CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM USING MACHINE LEARNING

Presented By:
1. ANCHURI HARINI-MALLA REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY-COMPUTER SCIENCE AND ENGINEERING.

OUTLINE

PROBLEM STATEMENT

With the increasing volume of data exchanged over communication networks, cyber-attacks have become more frequent and sophisticated. Traditional security systems often fail to detect complex, evolving threats in real time. One of the major challenges is accurately identifying and classifying various forms of network intrusions—such as Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) attacks—while distinguishing them from normal traffic. A robust, intelligent system is required to analyze network traffic and detect intrusions early, ensuring the security and integrity of communication networks.

PROPOSED SOLUTION

The proposed system aims to develop a Machine Learning-based Network Intrusion Detection System (NIDS) that can analyze network traffic and classify it as either normal or malicious. The system will be trained on the Kaggle "Network Intrusion Detection" dataset, which includes various types of attacks such as DoS, Probe, R2L, and U2R.

- The solution includes the following key components:
- **Data Collection:** Use the Kaggle dataset containing labeled network traffic records.
- **Data Preprocessing:** Handle missing values, convert categorical features, scale numerical data, and perform feature selection to improve model performance.
- **Model Training:** Train and evaluate various classification algorithms such as Random Forest, Decision Tree, SVM, and possibly deep learning models like ANN or LSTM.
- **Deployment:** Implement the system using IBM Cloud Lite services, such as IBM Watson Studio for model building and IBM Cloud Object Storage for managing the dataset.
- **Prediction & Alerting:** Once deployed, the system will monitor traffic and alert when suspicious patterns are detected.

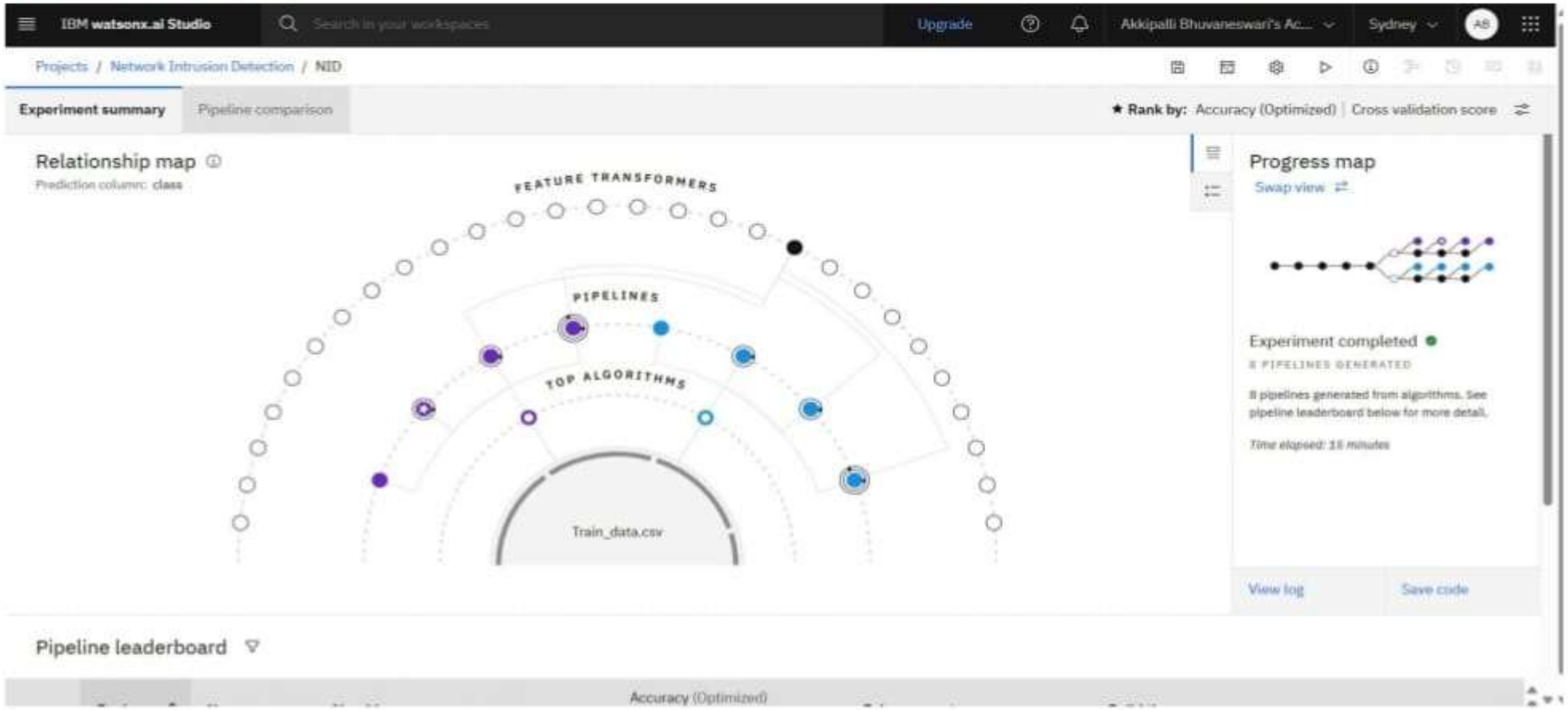
SYSTEM APPROACH

- The proposed Network Intrusion Detection System (NIDS) follows a data-driven machine learning approach to analyze network
- traffic and classify it into normal or attack categories.
- Approach Steps :
- 1. Data Collection
- Use benchmark datasets (NSL-KDD, CICIDS2017, UNSW-NB15) or captured live traffic.
- 2. Data Preprocessing
- Handle missing values, remove duplicates.
- Encode categorical features (protocol type, service, flag).
- Normalize numerical features for consistent scaling.
- 3. Feature Selection & Extraction
- Identify the most relevant features using correlation analysis or feature importance techniques.

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:**
  - Provide a brief overview of the chosen algorithm (e.g., time-series forecasting model, like ARIMA or LSTM) and justify its selection based on the problem statement and data characteristics.
- **Data Input:**
  - Specify the input features used by the algorithm, such as historical bike rental data, weather conditions, day of the week, and any other relevant factors.
- **Training Process:**
  - Explain how the algorithm is trained using historical data. Highlight any specific considerations or techniques employed, such as cross-validation or hyperparameter tuning.
- **Prediction Process:**
  - Detail how the trained algorithm makes predictions for future bike counts. Discuss any real-time data inputs considered during the prediction phase.

RESULT



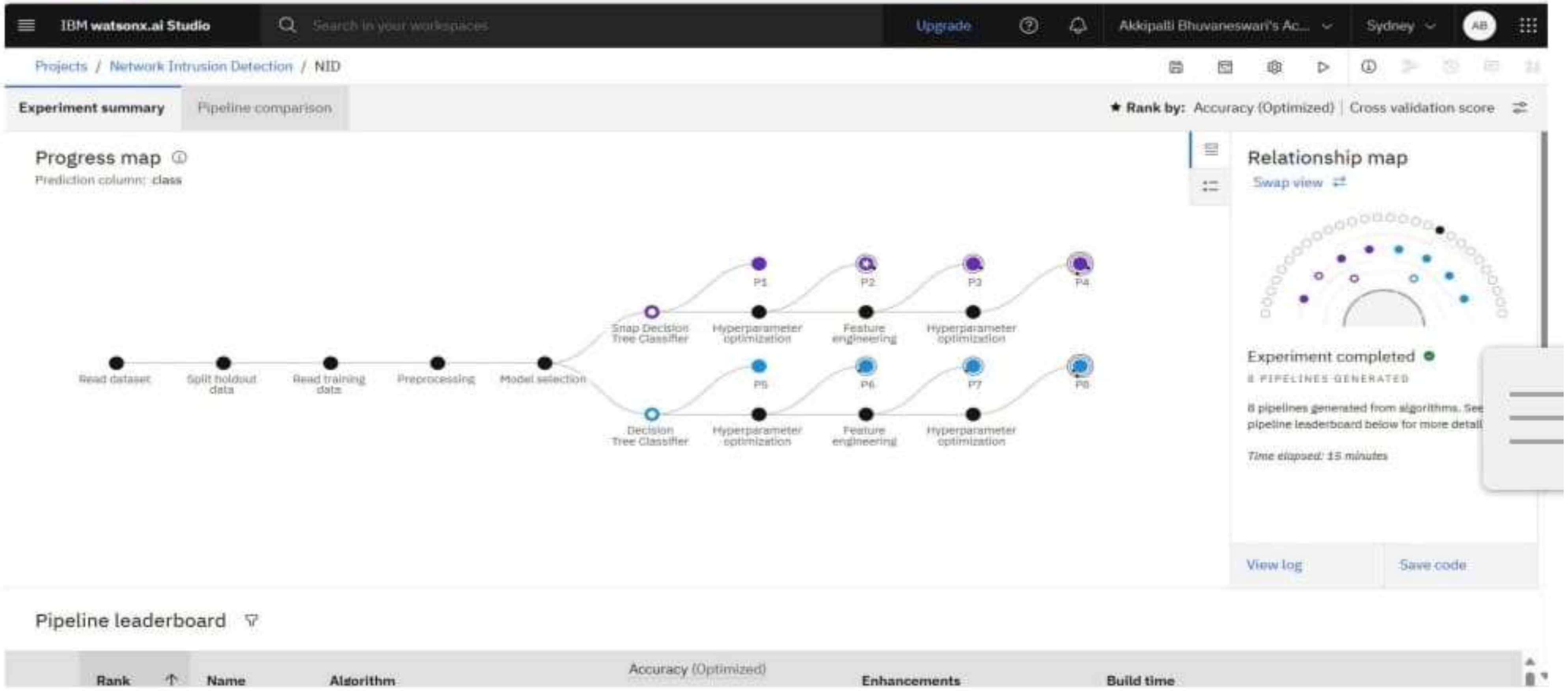
Experiment completed ●

8 PIPELINES GENERATED

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

Time elapsed: 15 minutes

RESULT


IBM watsonx.ai Studio

Projects / Network Intrusion Detection / NID

Experiment summary | Pipeline comparison

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

Progress map

Prediction column: class

Relationship map

Swap view

Experiment completed

8 PIPELINES GENERATED

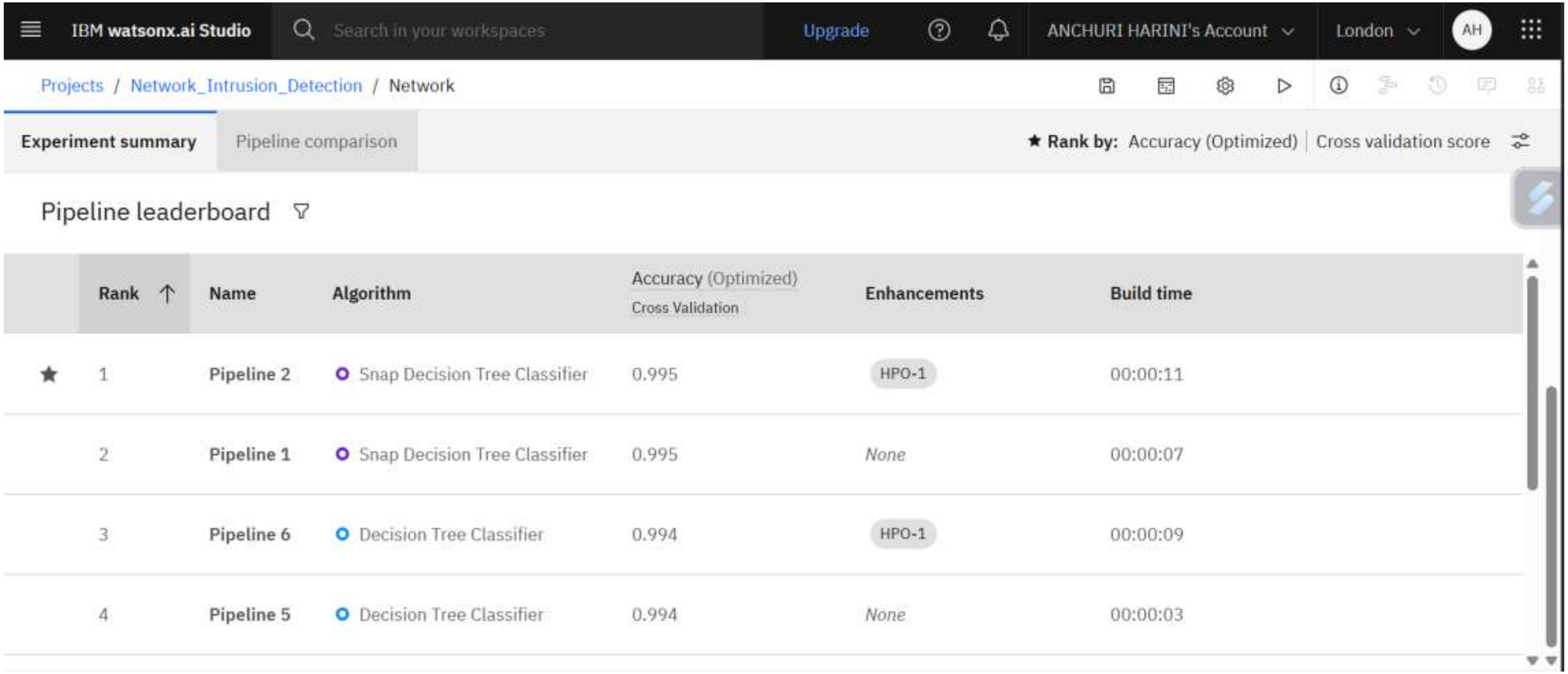
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail

Time elapsed: 15 minutes

Pipeline leaderboard

Rank	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time

RESULT


IBM watsonx.ai Studio

Projects / Network_Intrusion_Detection / Network

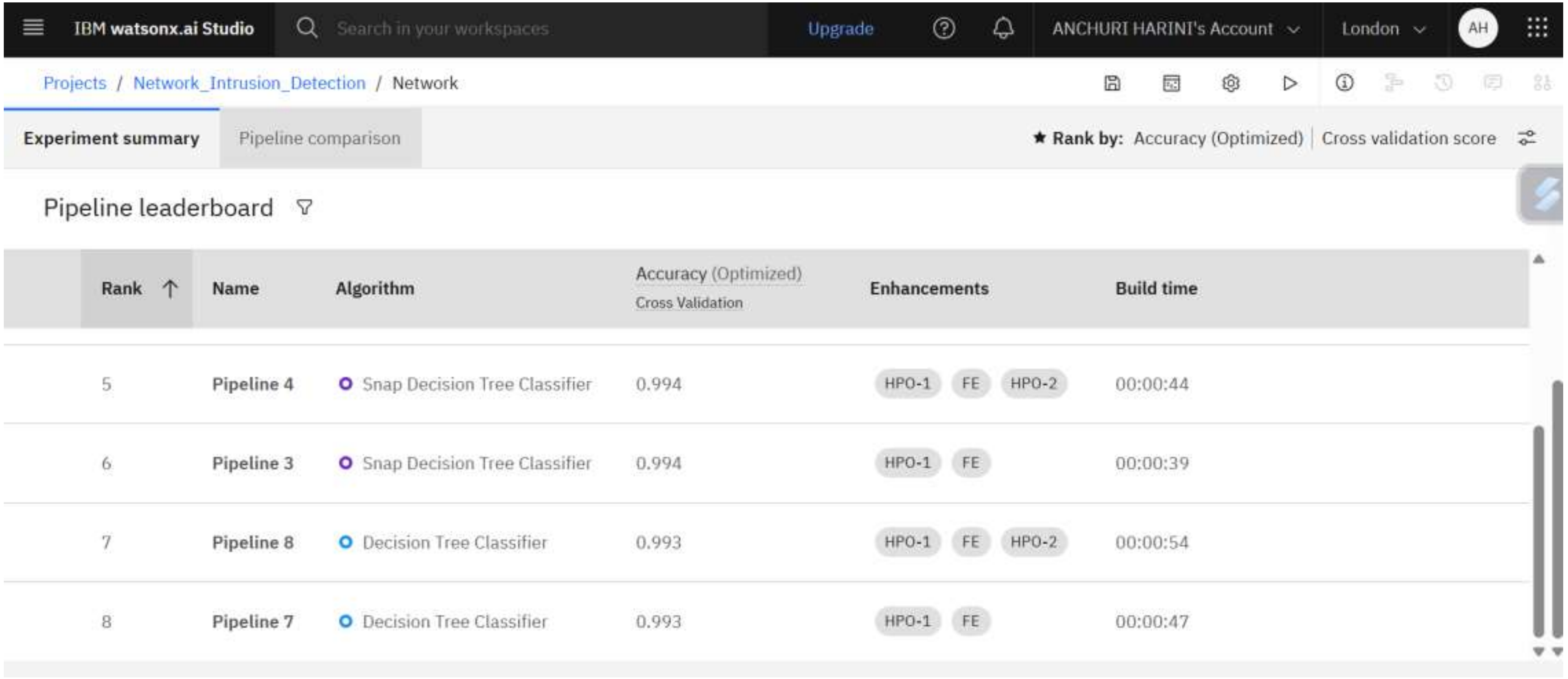
Experiment summary | Pipeline comparison

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

Pipeline leaderboard

	Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:11
	2	Pipeline 1	Snap Decision Tree Classifier	0.995	*None*	00:00:07
	3	Pipeline 6	Decision Tree Classifier	0.994	HPO-1	00:00:09
	4	Pipeline 5	Decision Tree Classifier	0.994	*None*	00:00:03

RESULT

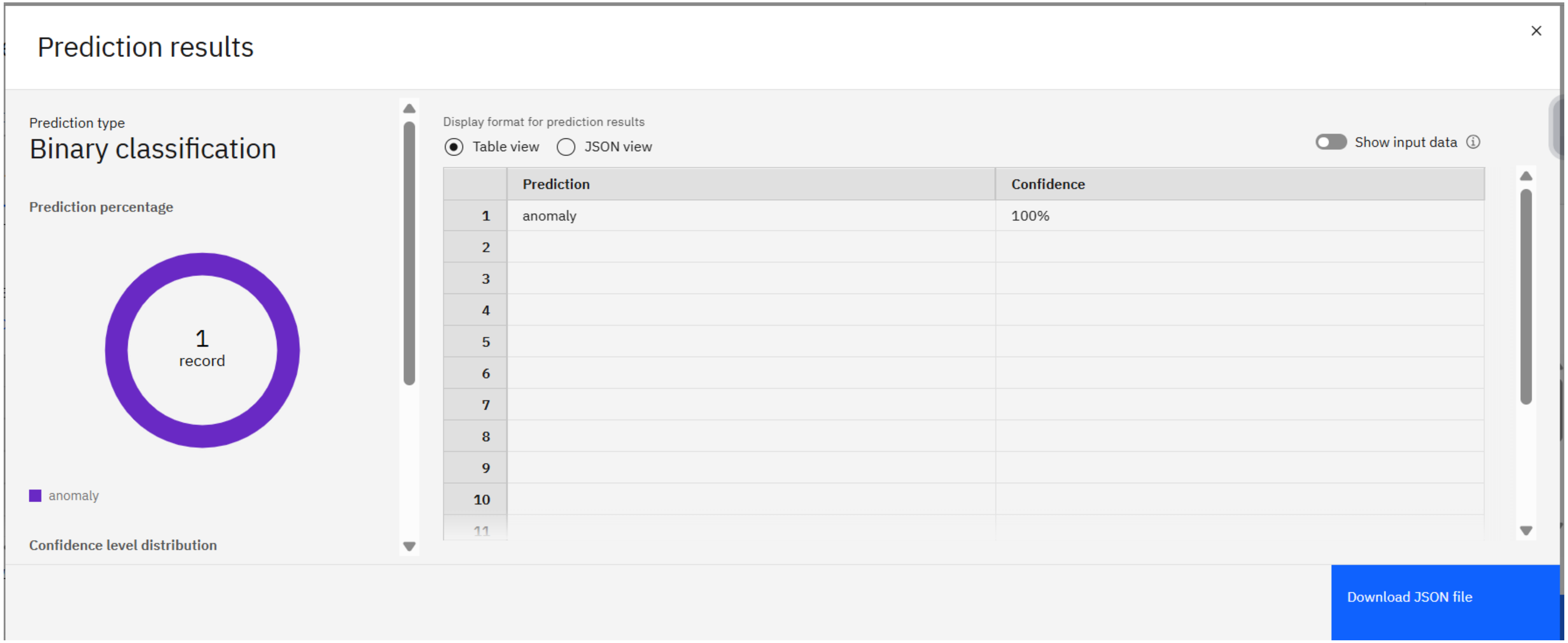


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
ANCHURI HARINI's Account
London
AH
Projects / Network_Intrusion_Detection / Network
Experiment summary
Pipeline comparison
★ Rank by: Accuracy (Optimized) | Cross validation score

Pipeline leaderboard

Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
5	Pipeline 4	Snap Decision Tree Classifier	0.994	HPO-1 FE HPO-2	00:00:44
6	Pipeline 3	Snap Decision Tree Classifier	0.994	HPO-1 FE	00:00:39
7	Pipeline 8	Decision Tree Classifier	0.993	HPO-1 FE HPO-2	00:00:54
8	Pipeline 7	Decision Tree Classifier	0.993	HPO-1 FE	00:00:47

RESULT



Prediction results

Prediction type

Binary classification

Prediction percentage

🟪 anomaly

Confidence level distribution

Display format for prediction results

[● Table view] ○ JSON view

Show input data

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

Download JSON file

CONCLUSION

- This project showed how machine learning can be used to spot different types of cyber-attacks by analyzing network traffic. The model worked well in identifying threats like DoS, Probe, R2L, and U2R, making it a useful tool for improving network security. We faced some challenges, especially with cleaning the data and choosing the right features, but those were handled through preprocessing and model tuning. In the future, this system can be made even better by adding real-time detection and learning from new types of attacks. Overall, it’s a strong step toward building smarter, more secure networks.

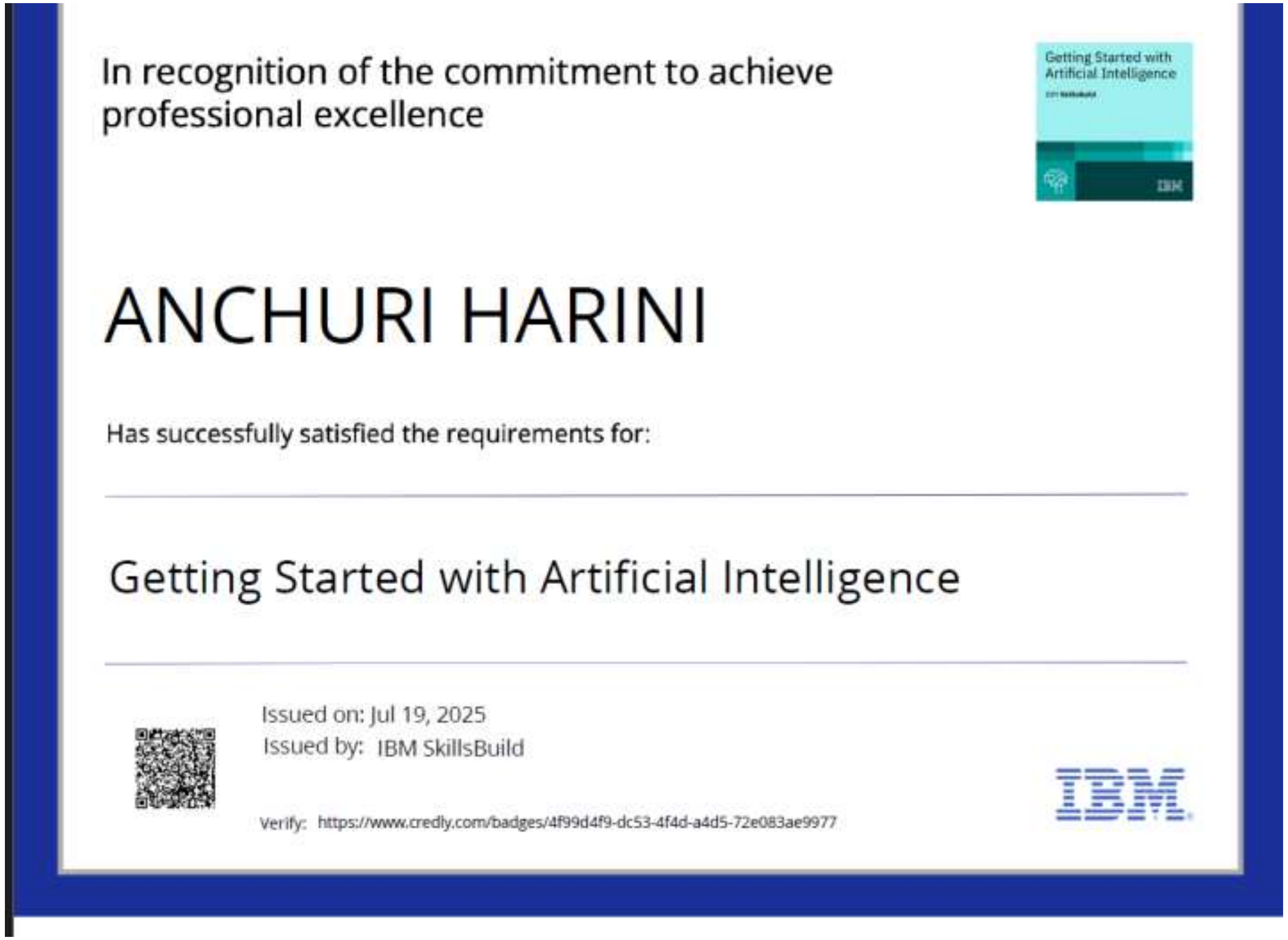
FUTURE SCOPE

- In the future, this system can be improved by adding real-time monitoring to catch threats as they happen. We can try advanced models like LSTM or CNN to boost accuracy and even let the system learn from new attacks over time. It could be deployed on edge devices for faster local detection and scaled easily using IBM Cloud. Adding more recent datasets would also help make it more reliable and effective across different types of networks.

REFERENCES

- IBM Developer. Build and Deploy Machine Learning Models on IBM Watson Studio.
- https://developer.ibm.com
- IBM Cloud. IBM Watson Machine Learning Service – Model Deployment and Scoring.
- https://cloud.ibm.com/catalog/services/watson-machine-learning

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

ANCHURI HARINI

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

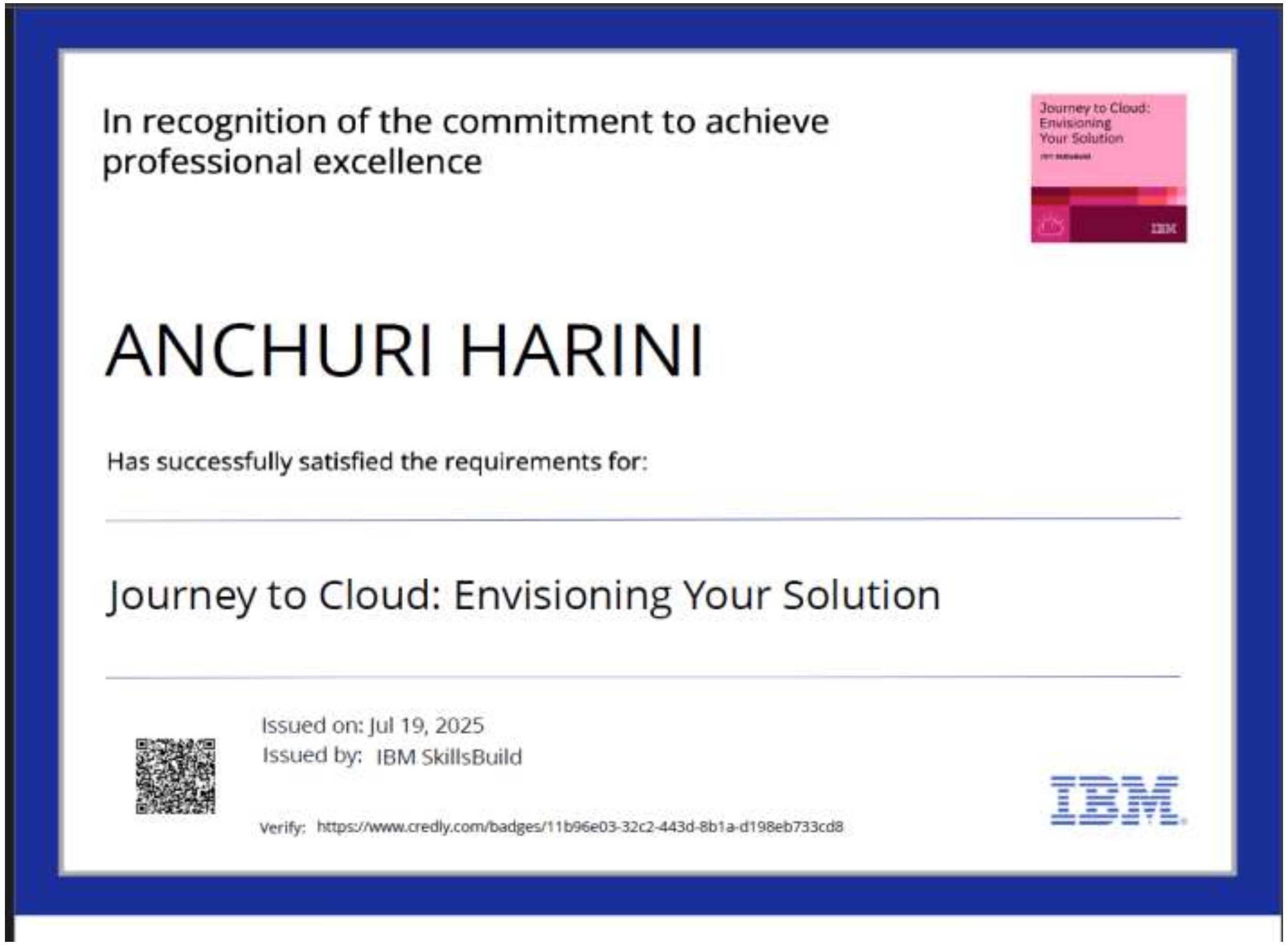
Issued on: Jul 19, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/4f99d4f9-dc53-4f4d-a4d5-72e083ae9977

IBM

edunet foundation

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

ANCHURI HARINI

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 19, 2025

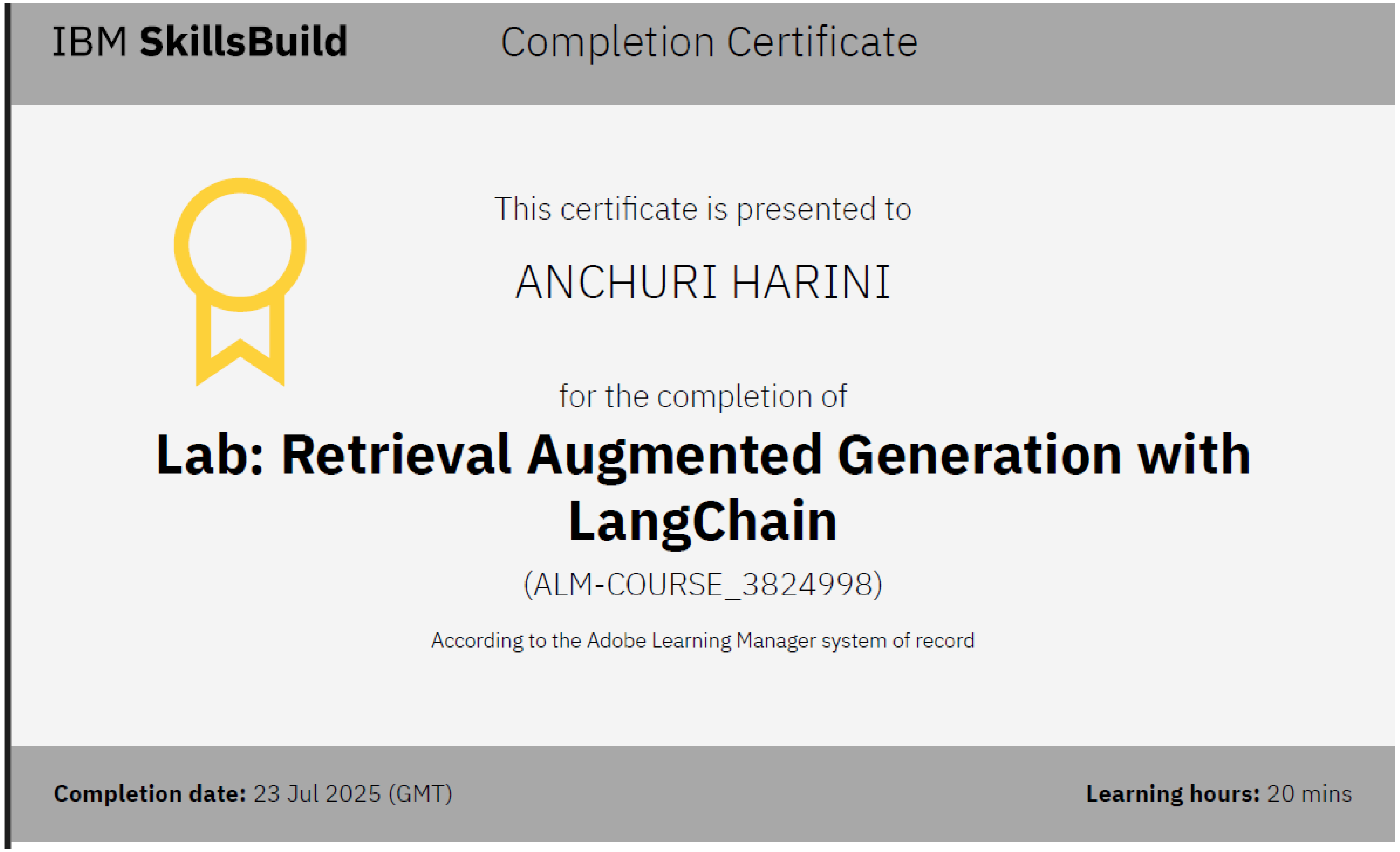
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/11b96e03-32c2-443d-8b1a-d198eb733cd8

IBM

edunet foundation

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

ANCHURI HARINI

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT) **Learning hours:** 20 mins

edunet foundation

THANK YOU