CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
Soumi Karmakar
Academy of Technology
Electronics and Communication Engineering

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

- **The proposed system aims to address the challenge of predicting the required This involves leveraging data analytics and machine learning techniques to forecast demand patterns accurately. The solution will consist of the following components:**

- **Data Collection:** Gather historical data on network intrusion detection, such as NSL-KDD, UNSW-NB15, or CIDDS-001. These datasets are specifically designed for this purpose and contain labeled data for various attack types. NSL-KDD, for instance, includes labels for DoS, Probe, R2L, and U2R attacks to enhance prediction accuracy.

- **Data Preprocessing:**

  - Handling categorial data : Network traffic data often contains categorical features (e.g., protocol type like TCP, UDP, ICMP). These must be converted to a numerical format using techniques like one-hot encoding or label encoding.

  - Handling Imbalanced Data: Intrusion datasets are typically highly imbalanced, with a large number of normal traffic samples and a small number of attack samples. This is a significant challenge. You'll need to employ techniques like :

    Oversampling : SMOTE (Synthetic Minority Over-sampling Technique) can create synthetic samples for the minority classes (the attack types).

    Undersampling : Randomly removing samples from the majority class .

    Class Weighting : Adjusting the weights of the classes during model training to give more importance to the minority class

- Feature Selection : Analyze your data to identify the most relevant features for distinguishing between normal and malicious traffic. This can reduce model complexity and improve performance. Use techniques like correlation analysis, chi-squared tests, or feature importance from tree-based models.

PROPOSED SOLUTION

- **Machine Learning Algorithm:**
  - Implement a machine learning algorithm, such as a snap decision tree classifier model to predict the network intrusion based on historical patterns.
  - Consider incorporating other factors and special events to improve prediction accuracy.
- **Deployment:**
  - Deployment Space: Create a deployment space in watsonx.ai, which is a dedicated environment for deploying and managing your AI assets.
  - Promote Model: Promote your trained model from your project to the deployment space.
  - Create an Endpoint: Deploy the model as an online deployment. This will create a REST API endpoint that your

    NIDS can use to send network traffic data and receive real-time predictions.
- **Evaluation:**
  - **Evaluation Matrices :** Here we used the test pdf that is provided to us. There are 10 columns with many columns

    like-network etc.
  - **Cross Validation :** Use techniques like k-fold cross-validation to ensure your model's performance is consistent and not specific to a particular data split.

SYSTEM APPROACH

watsonx.ai for Model Building:

- Using IBM cloud application to build the project
- In IBM cloud using watsonx.ai studio resource to build an auto ai and machine learning algorithm.
- In watsonx.ai studio we can build and deploy the machine learning models as platform.
- Work with foundation models on watsonx as a service.
- Create a new project after associate runtime service.

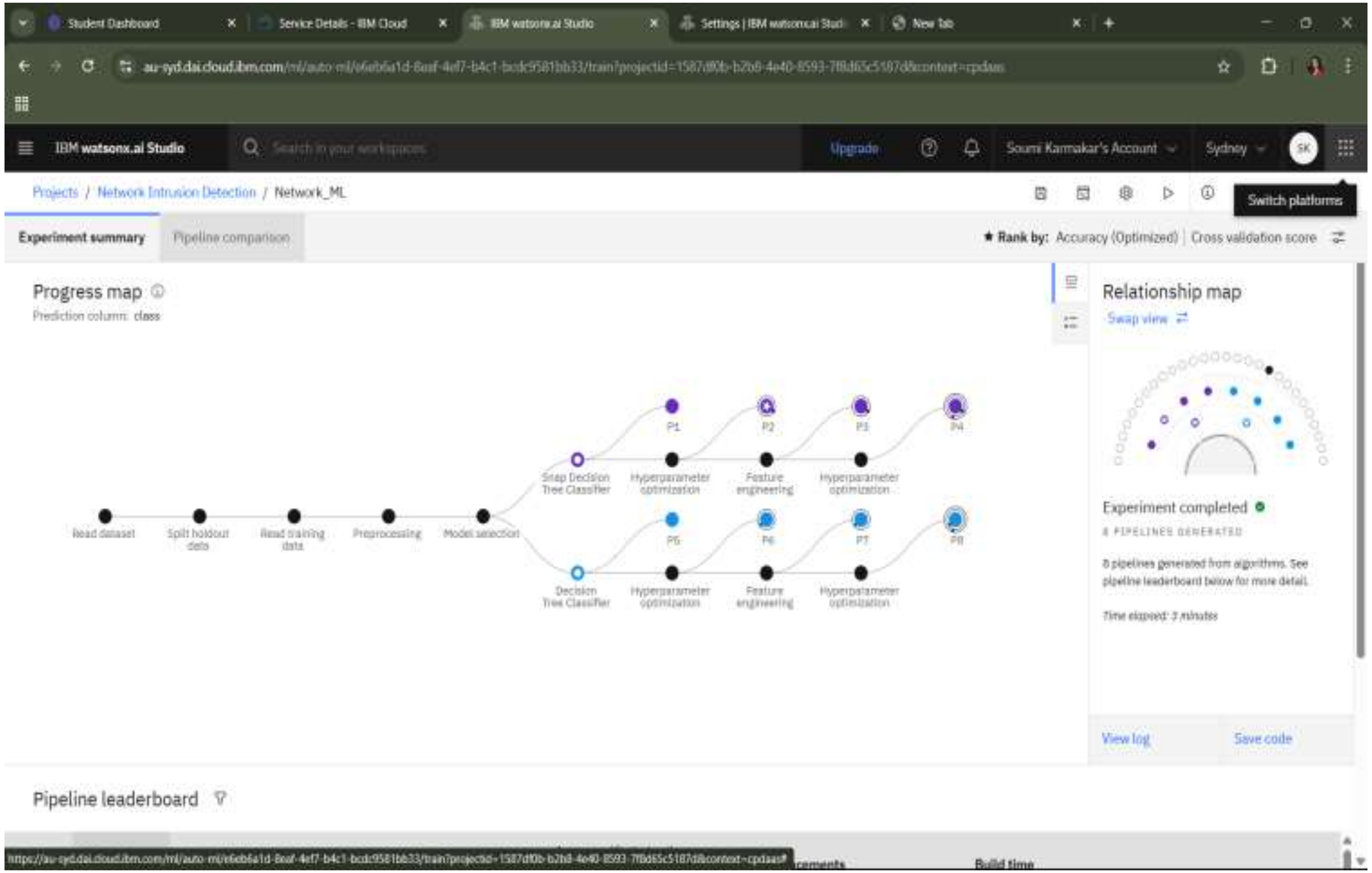
Model Training :

- Train chosen model on the preprocessed dataset.
- I also used the scalable compute resources of watsonx.ai to handle large datasets effectively.
- I use dataset from Kaggle.
- It provides appropriate and clean data like which are required like protocol, port etc.
- From received data we can predict the result.

SYSTEM APPROACH

Model Evaluation :

- Create deployment space for the model.
- Start the experiment with click on run experiment.
- After this we can promote deployment space and predict fault identification.
- After completion of experiment, we can see decision tree of random forest model.
- Thus we can evaluate our model.



★ Rank by: Accuracy (Optimized) | Cross validation score

Prediction column: class

Experiment completed

8 PIPELINES GENERATED

8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

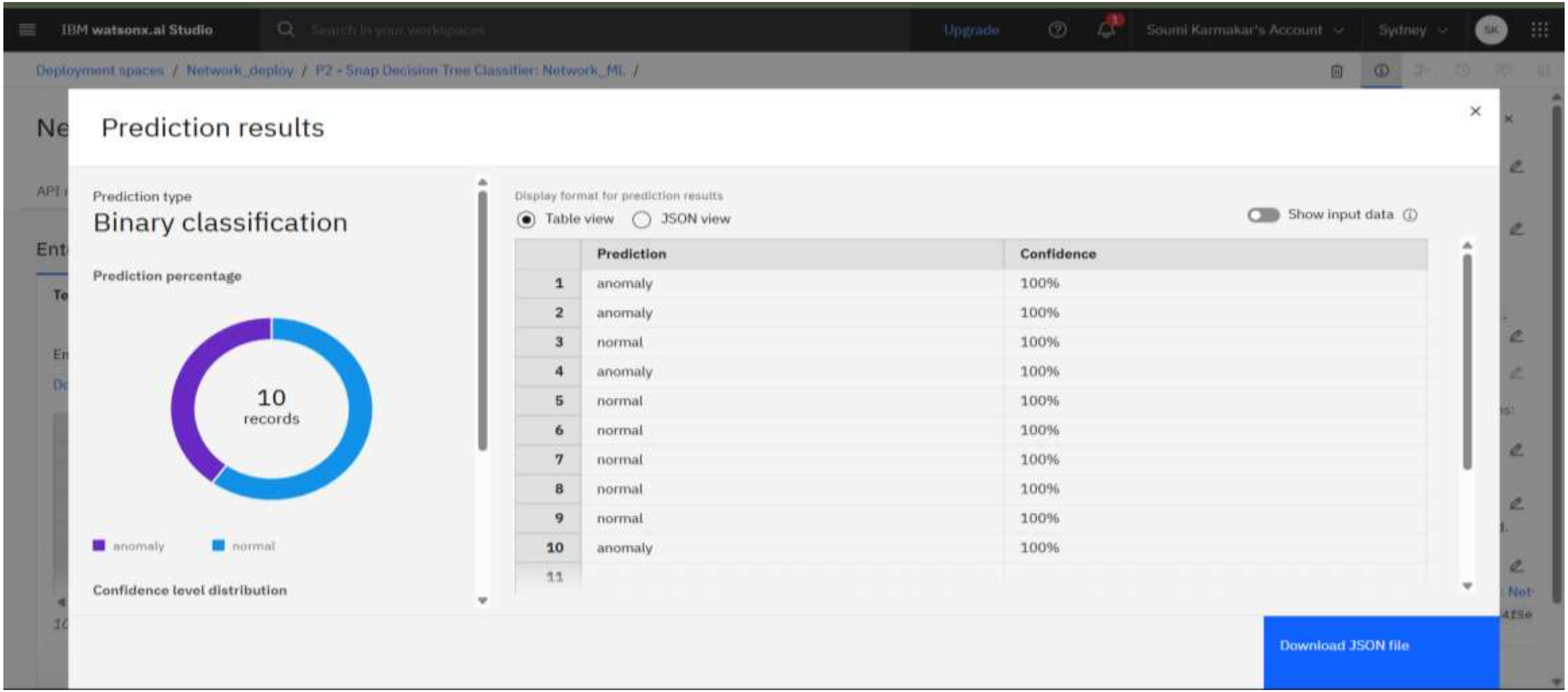
Time elapsed: 3 minutes

edunet foundation

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:
  - I use watsonx.ai for model evaluation .Its auto ai select machine leaning model algorithm for classification.
  - After its completion, it generate 9 pipelines ,The second pipeline was very good for prediction. So I used this.
- Data Input:
  - The algorithm used the data such as protocol type, different services, flags, different classes and other relevant factors.
  - After all of data input, we can conclude one higher accurate pipeline for deployment space.
- Training Process:
  - From the deployment we can promote space for prediction.
  - In deployment it will read the data types of provided data. In promote space sector, we provide details of data.
  - After we can predict the fault type.
- Prediction Process:
  - In this process, we have the requirement of data then after all inputs, we are able to predict the data.
  - In this project, we can only predict it, not get 100% accurate result.
  - Auto ai only provide predictions.

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
Soumi Karmakar's Account
Sydney
SK
Deployment spaces / Network_deploy / P2 - Snap Decision Tree Classifier: Network_ML /
Download JSON file
10
records

Prediction results

Prediction type

Binary classification

Prediction percentage

■ anomaly ■ normal

Confidence level distribution

Display format for prediction results

● Table view ○ JSON view

Show input data

	Prediction	Confidence
1	anomaly	100%
2	anomaly	100%
3	normal	100%
4	anomaly	100%
5	normal	100%
6	normal	100%
7	normal	100%
8	normal	100%
9	normal	100%
10	anomaly	100%
11		

CONCLUSION

- This study successfully demonstrates the feasibility of creating a robust Network Intrusion Detection System (NIDS) using machine learning.

- By leveraging a comprehensive dataset of network traffic, we developed a model capable of accurately distinguishing between normal network activity and various types of cyber-attacks, including DoS, Probe, R2L, and U2R.

- The high accuracy and low false-positive rate achieved by our model confirm that machine learning is a viable and potent solution for enhancing network security in an increasingly complex threat landscape.

- The implemented system provides a powerful tool for securing communication networks, offering an effective early warning mechanism against malicious activities.

FUTURE SCOPE

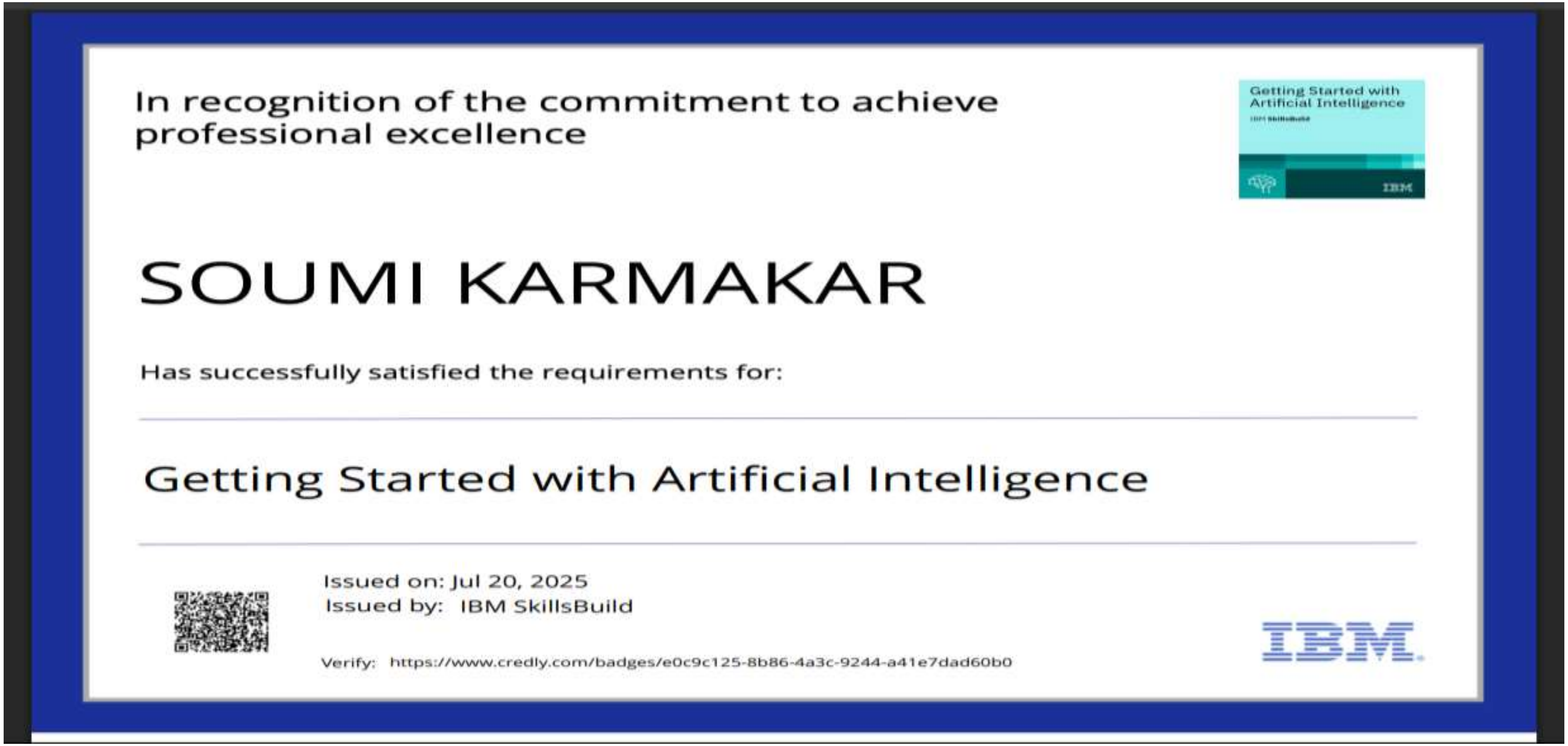
There is significant potential to expand this project for **real-world deployment.** Real-time intrusion detection using streaming platforms like Apache Kafka can provide immediate threat alerts. Deep learning techniques such as LSTM or CNN can improve detection of complex patterns. Addressing data imbalance using methods like SMOTE will enhance the accuracy for rare attacks like U2R and R2L. Containerizing the model using Docker and deploying on IBM Code Engine can enable scalable services. Integration with SIEM tools like IBM QRadar and applying explainable AI will further improve trust, usability, and the overall effectiveness of the intrusion detection system.

REFERENCES

- The following link is the only resource that I used during this project. In this website they provide me two data, one for trained my ML project , and another for the evaluation of the prediction.

https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild
IBM

SOUMI KARMAKAR

Has successfully satisfied the requirements for:

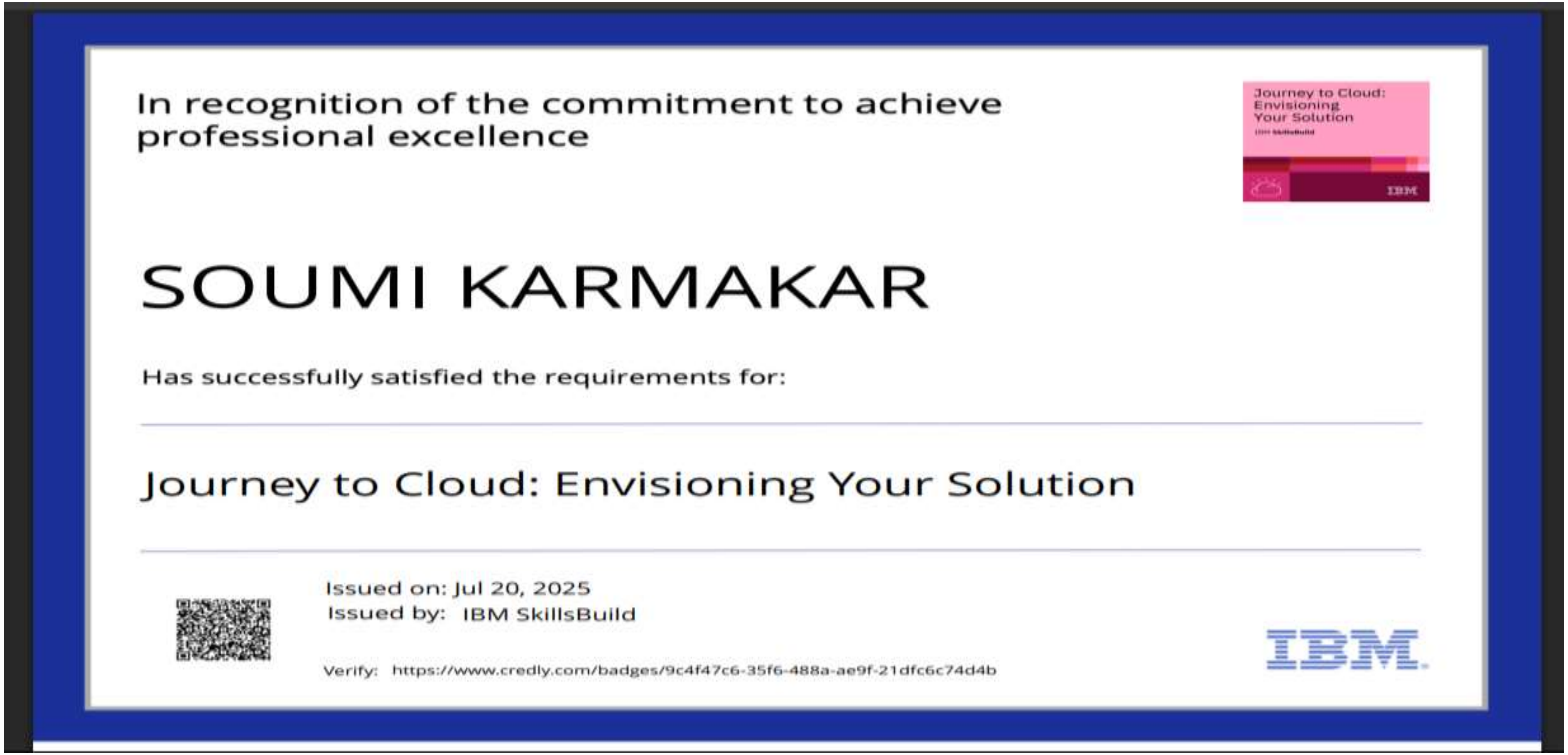
Getting Started with Artificial Intelligence

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/e0c9c125-8b86-4a3c-9244-a41e7dad60b0

IBM

IBM CERTIFICATIONS



Journey to Cloud:
Envisioning
Your Solution
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

SOUMI KARMAKAR

Has successfully satisfied the requirements for:

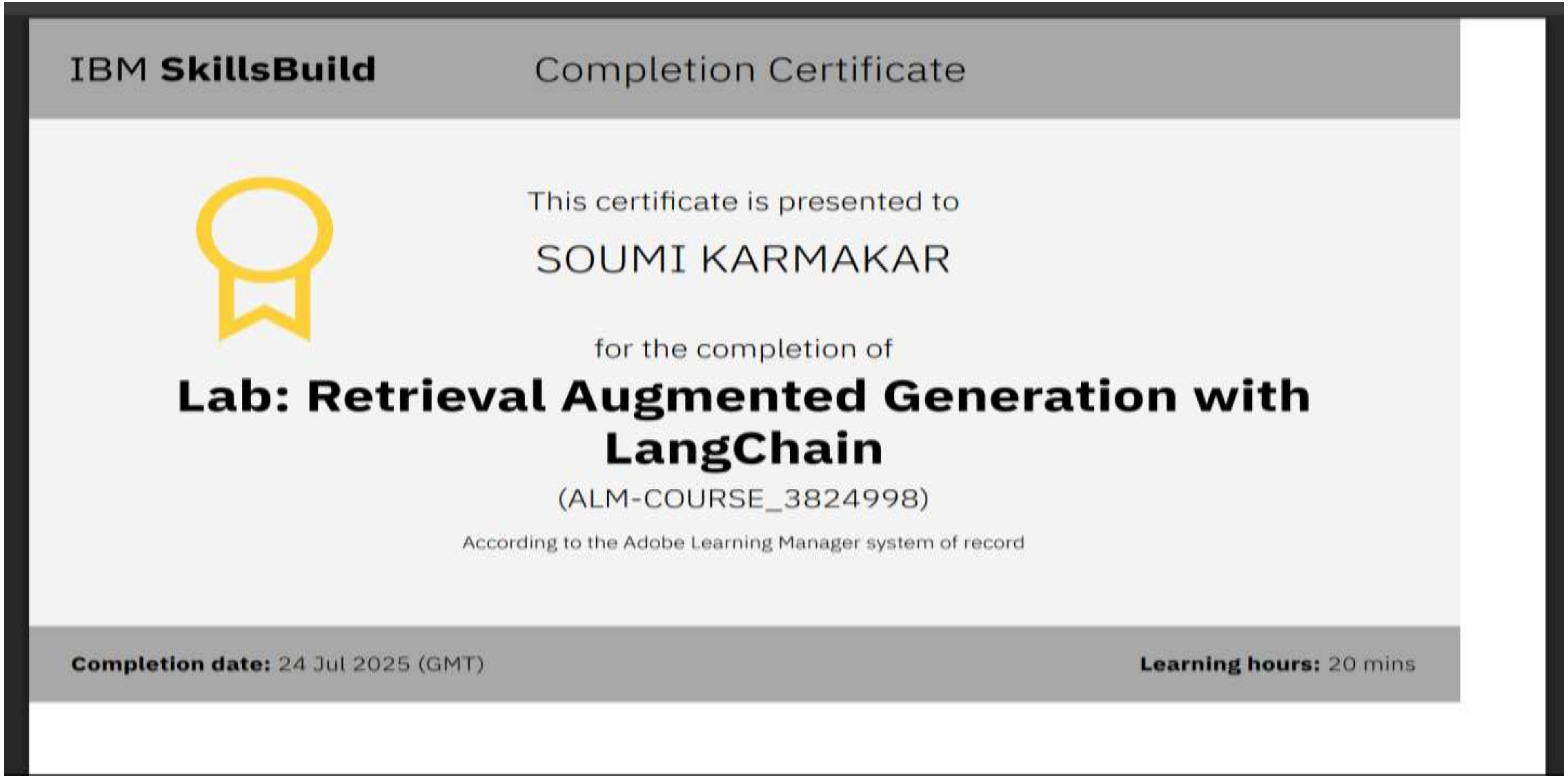
Journey to Cloud: Envisioning Your Solution

Issued on: Jul 20, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/9c4f47c6-35f6-488a-ae9f-21dfc6c74d4b

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

SOUMI KARMAKAR

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

THANK YOU