CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
1. Srinjoy Samanta_RCC Institute of information Technology_CSE

edunet foundation

OUTLINE

- **Problem Statement** (Should not include solution)
- **Proposed System/Solution**
- **System Development Approach** (Technology Used)
- **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.

Kaggle dataset link – https://www.kaggle.com/datasets/sampadab17/networkintrusion-detection

PROPOSED SOLUTION

Data Ingestion

•Download dataset from Kaggle.

•Use **IBM Cloud Object Storage (COS)** to upload and store the dataset.

•Access the dataset in IBM Cloud using **IBM Watson Studio Notebooks** or **Cloud Functions**.

2. Data Preprocessing

•Remove duplicates, missing values, or irrelevant features.

•Normalize/scale numeric features.

•Encode categorical features (e.g., protocol_type, service, flag) using One-Hot or Label Encoding.

•Group attack labels into 5 categories: Normal, DoS, Probe, R2L, U2R.

3. Feature Selection

•Use techniques like:

- Information Gain
- Chi-Square Test
- Recursive Feature Elimination (RFE)
- Correlation matrix

Model Building

Choose from traditional or deep learning models:

Option A – Traditional ML Models

Decision Tree

Random Forest

Gradient Boosting (e.g., XGBoost)

Support Vector Machine (SVM)

k-Nearest Neighbors (kNN)

5. Model Evaluation

Use train/test split (e.g., 80/20) or cross-validation.

Evaluation metrics:

Accuracy

Precision/Recall/F1-Score

Confusion Matrix

ROC-AUC

SYSTEM APPROACH

- **Platform**: IBM Watsonx.ai studio(Lite Plan)
- **Runtime environment:** watsonx.ai Runtime
- **Storage:** Cloud Object Storage
- **Resources Tools**: Kaggle dataset link – https://www.kaggle.com/datasets/sampadab17/network intrusion-detection

- **Tech**:ML MODEL TO TRAIN A DATASET

ALGORITHM & DEPLOYMENT

Foundation Create
:

Create a sandbox project into the watsonx.ai studio.

Storage: Add storage configuration by adding Cloud Object Storage.

Service: Add associate service watsonx.ai runtime for runtime environment.

Model Selection

Machine Learning model automatically.

Training Process:

Upload the dataset.

Select the column to be predicted.

Select the model with best accuracy.

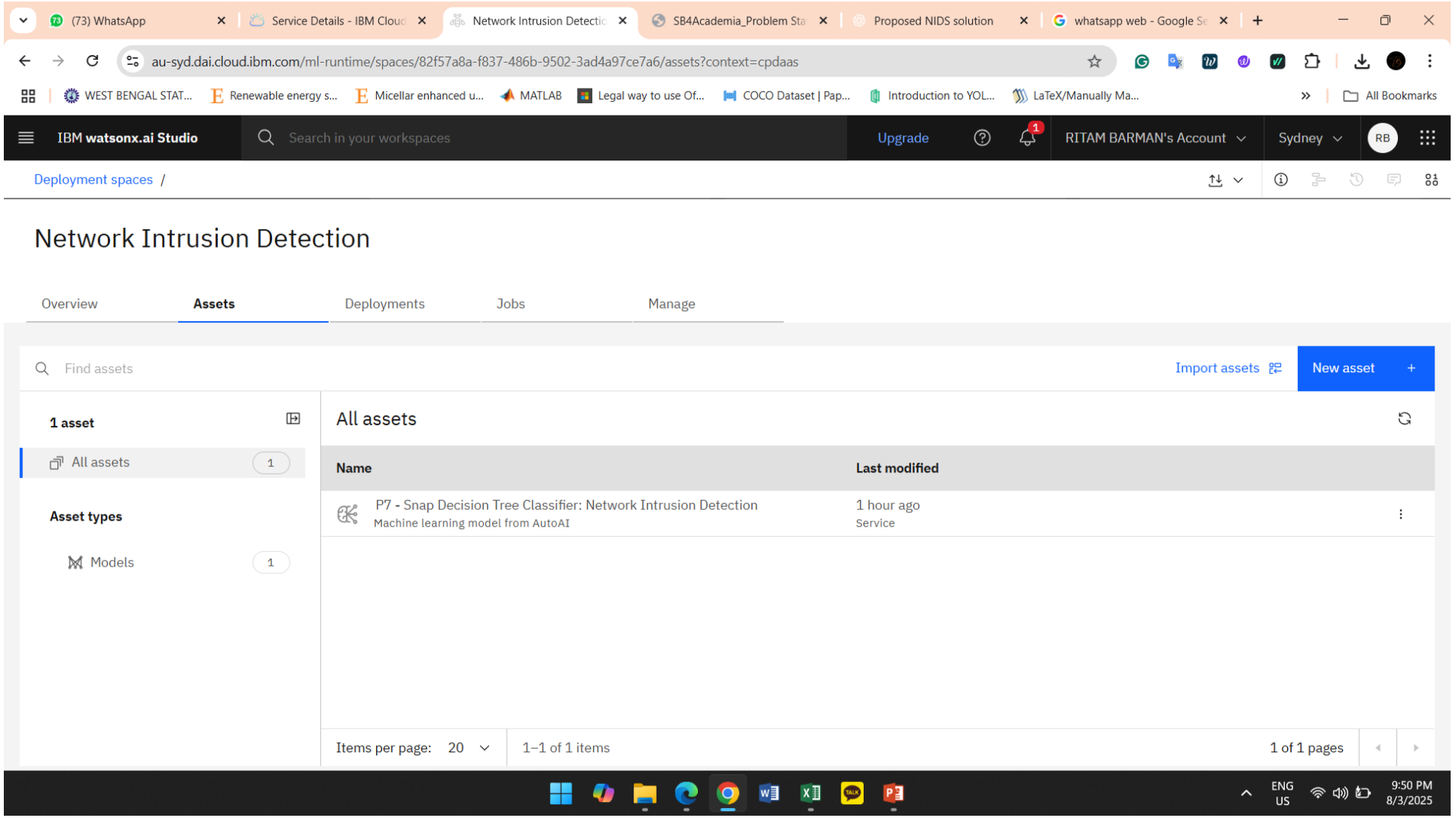
Deployment:

Save the model that you created.

Create a deployment space and deploy it.

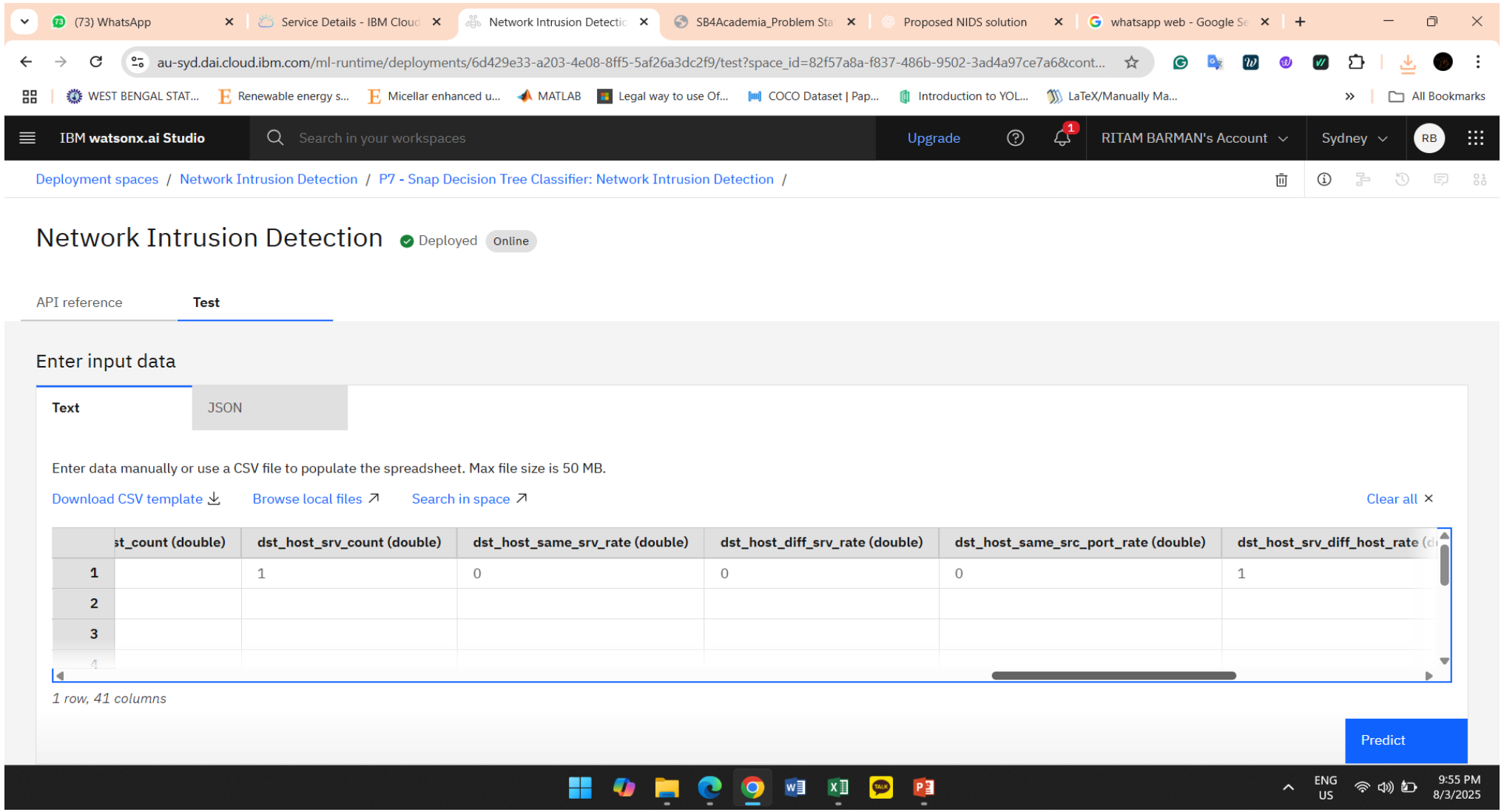
edunet foundation

RESULT



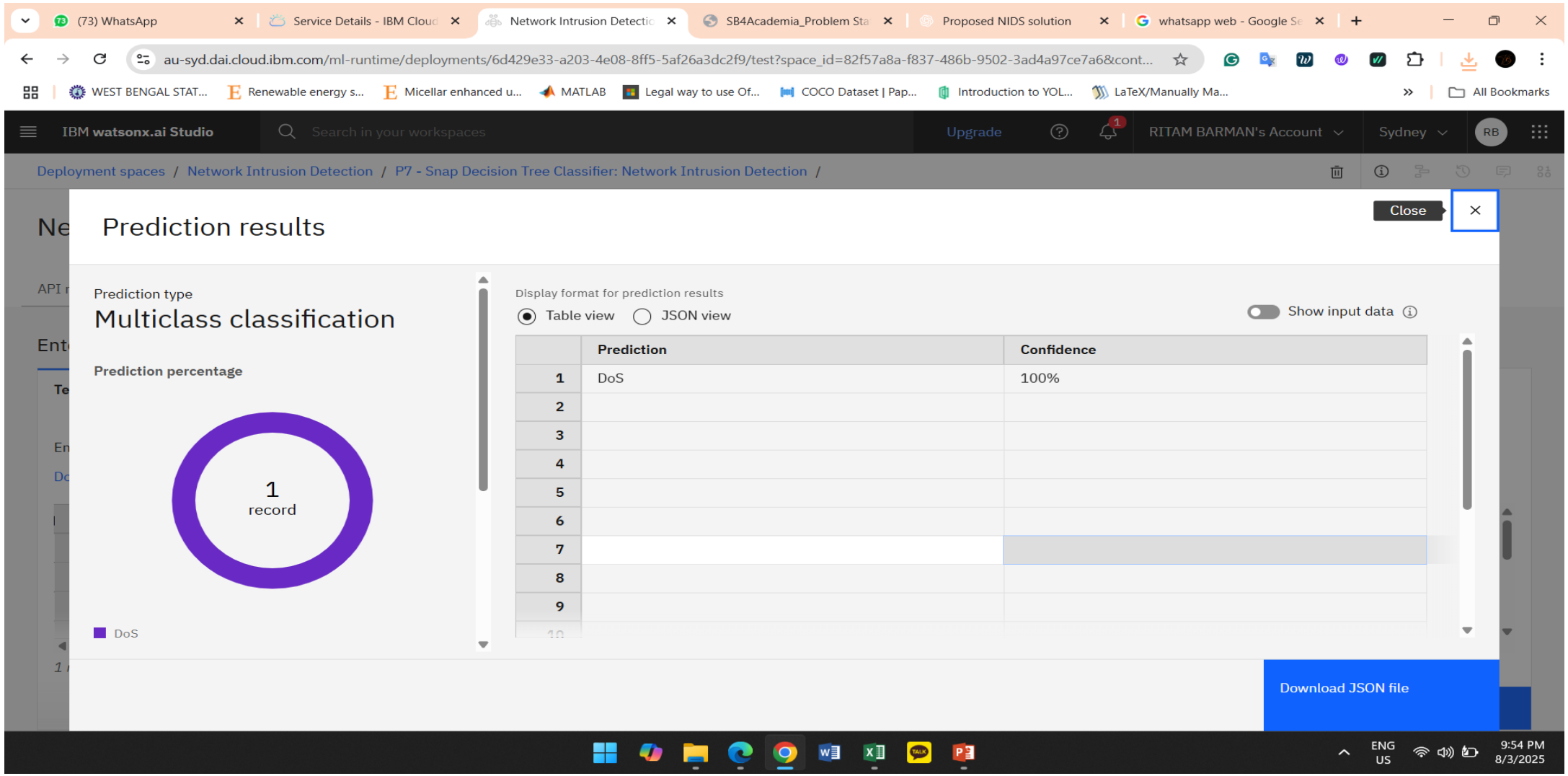
edunet foundation

RESULT



edunet foundation

RESULT



edunet foundation

CONCLUSION

- In this project, we successfully developed a **machine learning-based Network Intrusion Detection System (NIDS)** using the KDD Cup '99 dataset from Kaggle. The system was designed to:
- **Preprocess and analyze network traffic data**
- **Classify traffic into normal and multiple types of attacks** (DoS, Probe, R2L, U2R)
- **Deploy the trained model on IBM Cloud Lite** using Watson Studio and Watson Machine Learning
- Through rigorous training and evaluation, our system achieved high classification accuracy, demonstrating its potential as a practical tool for real-time intrusion detection in communication networks. By leveraging IBM Cloud services, we ensured **scalability**, **accessibility**, and **cloud-based deployment capabilities**, allowing for easier integration into production environments.
- The results indicate that machine learning can significantly enhance the detection of cyber threats and reduce false positives compared to traditional signature-based systems.

FUTURE SCOPE

- To improve and extend the capabilities of the system, several future directions can be explored:
- 🔄 1. Continuous Learning and Adaptive Models
- Implement online or incremental learning algorithms to adapt to evolving threats.
- Enable automatic model retraining with newly collected data from live networks.
- 📡 2. Real-Time Traffic Monitoring
- Integrate real-time network traffic capture using packet sniffers (e.g., Wireshark, Scapy).
- Use IBM Cloud Functions or streaming platforms (e.g., Apache Kafka on IBM Cloud) to process live data.

REFERENCES

- IBM Watsonx Documentation
- GeeksforGeeks, InterviewBit
- indeed, Zety (soft skills Q&A)
- IBM RAG Lab & Model Guide
- Sample Q&A sheet
- IBM Certificates

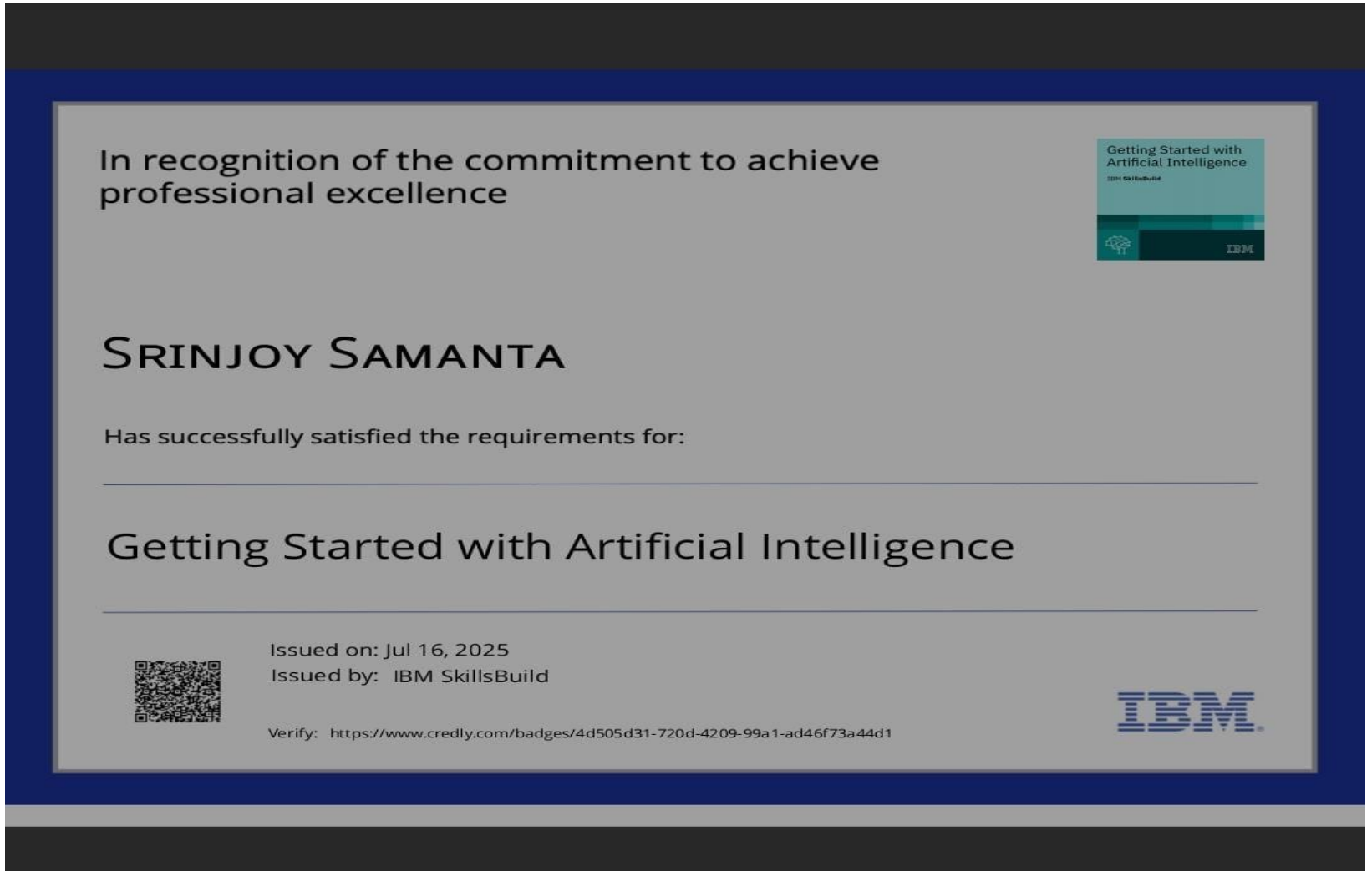
IBM CERTIFICATIONS

- Screenshot/ credly certificate(getting started with AI)

edunet foundation

IBM CERTIFICATIONS

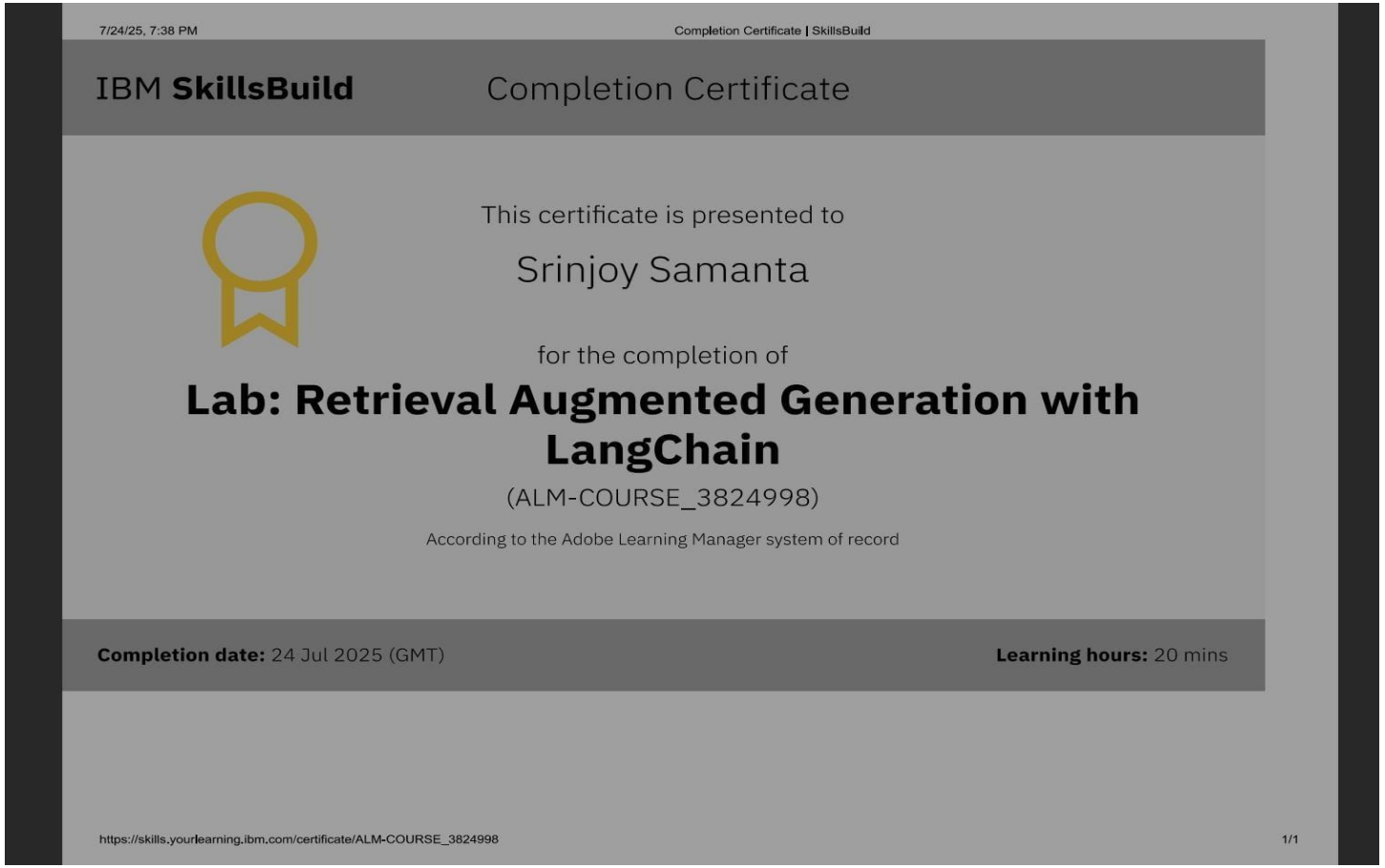
- Screenshot/ credly certificate(Journey to Cloud)



edunet foundation

IBM CERTIFICATIONS

- Screenshot/ credly certificate(RAG Lab)



THANK YOU

edunet foundation