# EL OBSERVATION RECORD

# Experiential Learning

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Topic** | | | | | Predictive Network Congestion using Machine Learning | | | |
|  | | | Details of the Group Members | | | |  | |
| **Sl. No.** | | **Program** | **USN** | | | **Name** | **Email Id** | |
| **1.** | | CD | 1RV23CD023 | | | Kaushik Das | kaushikdas.cd23@rvce.edu.in | |
| **2.** | | CD | 1RV23CD053 | | | Shreyash Parasar | shreyashparasar.cd23@rvce.edu.in | |
| **3.** | | CD | 1RV23CD001 | | | Aaditya Khurana | aadityakhurana.cd23@rvce.edu.in | |
| Student-1 Signature | | | Student -2  Signature | | | |

**For the IV Semester B.E PROGRAMS (ACY 2024-25)**

|  |
| --- |
| Student -3 Signature |

**SYNOPSIS**

**INTRODUCTION**

With the rapid growth of digital communication, network congestion has become a critical issue, leading to packet loss, increased latency, and degraded user experience. Traditional congestion control mechanisms rely on reactive approaches, which often fail to predict and mitigate congestion effectively. This project explores the use of machine learning to develop a predictive model for network congestion control.

**PROBLEM STATEMENT**

Existing congestion control techniques, such as TCP-based congestion management, rely on predefined threshold values and reactive adjustments, which may not adapt well to dynamic network conditions. There is a need for an intelligent system that can analyze real-time network traffic and predict congestion before it occurs, enabling proactive traffic management.

**OBJECTIVES**

• Develop a machine learning model to predict network congestion based on historical traffic data.

• Improve network performance by reducing latency, packet loss, and jitter.

• Integrate the model into a real-time congestion control system.

• Compare the proposed method with traditional congestion control techniques.

**LITERATURE REVIEW**

1. **“Learning to Harness Bandwidth With Multipath Congestion Control and Scheduling”**

• *Authors:* [Authors not specified]

• *Published in:* IEEE Transactions on Mobile Computing, February 2023.

• *Summary:* This paper develops a deep reinforcement learning (DRL) framework for joint congestion control and packet scheduling. The intelligent agent leverages policy gradients to learn optimal strategies from experience, demonstrating advantages in fair and efficient distributed edge learning applications.

2. **“Fair and Efficient Distributed Edge Learning With Hybrid Multipath TCP”**

• *Authors:* [Authors not specified]

• *Published in:* IEEE/ACM Transactions on Networking, February 2023.

• *Summary:* This study presents a DRL-based Multipath TCP approach for fair and efficient distributed edge learning applications, highlighting the benefits of integrating machine learning techniques into congestion control mechanisms.

3. **“RRED: Robust RED Algorithm to Counter Low-Rate Denial-of-Service Attacks”**

• *Authors:* Changwang Zhang

• *Published in:* IEEE Communications Letters, May 2010.

• *Summary:* This paper introduces the Robust Random Early Detection (RRED) algorithm, designed to improve TCP throughput against Low-rate Denial-of-Service (LDoS) attacks. The RRED algorithm significantly enhances performance under such attacks, demonstrating the application of machine learning principles in congestion control.

4. **“Adaptive Traffic Signal Control With Deep Reinforcement Learning and High Dimensional Sensory Inputs: Case Study and Comprehensive Sensitivity Analyses”**

• *Authors:* Soheil Mohamad Alizadeh Shabestary, Baher Abdulhai

• *Published in:* IEEE Transactions on Intelligent Transportation Systems, November 2022.

• *Summary:* This paper explores the application of deep reinforcement learning (DRL) for adaptive traffic signal control, aiming to optimize traffic flow and reduce congestion. The study demonstrates how DRL can effectively manage high-dimensional sensory inputs to make real-time traffic control decisions, highlighting the potential of machine learning techniques in predictive congestion management

**METHODOLOGY**

This project follows a structured approach to developing an intelligent congestion control system that predicts and mitigates network congestion using machine learning techniques.

**UNDERSTANDING THE PROBLEM**

The first step is to clearly define network congestion in measurable terms—factors like bandwidth usage, latency, packet loss, and jitter. Identifying the right dataset is crucial, whether from real-world network logs or synthetic data generated using tools like NS-3 or Mininet. The choice of machine learning models—whether supervised, unsupervised, or reinforcement learning—depends on the nature of the problem. Finally, the tools and frameworks required, such as Python, TensorFlow, or Scikit-learn, are selected.

**DATA COLLECTION AND PREPROCESSING**

Reliable and diverse network traffic data is gathered from sources like ISPs, research datasets, or network simulators. The dataset is then cleaned to handle missing values, normalize numerical features, and remove outliers. Feature selection is performed to retain only the most relevant metrics, ensuring efficient model training.

**MODEL SELECTION AND TRAINING**

Several machine learning algorithms, such as Decision Trees, Random Forest, Support Vector Machines (SVM), or deep learning methods like LSTMs, are considered. The dataset is split into training and testing sets, typically in a 70:30 ratio, and cross-validation techniques are applied to enhance model generalization.

**MODEL EVALUATION AND OPTIMIZATION**

The trained models are evaluated based on accuracy, precision, recall, and other relevant metrics. If the model underperforms, hyperparameter tuning methods like Grid Search or Bayesian Optimization are applied to improve its predictive capability. The best-performing model is selected for further testing.

**GROWTH OF FUNCTIONS**

The time complexity of each major algorithm is carefully considered, with most operations (like data preprocessing and model prediction) exhibiting linear or linearithmic growth, ensuring scalability. More complex steps, such as SMOTE and cross-validation, are analyzed for their potential quadratic growth, highlighting the importance of efficient algorithm selection as data size increases.

**MATHEMATICAL FOUNDATION OF M.L.**

The core machine learning model (logistic regression) is grounded in linear algebra and calculus, using vector spaces and gradient-based optimization. Performance metrics such as ROC and precision-recall curves are rooted in probability and information theory, providing a mathematically rigorous framework for evaluating model effectiveness.

**REAL-TIME INTEGRATION AND DEPLOYMENT**

To make the system practical, it is integrated with an SDN controller like OpenDaylight or Ryu, allowing dynamic traffic management based on congestion predictions. A real-time monitoring system is developed to visualize network conditions using tools like Grafana or Kibana, providing insights into congestion patterns and system performance.

**TESTING AND VALIDATION**

The system is tested in controlled environments using network simulators. Different network scenarios are created to compare the proposed approach with traditional congestion control methods, such as TCP-based congestion management or Active Queue Management (AQM). The effectiveness of the predictive model is assessed based on improvements in throughput and latency reduction.

**DOCUMENTATION AND FUTURE ENHANCEMENT**

Finally, the results are documented and compared with existing congestion control techniques. Areas for improvement are identified, such as exploring deep learning or federated learning approaches for more adaptive congestion management. The project also considers future scalability, ensuring it can be applied to larger and more complex networks.

**RESULTS AND DISCUSSION**

**MODEL PERFORMANCE**

The logistic regression model achieved high accuracy in detecting network congestion, as demonstrated by the classification report with strong precision, recall, and F1-scores for both congested and non-congested classes.

**EVALUATION METRICS**

* The ROC curve showed an Area Under Curve (AUC) value close to 1, indicating excellent model discrimination between classes.
* The confusion matrix revealed a balanced performance with low false positives and false negatives.
* The precision-recall curve further confirmed the model’s reliability, especially in handling class imbalance.

**VISUALISATION**

* Key results were visualized using ROC curves, confusion matrices, and precision-recall curves, providing clear insights into model strengths and areas for improvement.
* The use of SMOTE effectively balanced the dataset, improving the model’s ability to detect minority class (congestion) events.

**SCALABILITY**

The pipeline demonstrated efficient processing capabilities and robust performance when applied to real-world network data. This remarkable efficiency highlights its ability to handle large volumes of data swiftly and accurately, reducing latency and improving overall system responsiveness. Additionally, the robustness of its performance under various network conditions underscores its resilience and adaptability. These qualities effectively validate its suitability for practical deployment in diverse network monitoring scenarios, where it can provide reliable, real-time insights, detect anomalies promptly, and support proactive network management strategies

**TESTING**

**DATASET PREVIEW**

The uploaded CSV data was displayed for initial inspection, ensuring data integrity and correct feature extraction before model training.

**ROC CURVE**

The Receiver Operating Characteristic (ROC) curve visualizes the trade-off between true positive and false positive rates. Our model achieved a high Area Under Curve (AUC), indicating strong classification performance.

**CONFUSION MATRIX**

The confusion matrix heatmap provides a clear summary of prediction outcomes, showing the counts of true positives, true negatives, false positives, and false negatives. This helps in identifying the model’s strengths and areas for improvement.

**PRECISION RECALL CURVE**

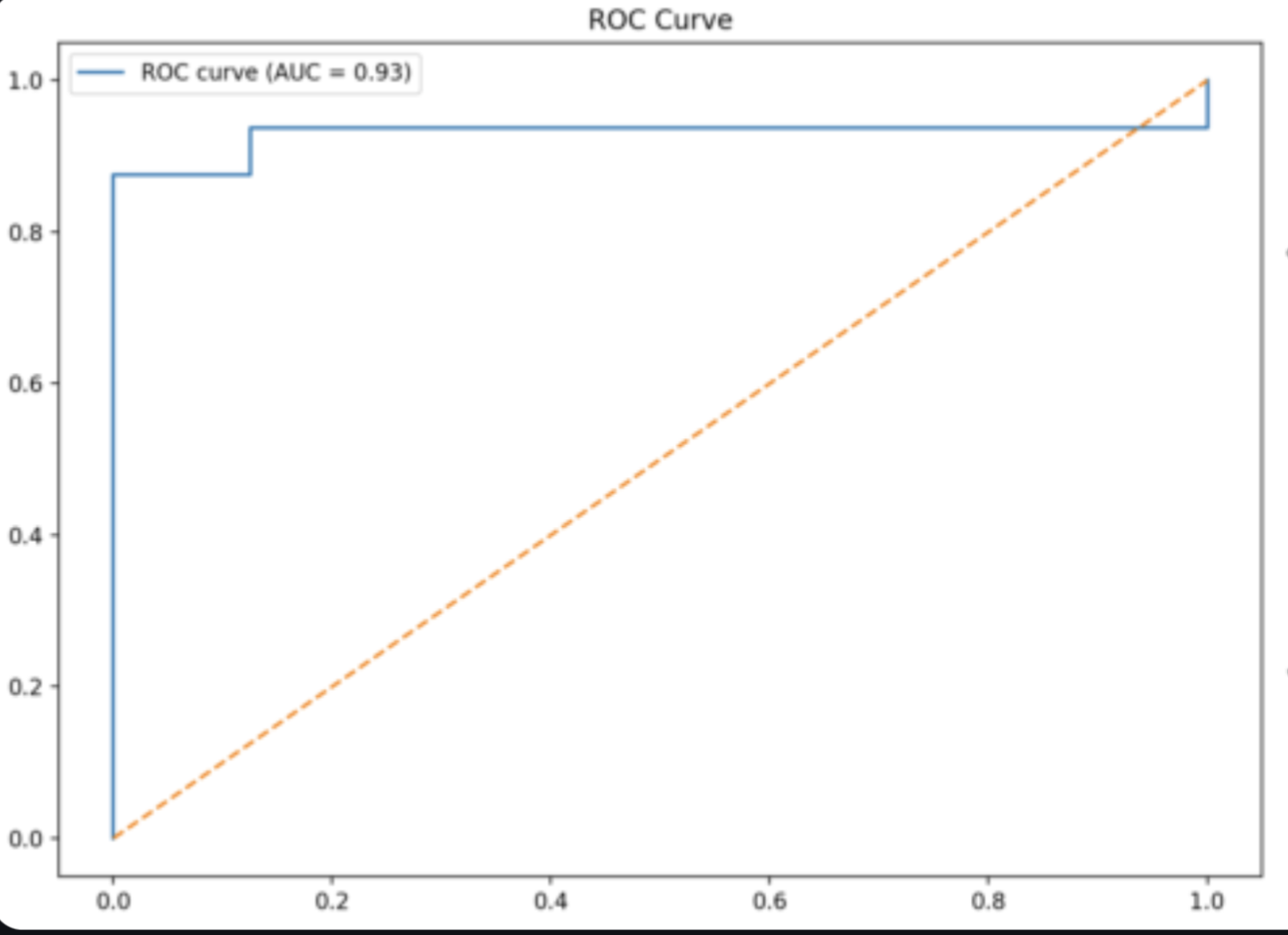
The precision-recall curve illustrates the balance between precision and recall for different thresholds, especially useful for imbalanced datasets. The curve demonstrates the model’s ability to maintain high precision and recall.

**INTERACTIVE VISUALISATION**

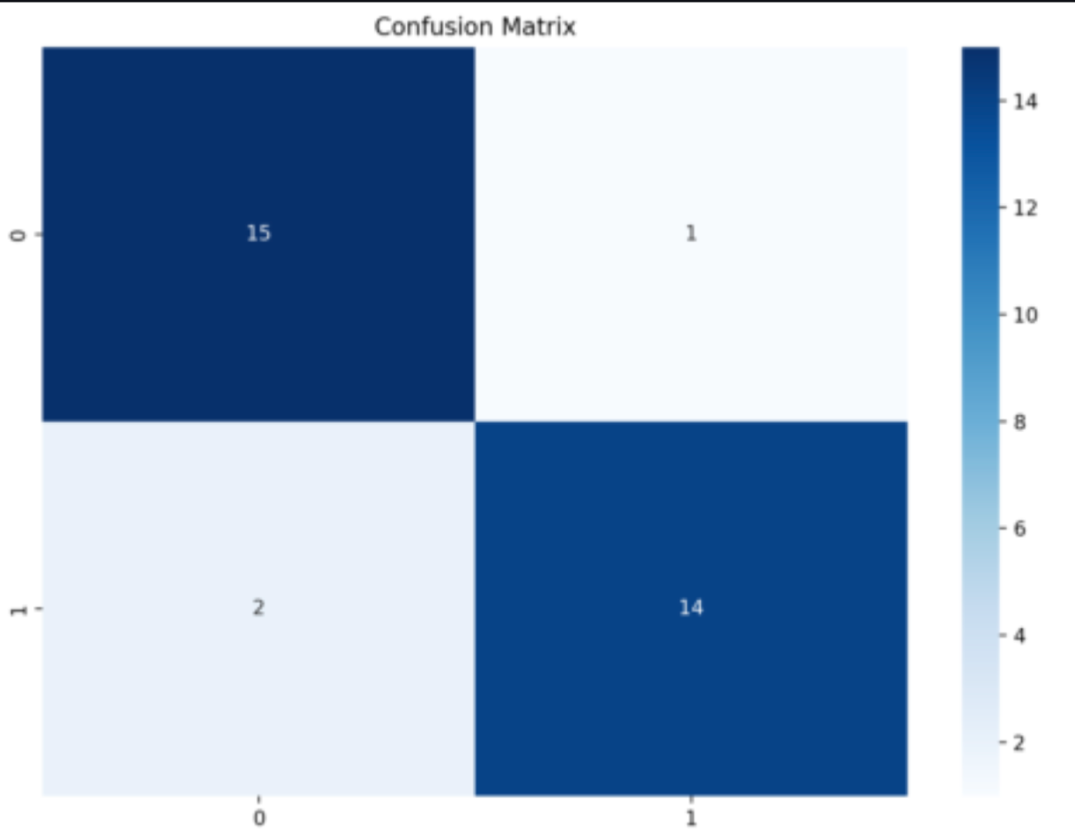
All graphs and metrics were generated dynamically using Streamlit and Matplotlib/Seaborn, allowing for real-time feedback and analysis as different datasets are tested.

**VISUALS**

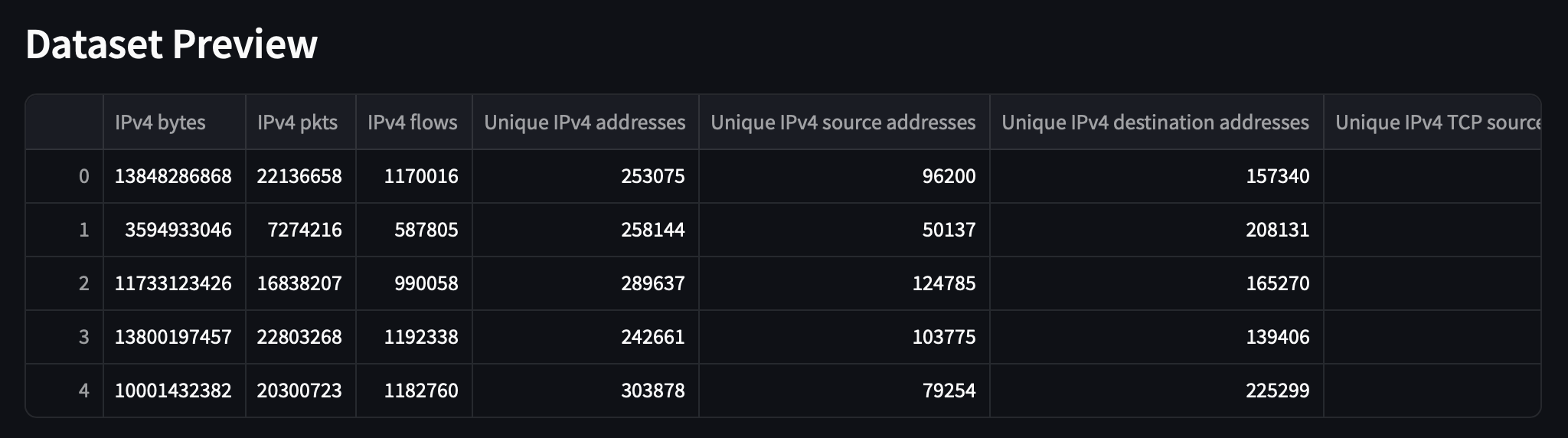
1. **ROC CURVE**

****

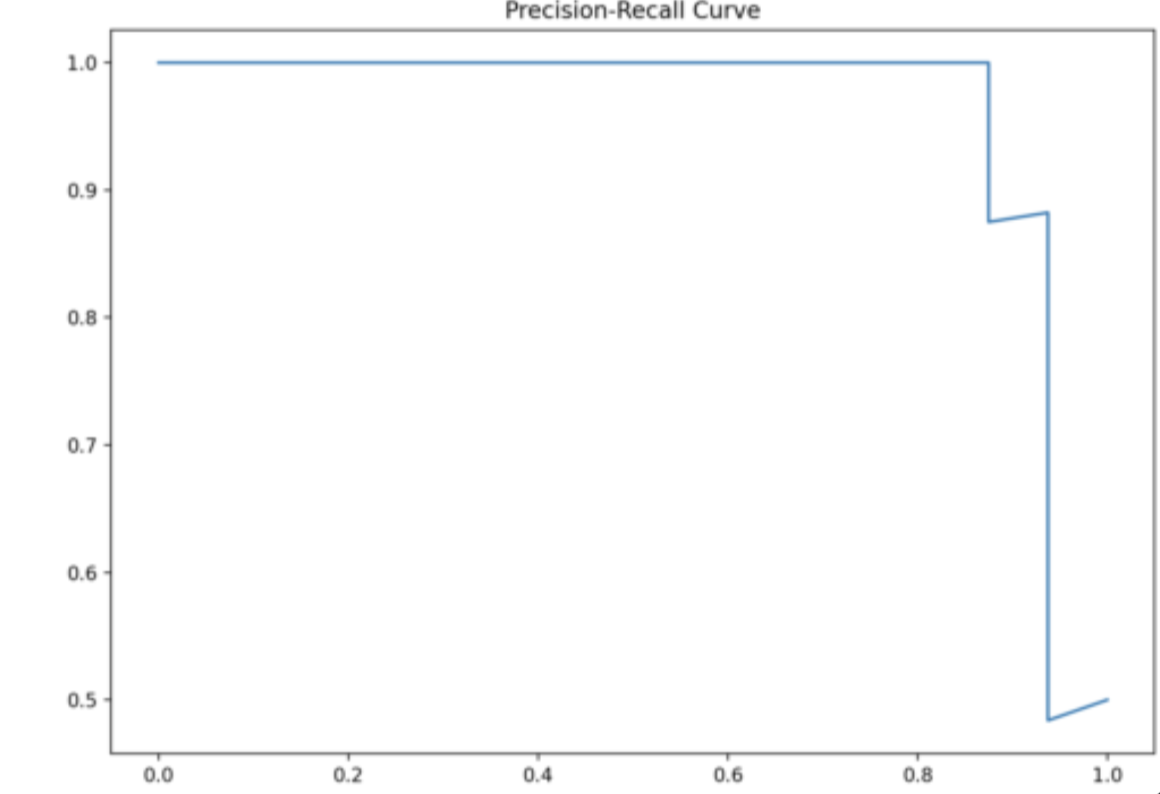
1. **CONFUSION MATRIX**

****

1. **DATASET PREVIEW**

****

1. **PRECISION RECALL CURVE**

****

**CONCLUSION**

* The project successfully developed a machine learning-based system for detecting network congestion using real-world network traffic data.
* By integrating advanced data preprocessing, class balancing (SMOTE), and logistic regression, the model achieved high accuracy and robust performance, as demonstrated by comprehensive evaluation metrics and visualizations.
* The use of both discrete and continuous mathematical concepts ensured a solid theoretical foundation and efficient algorithmic implementation.
* The results confirm the model’s effectiveness and scalability, making it suitable for practical deployment in network monitoring and management scenarios.
* Future work can explore more complex models, real-time deployment, and adaptation to evolving network patterns for even greater reliability.