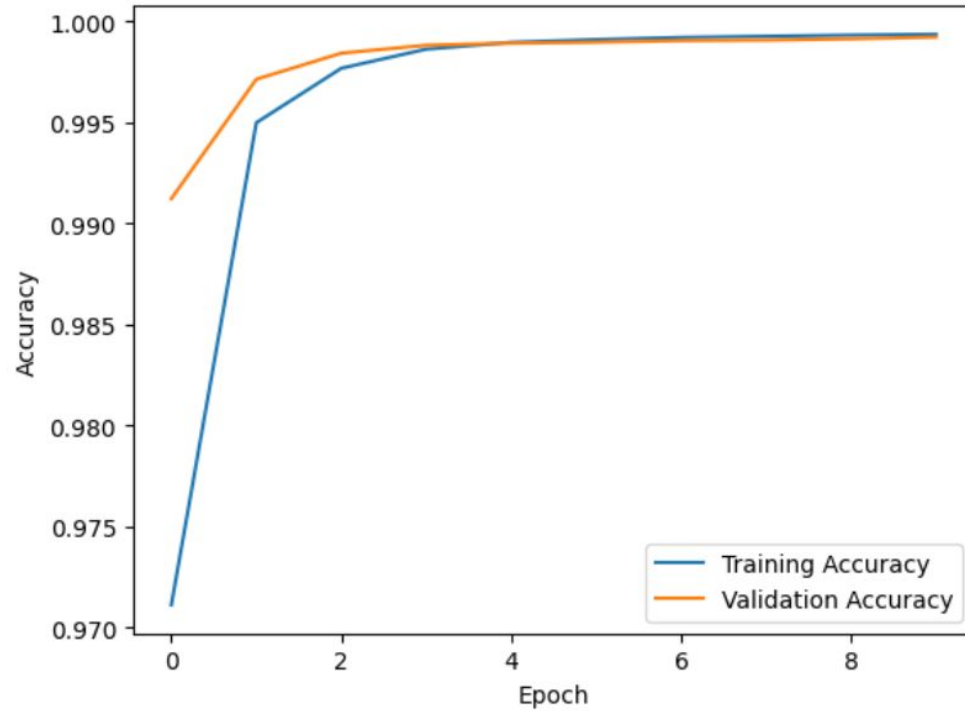


Figure: Accuracy vs Epochs

Usage Guide

1. Launch the Gradio Interface:

- Execute the provided code in your Python environment.
- The Gradio interface will be launched, providing a user-friendly platform for making predictions.


2. Upload a CSV File:

- In the Gradio interface, locate the file upload section.
- Click on the file upload button to select a CSV file from your local machine.
- The selected file will be processed by the `predict_csv()` function for intrusion detection predictions.

3. Prediction Results:

- After uploading the CSV file, the model will make predictions based on the network intrusion detection features.
- The results, including the prediction statement and attack type (e.g., "Network Intrusion Detected, Attack Type: r2l"), will be displayed in the Gradio interface.

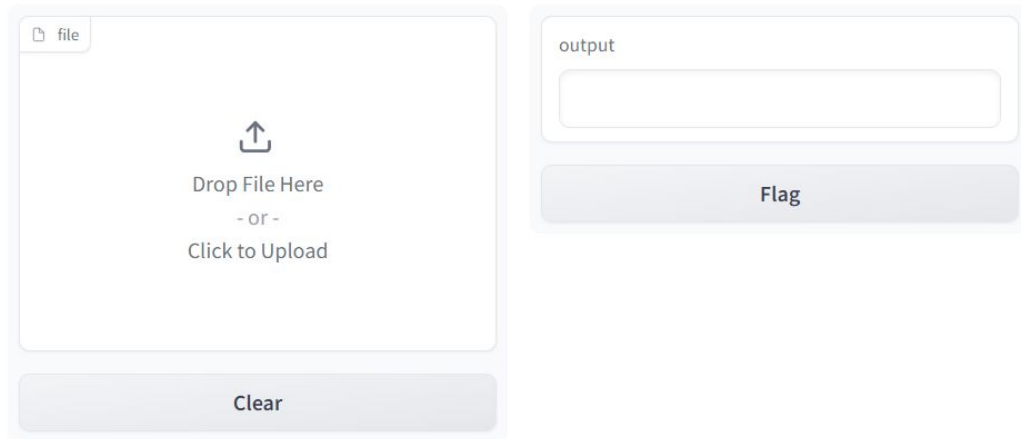
4. Interpretation of Results:

- The prediction statement indicates whether a network intrusion is detected or not.
 - The attack type provides additional information about the detected intrusion, such as the type of attack (e.g., "r2l,u2r").
- 

Screenshots: user interface and output visualization

Network Intrusion Detection System

Upload a CSV file, and the model will detect whether there is any network intrusion or not and the type of attack.



The screenshot displays a web-based user interface for a Network Intrusion Detection System. On the left, a file upload area is labeled 'file' in the top-left corner. It features a large upward-pointing arrow icon and the text 'Drop File Here - or - Click to Upload'. Below this area is a 'Clear' button. On the right, there is an 'output' label above a text input field. Below the input field is a 'Flag' button. The interface is clean and modern, with a light gray background and rounded corners.

Use via API  - Built with Gradio 

Output Visualization

Network Intrusion Detection System

Upload a CSV file, and the model will detect whether there is any network intrusion or not and the type of attack.

file

kddcup.data_10_percent.gz2.0 MB↓

Clear

output

Network Intrusion Detected, Attack Type: r2l

Flag

Links

Google Colab link -

https://colab.research.google.com/drive/1UK0zlerG6GpSnu5QwxJzAh6qC7_J0Kqh?usp=sharing

