CAPSTONE PROJECT

NETWORK INTRUSION DETECTION SYSTEM (NIDS) USING MACHINE LEARNING

Presented By:
Name:-
Shrikrushna Deokar
College :-
Pimpri chinchwad college of engineering & Research
Department:-
Computer engineering

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.

PROPOSED SOLUTION

- The proposed system aims to address the challenge of securing communication networks by detecting cyber-attacks using machine learning. The solution includes the following key components:
- **Data Collection:**

  Use labeled network traffic data containing both normal and malicious activities

  Dataset includes various attack types like **DoS, Probe, R2L, and U2R**.
- **Data Preprocessing:**

  Clean the data by handling missing values and removing inconsistencies.

  Perform feature engineering to select the most important network traffic features for training.
- **Machine Learning Algorithm:**

  Train a binary classification model using **AutoAI in IBM Watsonx.ai Studio.**

  Algorithm used: **Snap Random Forest Classifier..**

  OOTPUT Lable : normal or anomaly.
- **Deployment:**

  Evaluate the model's performance using metrics like **accuracy** and **confidence level**.

  Enable real-time prediction of network traffic as safe or potentially harmful.

  Evaluate the model's performance using metrics like **accuracy** and **confidence level**.

SYSTEM APPROACH

Technologies Used:

- **Platform:** IBM Watsonx.ai Studio
- **Language:** Python (AutoAI/AutoML on IBM Cloud)
- **Libraries/Tools:** Pandas, Scikit-learn (in background), AutoML
- **Dataset:** Publicly available intrusion detection dataset with features like:
  - protocol_type, service, flag, src_bytes, dst_bytes, etc.

- **Training File Used:** Train_data.csv

  Testing File Used: Test_data.csv

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 Used

  Snap Random Forest Classifier (AutoAI selected)

  Ensemble method with high accuracy (99.5%)
- Training Process

  Used IBM Watson AutoAI with Train_data.csv

  Automated feature selection, preprocessing, and HPO

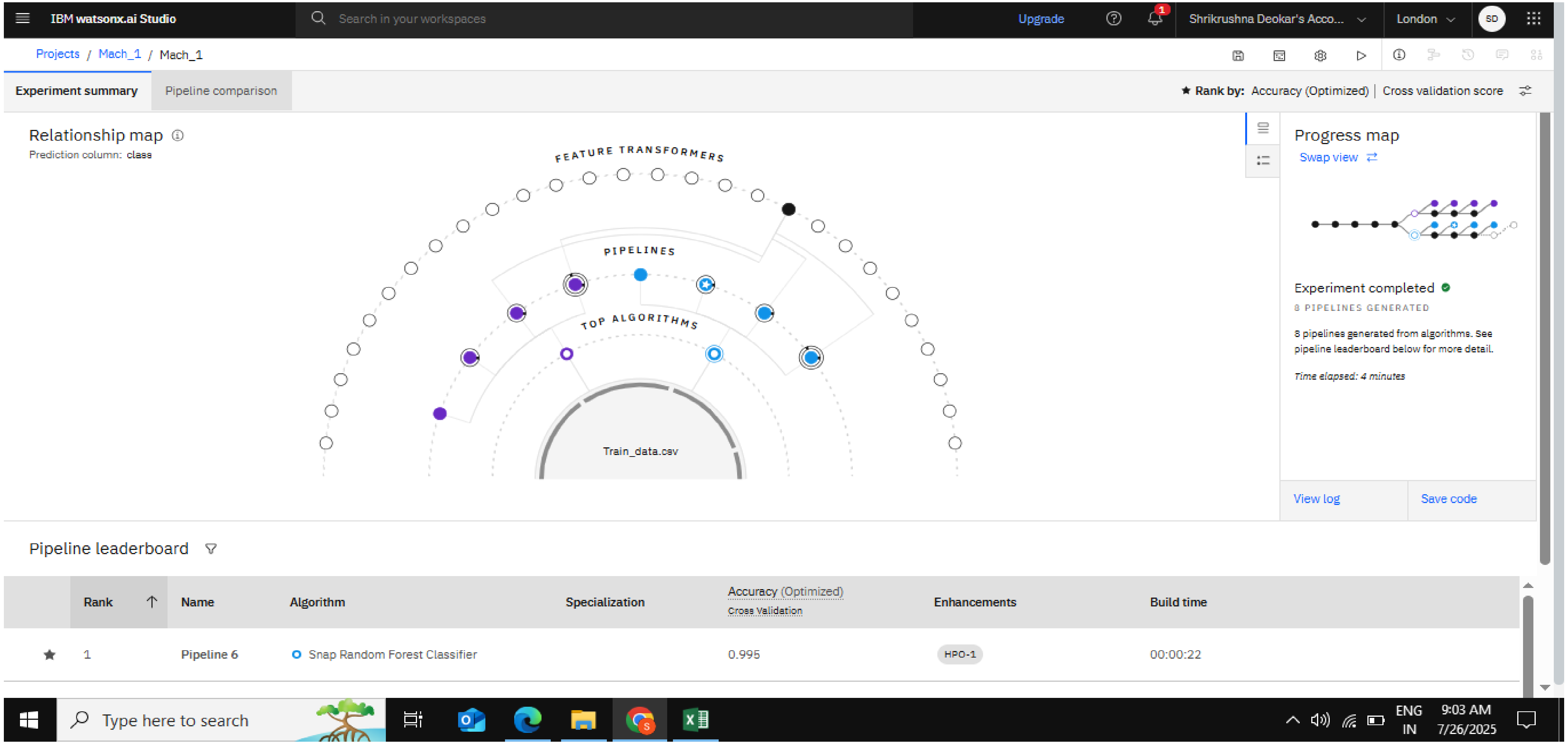
  Training completed in just 22 seconds
- Model Deployment

  Deployed via IBM Watson Machine Learning as REST API

  Supports real-time predictions and easy integration

RESULT

- Here we can train model using data



Experiment completed

8 PIPELINES GENERATED

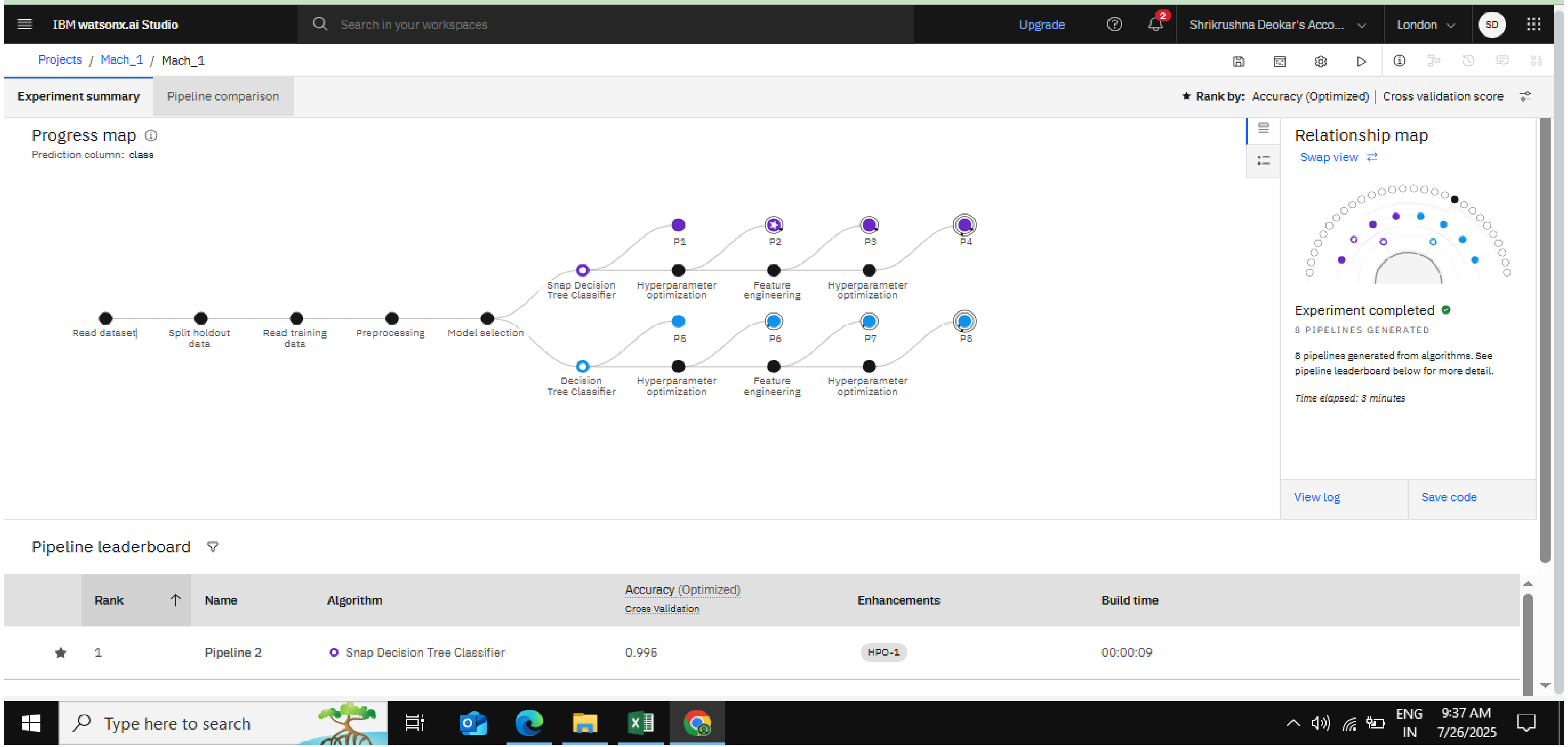
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 4 minutes

Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 6	○ Snap Random Forest Classifier		0.995	HPO-1	00:00:22

RESULT



Prediction column: class

Experiment completed

8 PIPELINES GENERATED

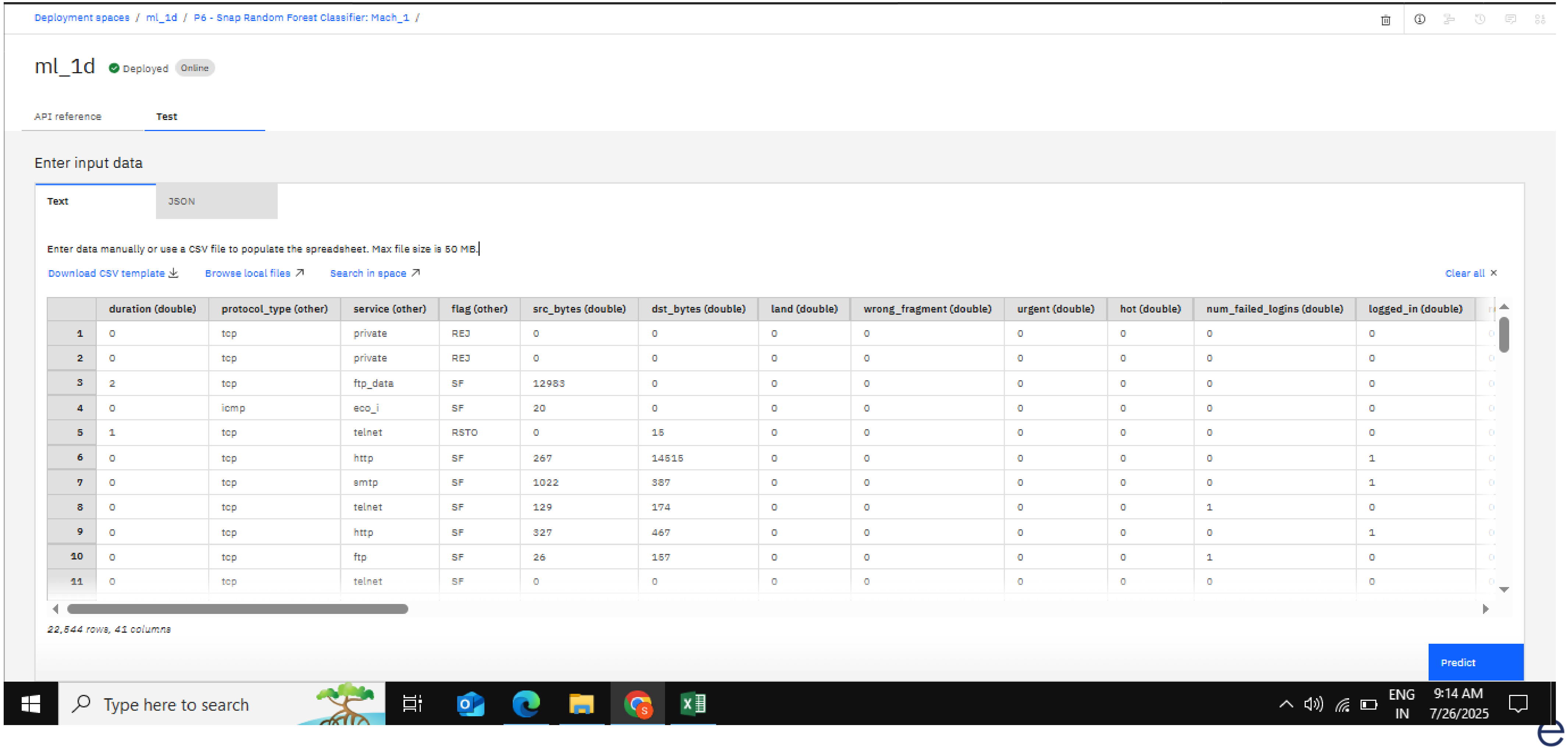
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 3 minutes

	Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:09

RESULT

Here we can upload data for test the model .



Deployment spaces / ml_1d / P6 - Snap Random Forest Classifier: Mach_1 /

ml_1d Deployed Online

API reference Test

Enter input data

Text JSON

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

Download CSV template Browse local files Search in space Clear all ×

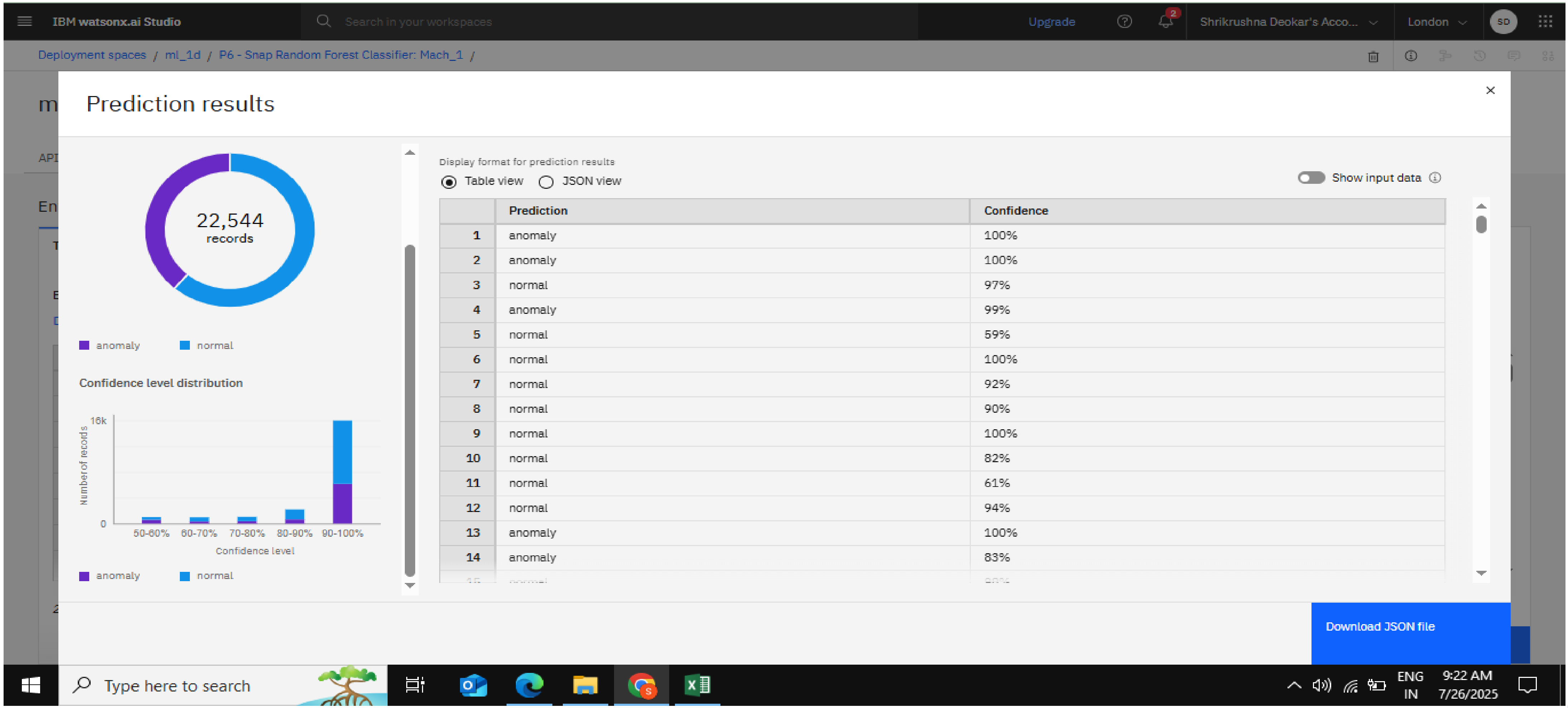
	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)	hot (double)	num_failed_logins (double)	logged_in (double)
1	0	tcp	private	REJ	0	0	0	0	0	0	0	0
2	0	tcp	private	REJ	0	0	0	0	0	0	0	0
3	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0
4	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0
5	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0
6	0	tcp	http	SF	267	14515	0	0	0	0	0	1
7	0	tcp	smtp	SF	1022	387	0	0	0	0	0	1
8	0	tcp	telnet	SF	129	174	0	0	0	0	1	0
9	0	tcp	http	SF	327	467	0	0	0	0	0	1
10	0	tcp	ftp	SF	26	157	0	0	0	0	1	0
11	0	tcp	telnet	SF	0	0	0	0	0	0	0	0

22,544 rows, 41 columns

Predict

RESULT

- Here prediction final results.



Prediction results

Display format for prediction results

● Table view ○ JSON view

	Prediction	Confidence
1	anomaly	100%
2	anomaly	100%
3	normal	97%
4	anomaly	99%
5	normal	59%
6	normal	100%
7	normal	92%
8	normal	90%
9	normal	100%
10	normal	82%
11	normal	61%
12	normal	94%
13	anomaly	100%
14	anomaly	83%

CONCLUSION

- We successfully developed and deployed a Machine Learning-based Network Intrusion Detection System (NIDS) capable of accurately identifying and classifying various types of cyber-attacks in real time. By analyzing network traffic patterns, the system can effectively distinguish between normal and anomalous activities. The use of ML enhances detection capabilities, enabling early warning and proactive responses to threats like DoS, R2L, and U2R attacks. Compared to traditional static rule-based systems, this approach offers greater flexibility and accuracy. Overall, the project demonstrates how AI can strengthen cybersecurity infrastructure and minimize potential damage.

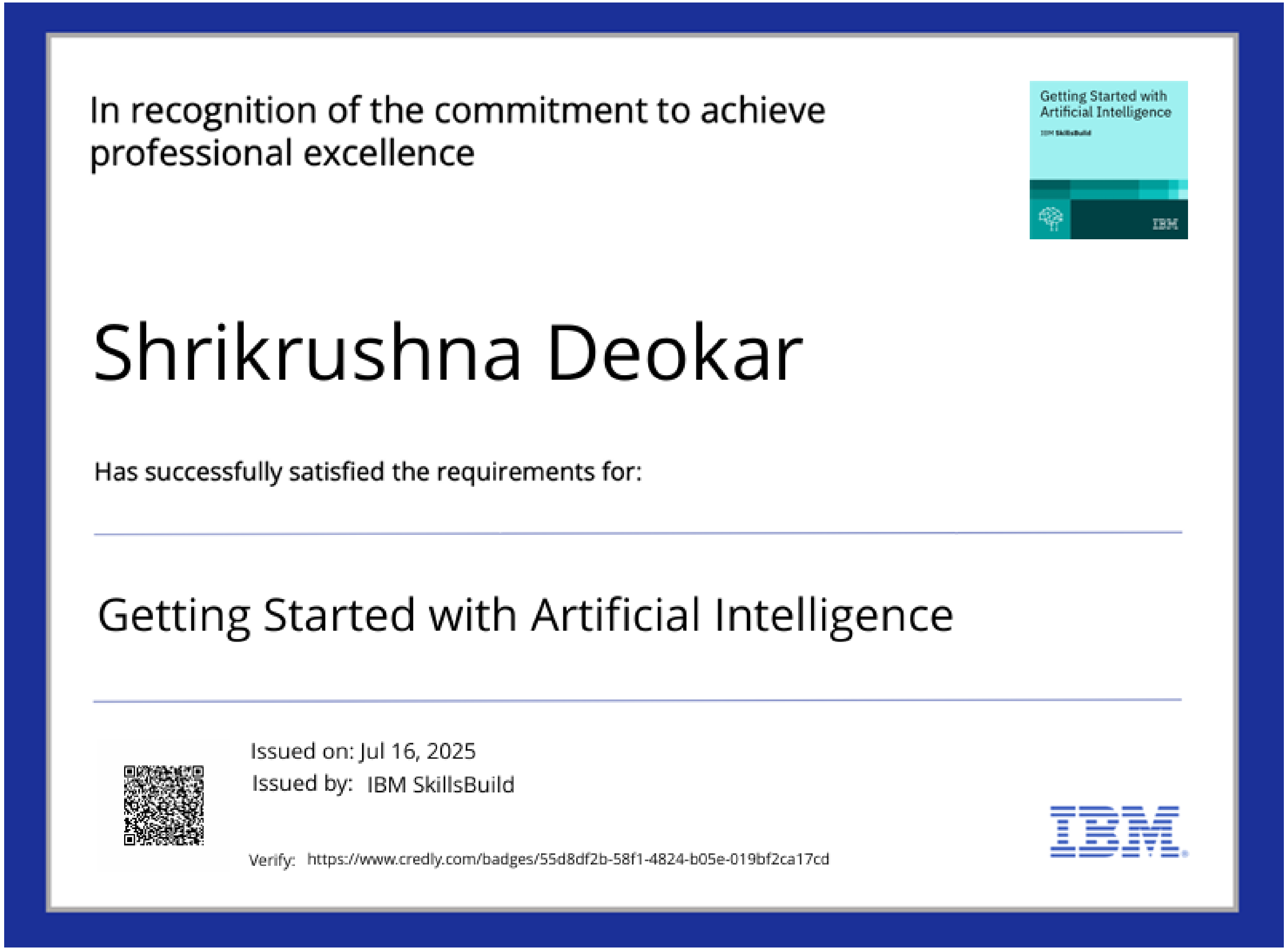
FUTURE SCOPE

- In the future, the system can be extended to perform multi-class classification, allowing it to detect and identify specific types of cyber-attacks such as DoS, R2L, U2R, and Probe. The integration of real-time alert mechanisms using streaming data technologies like Apache Kafka or Spark can help in prompt threat detection and response. Furthermore, the system can be connected to existing firewall and antivirus solutions to enable automatic threat mitigation. To enhance the detection accuracy and adaptiveness, advanced deep learning models like Convolutional Neural Networks (CNNs) or Long Short-Term Memory networks (LSTMs) can be explored and implemented.

REFERENCES

- IBM Watsonx.ai Documentation:
  https://www.ibm.com/docs/en/watsonx
- Dataset Source – KDD Cup 1999 (Network Intrusion Detection Dataset):
  http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
- Scikit-learn Machine Learning Library Documentation:
  https://scikit-learn.org/stable/documentation.html
- Cybersecurity Research Papers and Tutorials:
  IEEE Xplore, SpringerLink, ResearchGate
- Python Official Documentation:
  https://docs.python.org/3/

IBM CERTIFICATIONS



Getting Started with Artificial Intelligence
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Shrikrushna Deokar

Has successfully satisfied the requirements for:

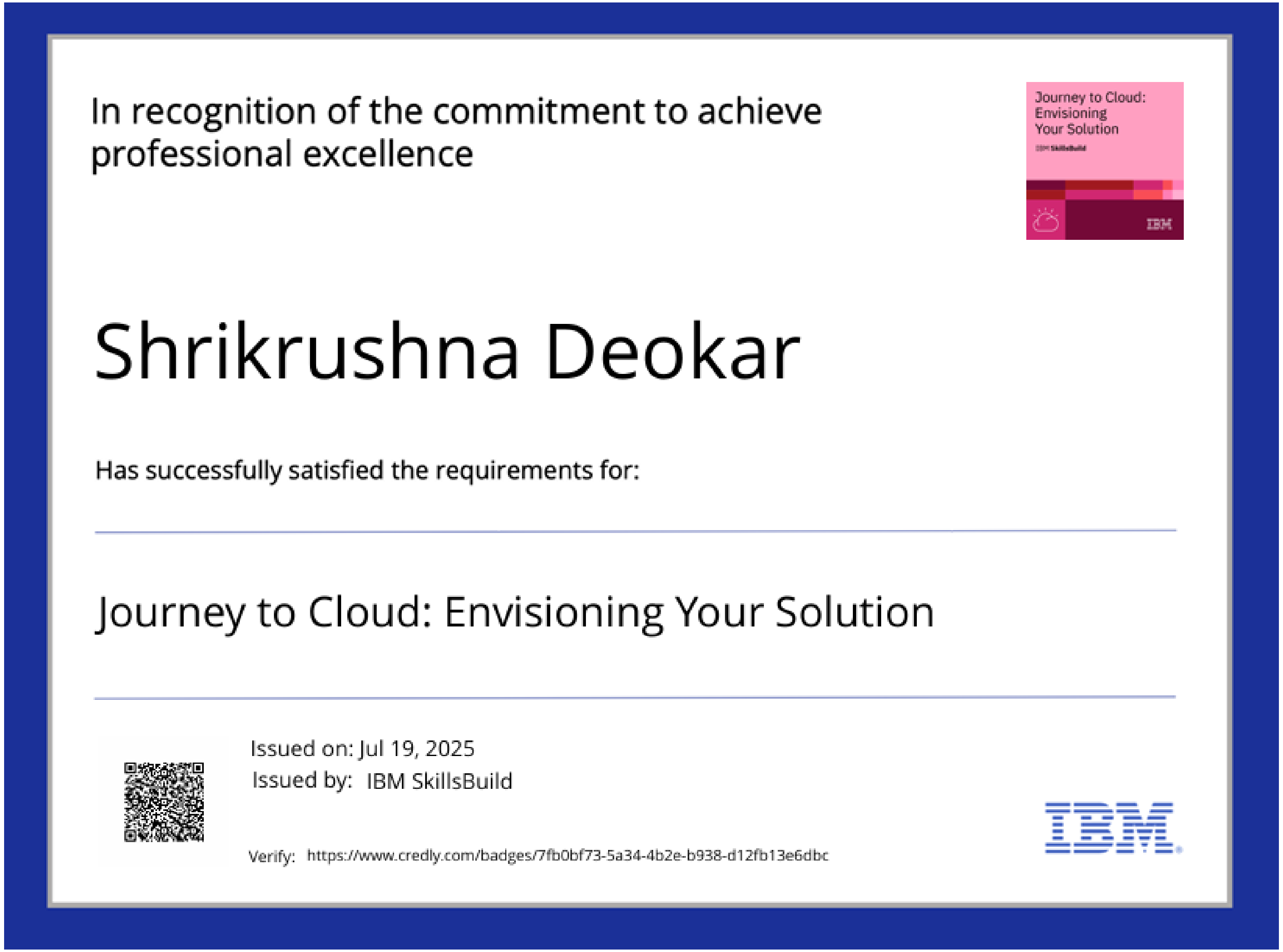
Getting Started with Artificial Intelligence

Issued on: Jul 16, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/55d8df2b-58f1-4824-b05e-019bf2ca17cd

IBM

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

Shrikrushna Deokar

Has successfully satisfied the requirements for:

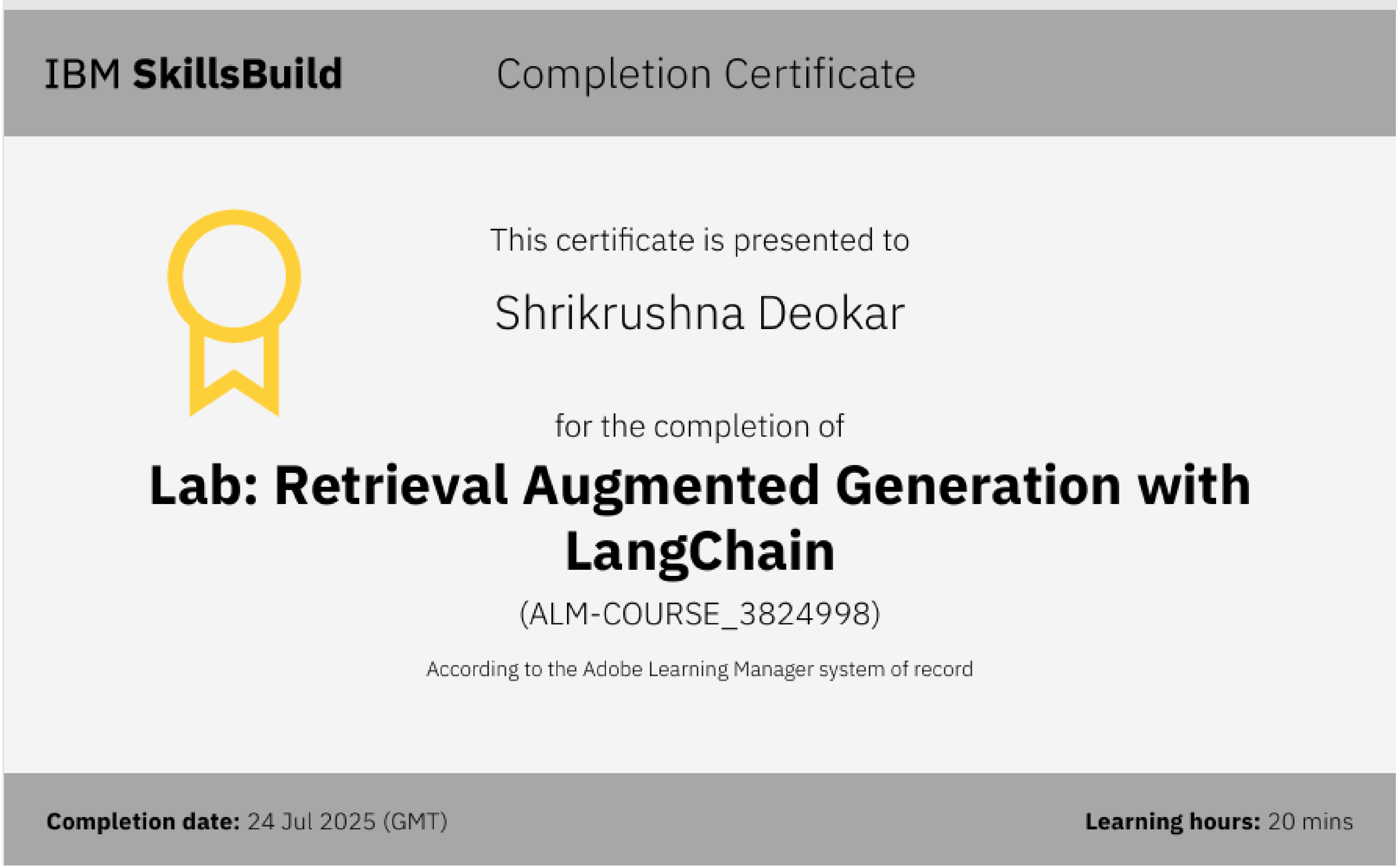
Journey to Cloud: Envisioning Your Solution

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

Verify: https://www.credly.com/badges/7fb0bf73-5a34-4b2e-b938-d12fb13e6dbc

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Shrikrushna Deokar

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