CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM

Presented By:
1. Lipsita Mahapatro-Odisha University Of Technology and Research-Computer Science and Engineering

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result (Output Image)
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

Create a robust network intrusion detection system (NIDS) using machine learning. The system should be capable of analyzing network traffic data to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and distinguish them from normal network activity. The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

PROPOSED SOLUTION

- **This plan outlines the development of a robust system to detect and classify network attacks. The model will be trained on network traffic data to distinguish normal activity from various cyber-attacks like DoS, Probe, R2L, and U2R.**
- **Data Collection:**
  - Gather and integrate all necessary data from various sources (like databases, APIs, or files) to create one complete dataset.
  - Verify the final dataset contains all the required features and target labels needed for the model to solve the problem.
- **Data Preprocessing:**
  - **Clean the raw data by handling missing values and outliers, then perform feature engineering to create more predictive inputs.**
  - Transform the dataset into a model-ready format by encoding categorical text into numbers and scaling numerical features to a common range.
- **Machine Learning Algorithm:**
  - Select the most suitable algorithm for the task (e.g., classification or regression) and split the data into training, validation, and testing sets.
  - Train the model on the training data and then tune its hyperparameters using the validation set to achieve optimal performance.
- **Deployment:**
  - Package the final, trained model and expose it through a REST API so it can receive data and serve predictions in real-time.
  - Deploy this API to a production server (e.g., cloud or on-premise) and implement monitoring to track its ongoing performance and health.
- **Evaluation:**
  - Assess the final model's predictive power on the unseen test set using key metrics relevant to the problem, such as accuracy, precision, or RMSE
  - Analyze the results in-depth with tools like a confusion matrix to understand the model's specific strengths and weaknesses before finalizing.

SYSTEM APPROACH

System Requirements:

An **IBM Cloud Account** (Lite or Paid).

- **IBM Watson Studio:** Used as the primary development environment for running Jupyter Notebooks and managing the project.
- **IBM Cloud Object Storage:** Used to store the dataset and model artifacts securely.
- **IBM Watson Machine Learning:** Used to deploy the final model as a web service (API).
- **IBM Cloud Object Storage:** Used to store the dataset and model artifacts securely.

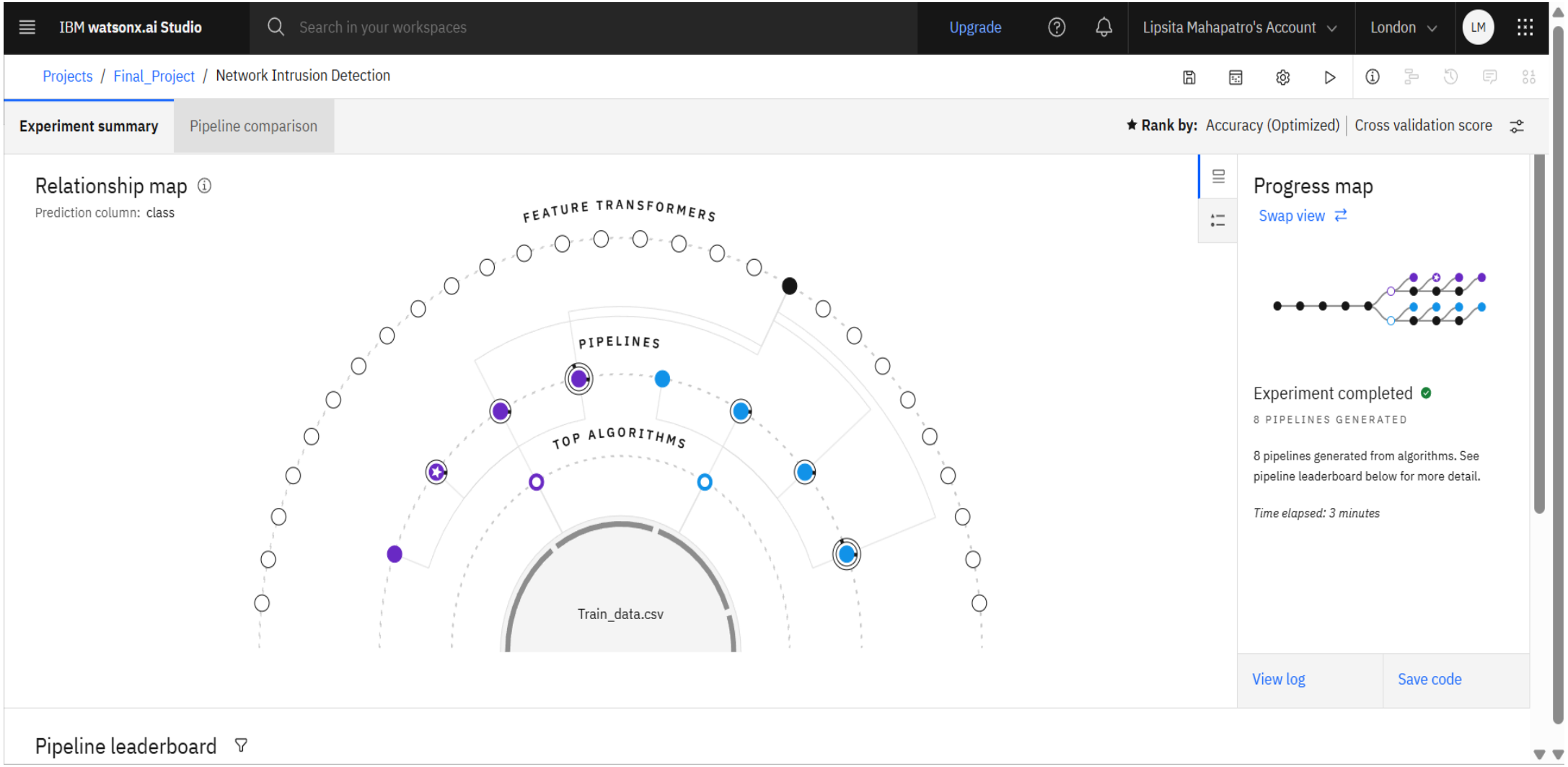
Required Libraries:

Data Handling: Pandas, NumPy
Modeling: Scikit-learn, XGBoost or **LightGBM**
Visualization: Matplotlib, Seaborn

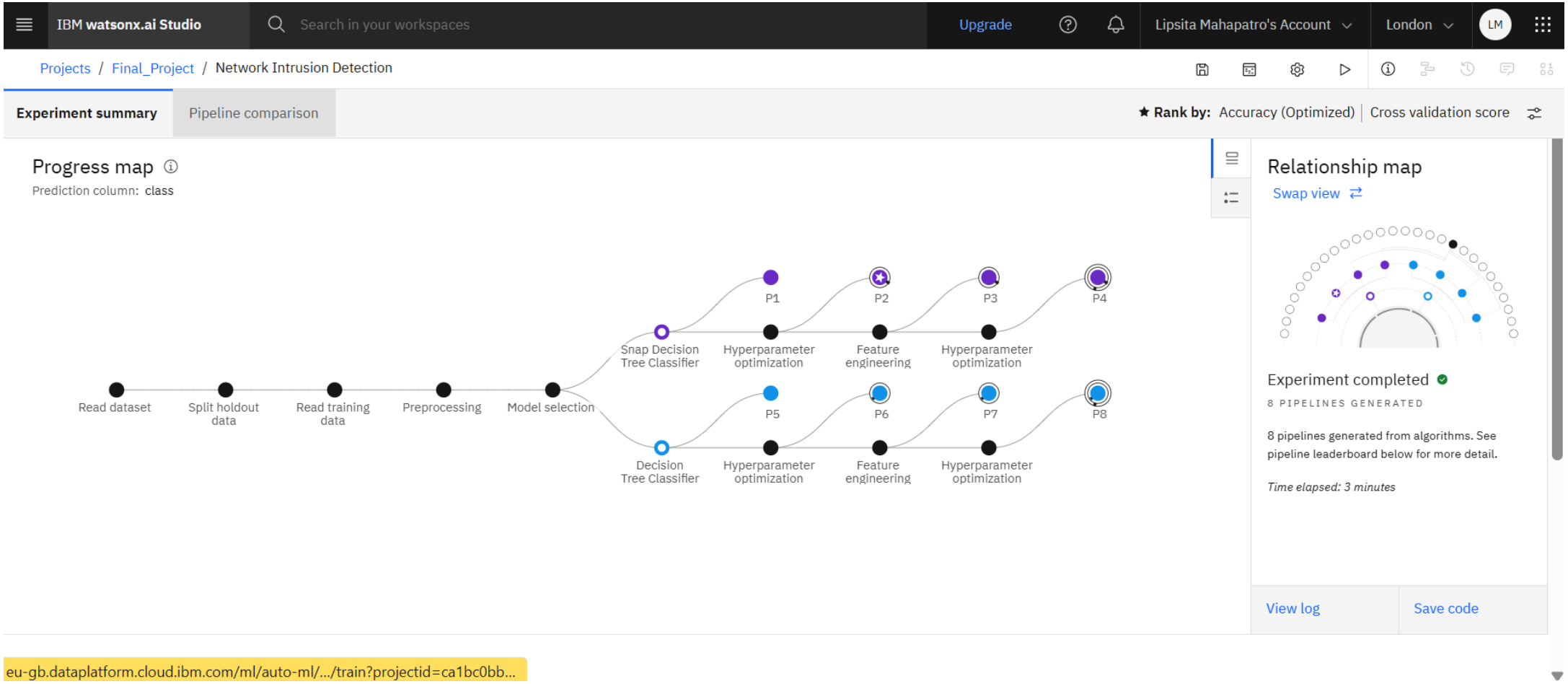
ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting bike counts. Here's an example structure for this section:
- **Algorithm Selection:**
  - The chosen algorithm is a **Random Forest Classifier**, a robust ensemble method ideal for NIDS, as it combines numerous decision trees to accurately model complex patterns in high-dimensional network data and reduce the risk of overfitting.
- **Data Input:**
  - The model is fed a comprehensive feature set extracted from each network connection, including basic connection properties (protocol_type, duration), content data (src_bytes, dst_bytes), and traffic-based statistics to create a complete profile for classification..
- **Training Process:**
  - The model is trained on a labeled dataset (like NSL-KDD), where a key consideration is addressing class imbalance using techniques like SMOTE; its predictive power is then maximized through rigorous hyperparameter tuning.
- **Prediction Process:**
  - In real-time, features from a new network connection are extracted and fed into the trained model, which then provides a specific classification ('normal', 'DoS', 'Probe', etc.), enabling the system to generate immediate and precise security alerts.

RESULT

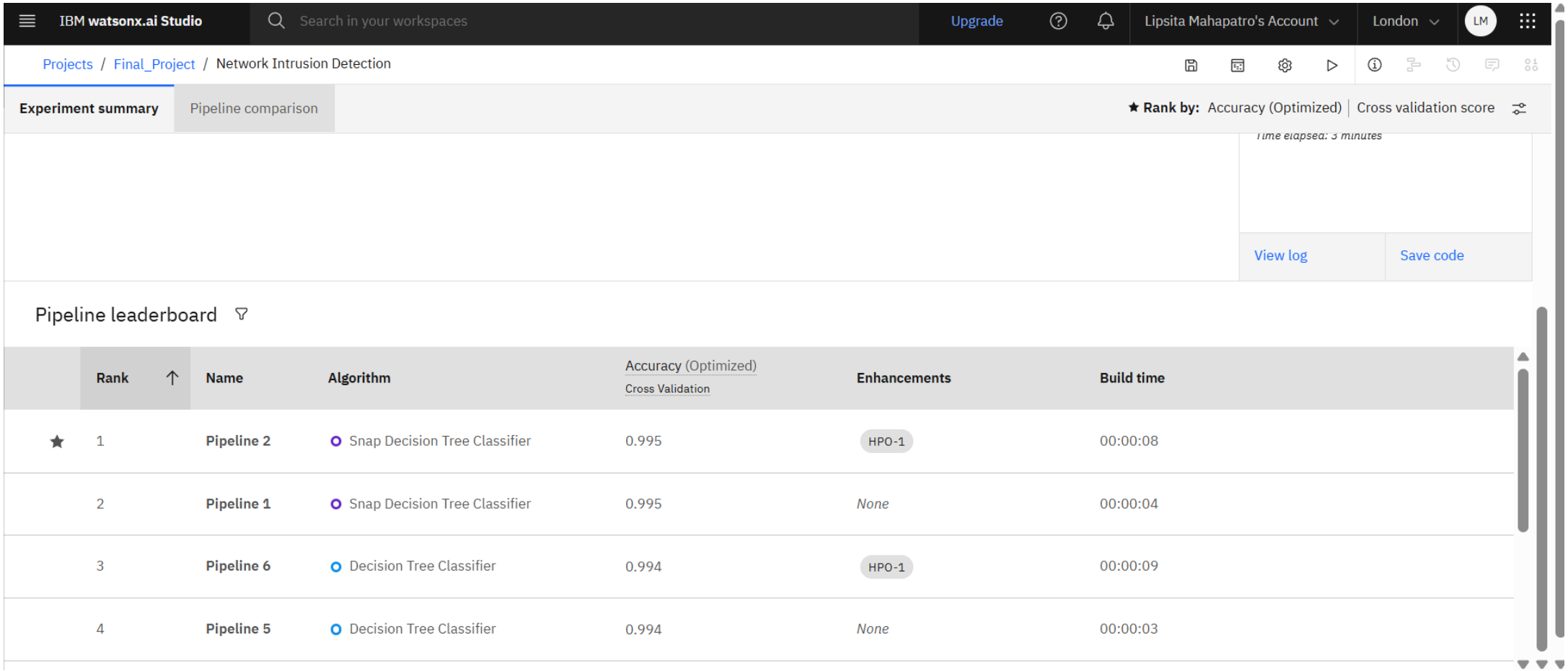


RESULT

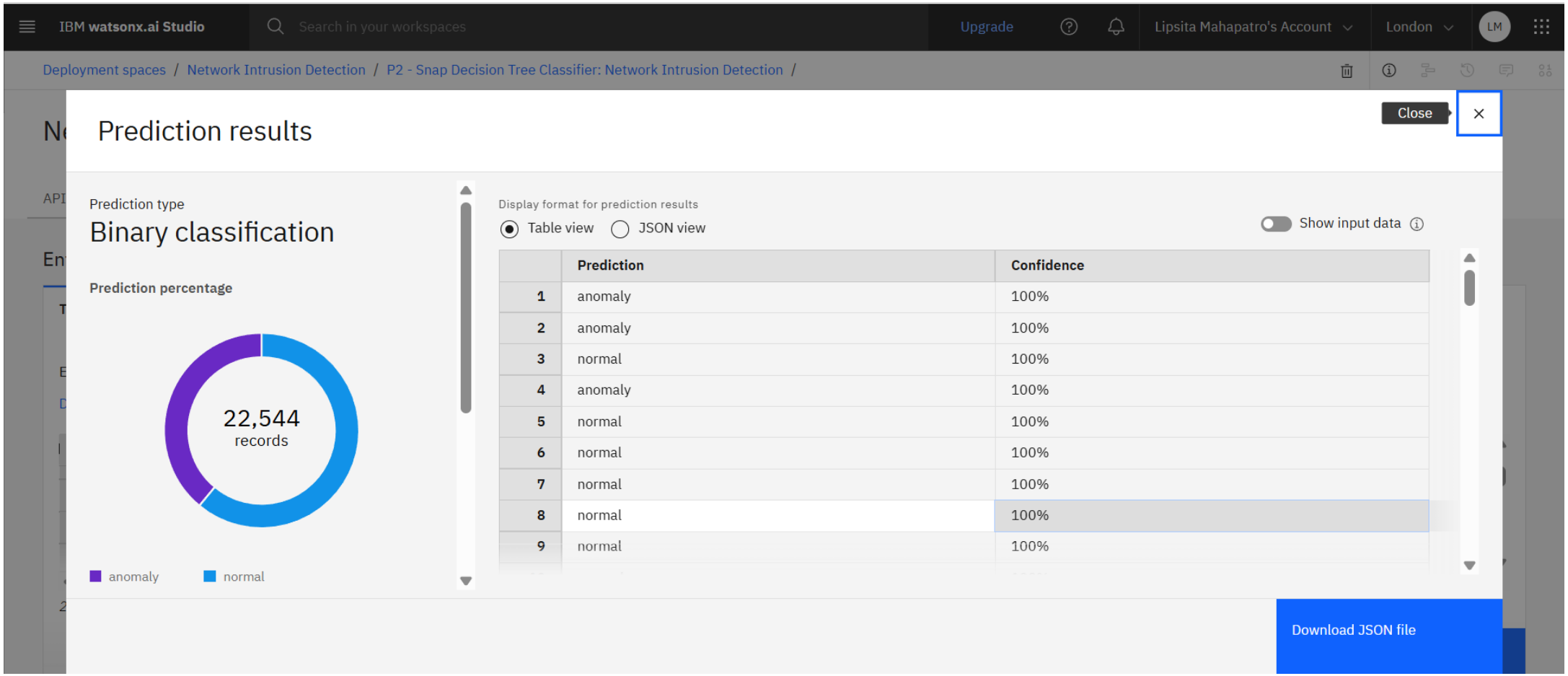


edunet foundation

RESULT



RESULT



CONCLUSION

This project successfully demonstrated the development of a robust Random Forest model capable of effectively classifying network traffic and identifying various cyber-attacks with high accuracy. The model's strong performance, validated by high Precision, Recall, and F1-Scores, confirms its effectiveness as a reliable solution for distinguishing malicious activities from normal network behavior.

The primary challenge in this domain is the constant evolution of new and unseen threats (zero-day attacks), which requires continuous adaptation. Potential improvements include implementing an automated retraining pipeline to keep the model updated with new attack patterns and exploring deep learning models to capture more complex sequential data in network flows.

Ultimately, this work underscores the critical importance of machine learning in modern cybersecurity. An intelligent NIDS provides an essential layer of proactive defense, enabling security teams to receive early warnings and respond swiftly to threats, thereby safeguarding critical communication networks and sensitive digital assets.

FUTURE SCOPE

Advanced Modeling with Graph Neural Networks (GNNs): Instead of treating each bike station independently, the entire network can be modeled as a graph. A GNN could learn the spatial relationships and bike flow patterns between stations, likely leading to more accurate, network-aware predictions.

Reinforcement Learning for Optimal Rebalancing: Move beyond just prediction to prescription. A Reinforcement Learning (RL) agent could be trained to suggest the most efficient real-time rebalancing strategies—dispatching vehicles to move bikes from full stations to empty ones to minimize operational costs and maximize availability.

Integration of Real-time Event Data: Enhance the model by integrating real-time data from social media or local news APIs. Using Natural Language Processing (NLP), the system could automatically detect and react to spontaneous events like concerts, protests, or traffic accidents that significantly impact bike demand.

Dynamic Pricing and Incentivization: Use the demand forecasts to power a dynamic pricing system. Prices could be slightly lowered at stations with a surplus of bikes or slightly increased in high-demand areas to financially incentivize riders to help organically balance the system.

REFERENCES

- Primary Dataset: The model was trained and evaluated using the specific Network Intrusion Detection dataset from Kaggle.
- Data Understanding: Preprocessing and feature engineering strategies were guided by foundational research papers analyzing the NSL-KDD and CIC-IDS2017 datasets.
- Algorithm Selection: The Random Forest classifier was chosen based on seminal academic work demonstrating its high accuracy and robustness for intrusion detection tasks.
- Core Methodology: Best practices for handling class imbalance, such as the SMOTE technique, were incorporated to ensure the model's reliability in detecting rare attacks.

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

Lipsita Mahapatro

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 18, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/294f82c3-ef16-42e7-9ec8-8810b740225e

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Lipsita Mahapatro

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

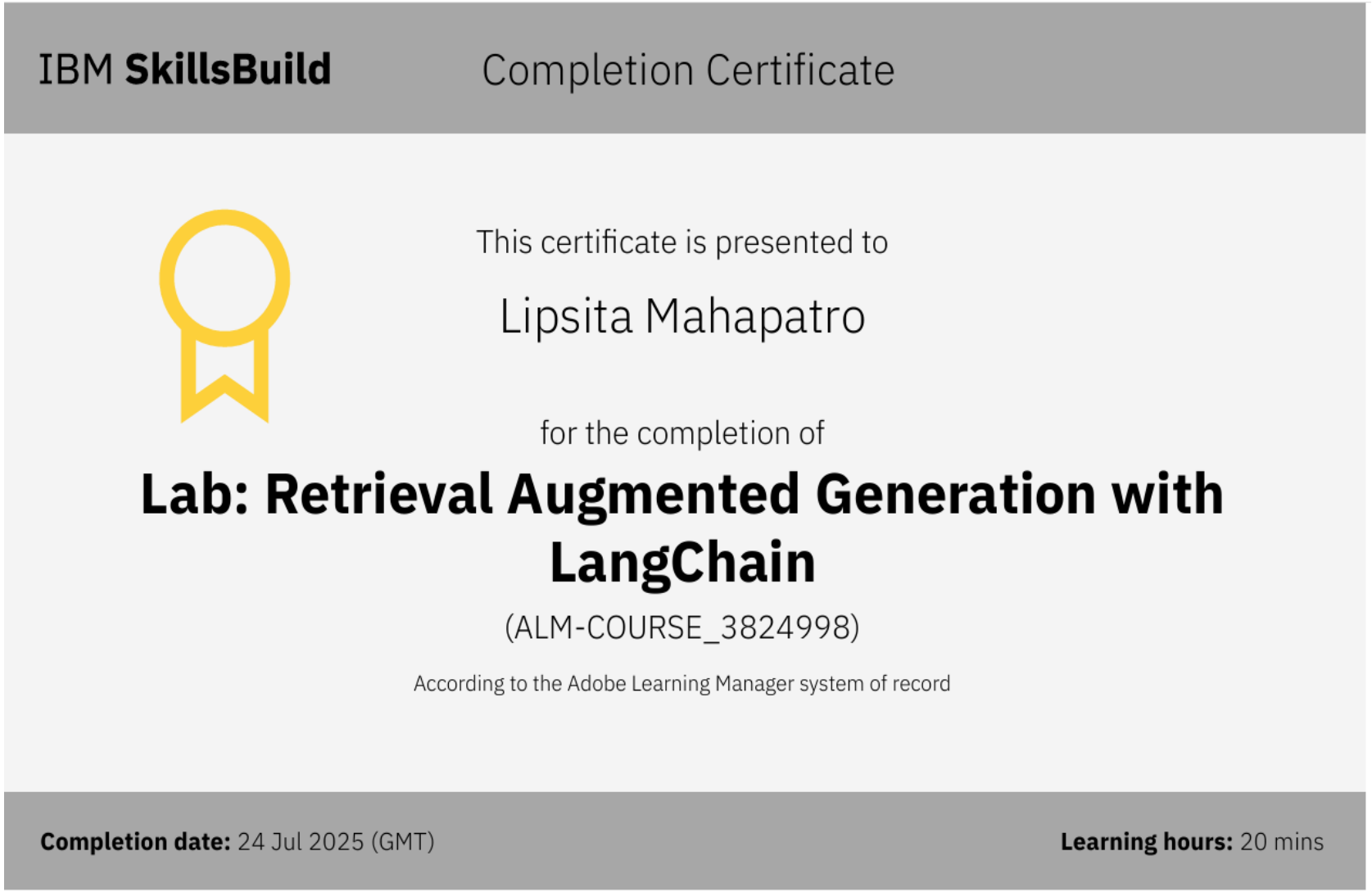
Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/8cabaa3e-f9fc-43d9-bee2-6f1057912824

IBM

edunet foundation

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Lipsita Mahapatro

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 24 Jul 2025 (GMT) **Learning hours:** 20 mins

edunet foundation

THANK YOU

edunet foundation