ABESEC Ghaziabad
Department of Computer Science & Engineering

Project Title:
Network Traffic analyzer using Deep Learning

Semester: 5th				
Project Type = Cyber Security Project				
	Name	**Roll Number**	**Section**	**Signature**
Group member (1)	**Krishan Tiwari**	**2200320100088**	**B**	
Group member (1)	**Kshitiz Singh**	**2200320100090**	**C**	
Project Guide	**Dr. Devendra Kumar Mishra**	**Remarks from Guide:**		
Signature				
Date of submission	**9/10/2024**			
Examiner 1	**Ms. Bharti Shukla**	**Remarks from Examiner:**		
Date of presentation1	**09/10/2024**			

1.1 Problem Introduction

As digital transactions and online activities increase, the risk of fraudulent activities in network traffic and credit card transactions also escalates. Traditional methods of detecting anomalies and fraudulent behavior often fall short due to the high volume of data and the sophisticated techniques used by fraudsters. This project aims to develop a robust system that utilizes deep learning techniques for analyzing network traffic and detecting credit card fraud.

1.1.2 Project Objective

1. Network Traffic Analysis: To analyze network traffic data to identify patterns and anomalies that indicate potential security threats or malicious activities.

1.1.3 Scope of the Project

Data Scope:

- Analyze network traffic data (packet information, timestamps) and credit card transaction data (amounts, merchant details).

Technological Scope:

- Utilize deep learning techniques (CNNs, RNNs, LSTMs) with frameworks like TensorFlow or PyTorch.

Functional Scope:

- Data preprocessing, model training, anomaly detection, performance evaluation, and prototype deployment.

1.2 Related Previous Work

Network Traffic Analysis

1. **CICIDS 2017 Dataset and Analysis**
   - **Project**: The Canadian Institute for Cybersecurity created the CICIDS 2017 dataset, which includes a variety of network traffic scenarios. This dataset has been used in multiple studies for training deep learning models for intrusion detection.
   - **Paper**: "A Comprehensive Review on Network Traffic Analysis for Intrusion Detection Systems" (2020). This paper reviews various machine learning and deep learning techniques applied to network traffic analysis.
2. **Deep Learning-Based Intrusion Detection System**
   - **Paper**: "Deep Learning for Network Intrusion Detection: A Review" (2021). This research reviews the application of deep learning models for network intrusion detection, discussing different architectures and their performance.
3. **LSTM for Network Anomaly Detection**
   - **Paper**: "Network Anomaly Detection Using LSTM Neural Network" (2019). This study presents an LSTM-based approach for detecting anomalies in network traffic, achieving high accuracy in distinguishing between normal and malicious traffic.

1.3 Software and Hardware Requirements

Software Requirements

1. **Programming Languages**:
   - **Python**: Primary programming language for data analysis and model development.
   - **R**: Optional for statistical analysis and data visualization.
2. **Deep Learning Frameworks**:
   - **TensorFlow**: For building and training deep learning models.
   - **Keras**: High-level API for TensorFlow, simplifying model design and training.
   - **PyTorch**: Alternative deep learning framework with dynamic computation graphs.
3. **Data Processing Libraries**:
   - **Pandas**: For data manipulation and analysis.
   - **NumPy**: For numerical computations and handling arrays.
   - **Scikit-learn**: For traditional machine learning algorithms and preprocessing.
4. **Visualization Tools**:
   - **Matplotlib**: For plotting graphs and visualizing data.
   - **Seaborn**: For enhanced statistical data visualization.
5. **Database Management**:
   - **SQLite/MySQL/PostgreSQL**: For storing and retrieving large datasets efficiently.
6. **Development Environment**:
   - **Jupyter Notebook**: For interactive coding and documentation.
   - **Integrated Development Environment (IDE)**: Such as PyCharm or Visual Studio Code.

Hardware Requirements

1. **Computing Resources**:
   - **CPU**: Multi-core processor (Intel i5 or AMD Ryzen 5 or higher) for general data processing.
   - **GPU**: NVIDIA GPU (e.g., GTX 1660 or RTX 2060) for accelerated training of deep learning models, especially for large datasets.
2. **Memory (RAM)**:
   - **Minimum**: 16 GB of RAM for moderate datasets.
   - **Recommended**: 32 GB or more for larger datasets and more complex models.
3. **Storage**:
   - **SSD**: Solid State Drive for faster data access and improved performance.
   - **Minimum Storage**: 500 GB for storing datasets, models, and outputs.
   - **External Backup**: External hard drives or cloud storage solutions for data backup and redundancy.
4. **Network Requirements**:
   - **Internet Connection**: Stable and high-speed internet connection for downloading datasets, libraries, and tools.

1.4 Deliverables

Project Documentation

- **Project Proposal**: A detailed document outlining the project objectives, scope, and methodology.
- **Technical Documentation**: Comprehensive documentation covering system architecture, data preprocessing methods, model development, and deployment processes.

Data Datasets

- **Cleaned Datasets**: Preprocessed datasets for both network traffic and credit card transactions, ready for model training.
- **Feature Sets**: Detailed descriptions of features used in the models, including rationale for selection.

Model Development

- **Trained Deep Learning Models**:
  - Network traffic anomaly detection model.
  - Credit card fraud detection model.
- **Model Architectures**: Diagrams and explanations of the architectures used (e.g., CNNs, LSTMs).

Evaluation Reports

- **Performance Metrics**: Detailed reports on model performance, including accuracy, precision, recall, F1-score, and ROC-AUC for both models.
- **Comparative Analysis**: Summary of how the deep learning models compare to traditional methods.

Implementation Prototype

- **Working Application**: A prototype application capable of real-time network traffic analysis and credit card fraud detection.
- **User Interface**: Basic UI for displaying results, alerts, and analytics (if applicable).

Deployment Framework

- **Deployment Guidelines**: Step-by-step instructions for deploying the models in a real-time environment.
- **Monitoring Tools**: Tools and scripts for monitoring model performance and handling updates.

Presentation Materials

- **Final Presentation**: Slides summarizing the project findings, methodologies, and results for stakeholders.
- **Demo Video**: A short video demonstrating the application’s capabilities and results.

Future Work Recommendations

- **Suggestions for Improvement**: Recommendations for enhancing model performance, scalability, and compliance with regulations.
- **Potential Applications**: Ideas for extending the project into other areas of cybersecurity or fraud detection.

1.7 Gantt Chart

Task	Duration	Timeline
Project Planning	1 week	Week 1
Data Collection	1 week	Week 1-2
Data Preprocessing	1 week	Week 2
Exploratory Data Analysis	1 week	Week 3
Model Development	1 week	Week 4
Develop application prototype	1 week	Week 5
User Interface Development	1 week	Week 5
Testing and Feedback	1 week	Week 6
Final Report and Presentation	1 week	Week 6

1.8 References

Network Traffic Analyzer

1. **Books:**
    - Network Traffic Analysis: A Comprehensive Guide by D. T. D. R. P. G. K. Choudhary
    - Network Security Monitoring by Chris Sanders and Jason Smith
2. **Research Papers:**
    - A Survey on Network Traffic Analysis Techniques - IEEE Xplore
    - Deep Learning for Network Traffic Analysis: A Review - ScienceDirect
3. **Online Resources:**
    - Wireshark User Guide
    - Cisco’s Network Traffic Analysis Tools