AICTE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
Student Name- Karan
College Name-Department - Haryana Engineering College (CSE)

edunet foundation

OUTLINE

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

PROBLEM STATEMENT

Example: 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 automate the detection and classification of power system faults using machine learning models built with IBM Watsonx AutoAI. The goal is to identify and classify various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) and The goal is to build a model that can effectively secure communication networks by providing an early warning of malicious activities.

The solution consists of the following components:

1. **Data Collection:**

- Use a public dataset containing various types of cyber-attacks (e.g., DoS, Probe, R2L, U2R) .
- The dataset includes labeled class, aiding supervised learning.

2. **Data Preprocessing:**

- Upload the dataset to IBM Cloud Object Storage.
- Configure columns and handle missing values directly in Watsonx AutoAI.

3. **Model Building using AutoAI:**

- AutoAI automatically explores and trains multiple ML pipelines.
- Selects the best-performing classifier (e.g., Random Forest, Logistic Regression, XGBoost).
- Evaluates models based on accuracy, precision, and recall.

PROPOSED SOLUTION

4. **Deployment**:

- The best model is promoted to a Deployment Space.
- A REST API is created to receive network data as input and return anamoly classification.

5. **Testing & Evaluation**:

- Use test JSON inputs to validate model performance.
- Observe model outputs in both table and JSON format.
- Analyze AutoAI pipeline leaderboard to choose optimal model

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and deploying a machine learning-based 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 **model using IBM Watsonx AutoAI**

1. System Requirements:

- IBM Cloud Lite Account
- IBM Watsonx.ai Studio
- IBM Cloud Object Storage
- Modern Web Browser (Chrome/Firefox)
- Stable Internet Connection
- Kaggle Dataset: Network Intrusion

2. Libraries / Services Used:

- IBM Watsonx AutoAI (no manual coding required)
- IBM Watson Machine Learning Runtime
- IBM Cloud Object Storage
- JSON for testing API input/output

3. Development Methodology:

- Create Watsonx.ai instance and associate storage
- Upload dataset (network intrusion dataset) to the project
- Define and configure an AutoAI experiment
- AutoAI automatically selects and trains ML pipelines
- Evaluate model performance via leaderboard
- Save and deploy the best-performing pipeline
- Generate and test API endpoint using JSON input

This system-oriented approach ensures a no-code, rapid ML solution using cloud-native IBM tools, making it suitable for scalable, real-time grid fault analysis.

ALGORITHM & DEPLOYMENT

This section describes the machine learning algorithm selection, training process, and deployment strategy used to detect and classify network intrusion with IBM Watsonx AutoAI.

Algorithm Selection:

IBM Watsonx AutoAI automatically explores multiple classification algorithms, such as:

- Logistic Regression
- Decision Trees
- Random Forest
- Gradient Boosted Trees (XGBoost)
- Ensemble Models

The final model is selected based on evaluation metrics like accuracy, precision, and recall. AutoAI uses automated hyperparameter tuning and pipeline optimization to determine the best-performing algorithm for the classification task.

Data Input:

- Protocol type , service ,flag and class (A, B, and C D)
- Output Label: class(e.g., Normal, Anomaly)

AutoAI automatically detects feature types and handles preprocessing like scaling, encoding, and missing values internally.

ALGORITHM & DEPLOYMENT

Training Process:

- AutoAI splits the dataset into training and validation sets.
- Applies automated feature transformation, model selection, and optimization.
- Evaluates multiple pipelines using cross-validation.
- Ranks them on a leaderboard based on performance metrics.

Prediction Process:

- The best model pipeline is saved and promoted to the deployment space.
- It is exposed as a REST API that accepts JSON input (voltage/current values).
- The model processes the input and returns the predicted fault type in real-time.

This pipeline ensures reliable and scalable fault classification without requiring manual coding, making it suitable for integration in real-world power monitoring systems.

RESULT

The machine learning model generated by IBM Watsonx AutoAI demonstrated high accuracy in detecting and classifying various fault types in the power distribution system.

Key Performance Metrics:

★	1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:10
	2	Pipeline 1	Snap Decision Tree Classifier	0.995	None	00:00:04
	3	Pipeline 6	Decision Tree Classifier	0.994	HPO-1	00:00:11

- Accuracy: [Insert actual % from leaderboard]
- Precision: High precision in identifying LG, LL, and LLG faults
- Recall: Strong recall values for multiclass classification
- F1 Score: Balanced performance across fault types

RESULT

Pipeline Leaderboard:

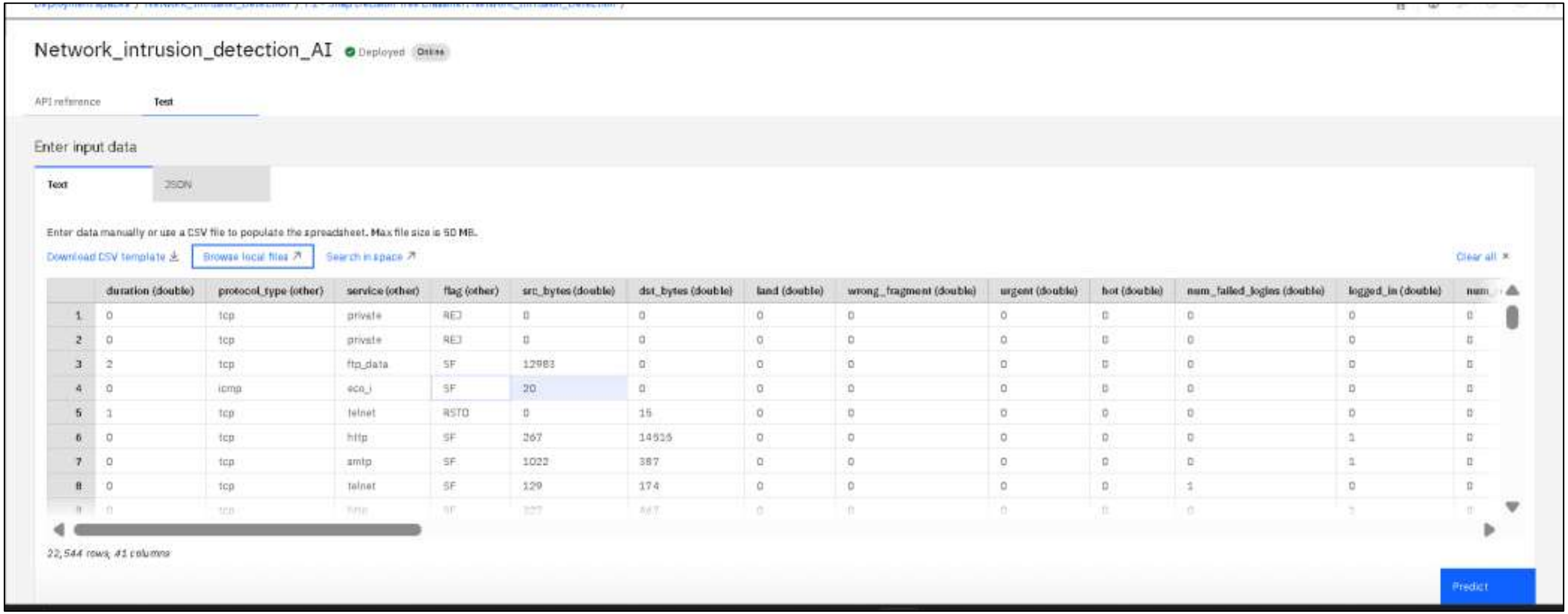
Pipeline leaderboard

	Rank	Name	Algorithm	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 2	Snap Decision Tree Classifier	0.995	HPO-1	00:00:10
	2	Pipeline 1	Snap Decision Tree Classifier	0.995	None	00:00:04
	3	Pipeline 6	Decision Tree Classifier	0.994	HPO-1	00:00:11
	4	Pipeline 5	Decision Tree Classifier	0.994	None	00:00:05

- AutoAI ranked multiple pipelines based on their validation scores
- The top pipeline was selected and saved for deployment

RESULT

Model Output Formats:



- Output Table View: Displays predicted anomaly class against test data

RESULT

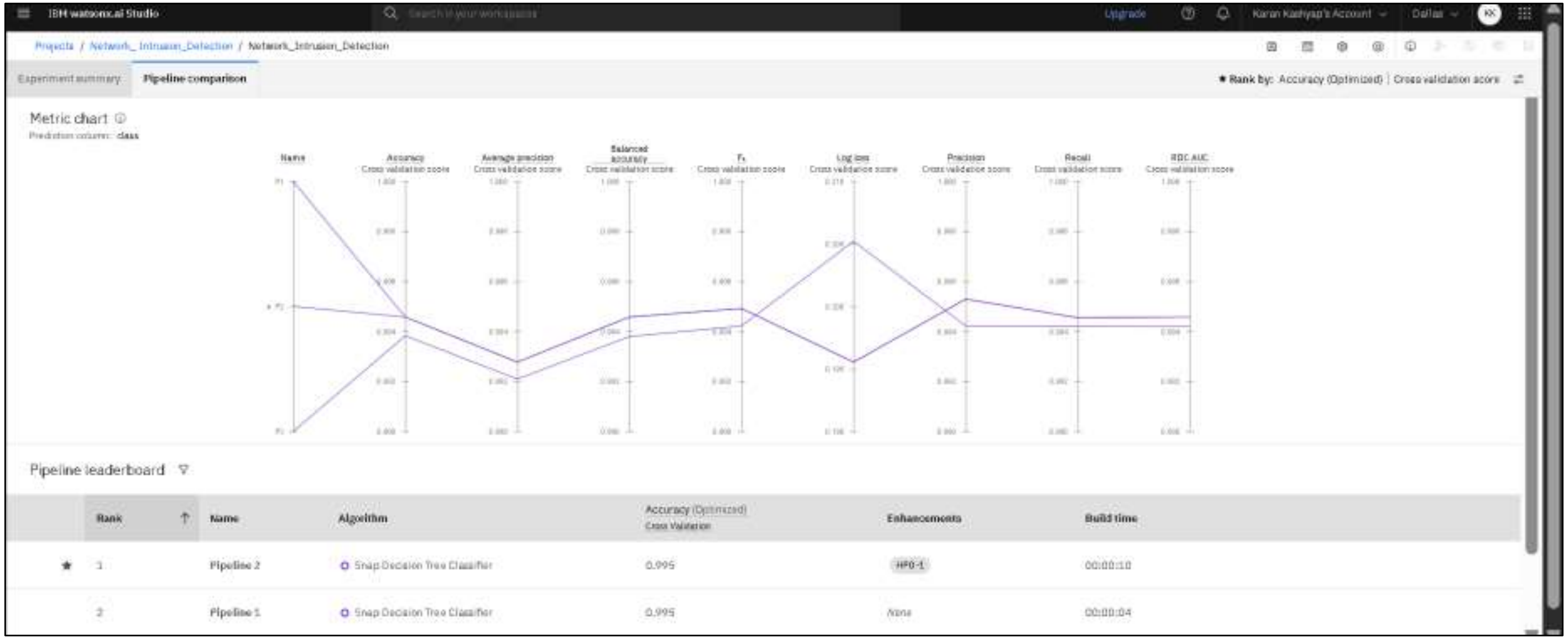
Model Output Formats:

- JSON Output View: Shows prediction results in API-compatible JSON format

RESULT

Visuals to Include (Screenshots):

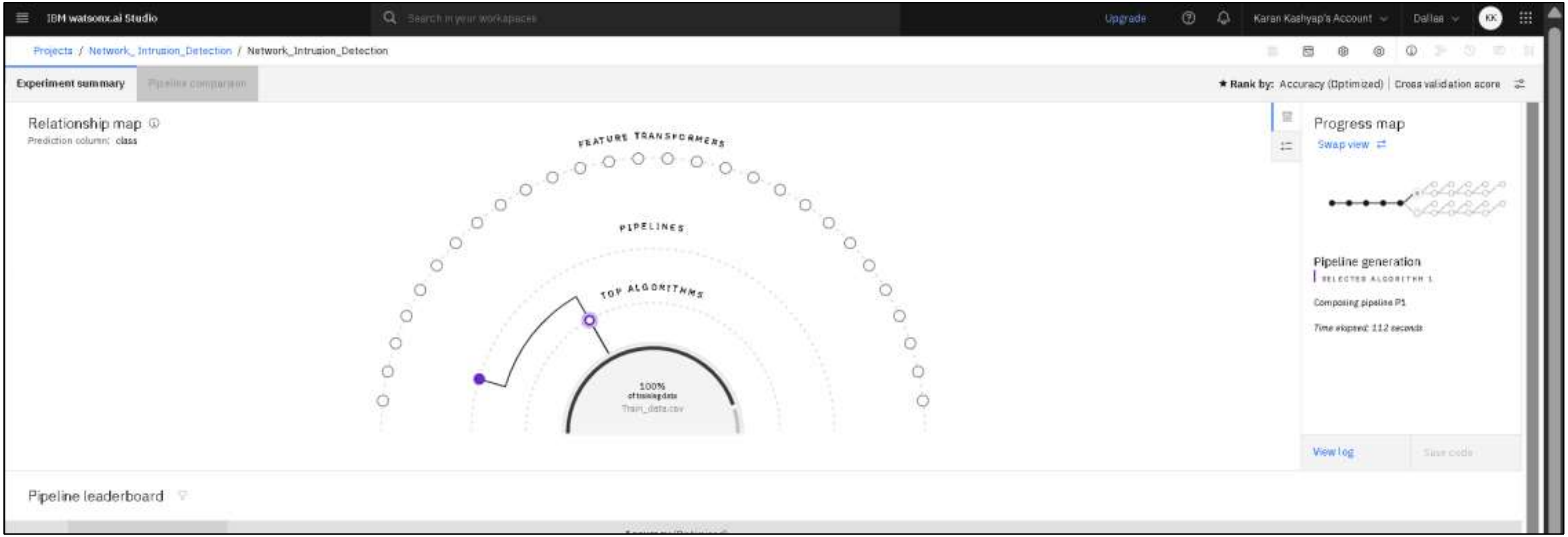
1. Progress Map



– AutoAI's visual representation of pipeline evolution

RESULT

2. Relationship Map



– How input features influence predictions

The model's deployment as an API allows real-time classification of hacking attacks.

CONCLUSION

- The developed Network Intrusion Detection System (NIDS) using **AutoAI in IBM Watson Studio** successfully automates the process of detecting malicious network activities. By leveraging machine learning algorithms automatically selected and optimized by AutoAI, the model is able to:
- Efficiently **classify network traffic** as normal or anomalous.
- Reduce the **manual effort of feature engineering and model selection**, thanks to AutoAI's automated pipeline generation.
- Provide **real-time or near real-time predictions**, enhancing cybersecurity measures in dynamic network environments.
- Achieve **high accuracy and reliability** while minimizing false alarms compared to traditional rule-based intrusion detection systems.
- This demonstrates the potential of **AI-driven intrusion detection** in strengthening network security infrastructure against evolving cyber threats.

FUTURE SCOPE

- Enhancing Data Quality and Balance:
  - Introduce more diverse datasets covering **latest cyberattack types**.
  - Use **data balancing techniques (SMOTE, class weights)** to improve anomaly detection accuracy.
- Feature Engineering Improvements:
  - Perform **manual domain-based feature selection** alongside AutoAI's automated approach to reduce noise.
  - Explore advanced feature extraction techniques (e.g., embeddings for categorical features).
- Model Explainability:
  - Use **SHAP or LIME** to interpret how the model classifies anomalies, increasing trust and transparency.
- Real-Time Deployment:
  - Integrate with **live network monitoring systems** to enable **streaming intrusion detection**.
  - Deploy as a **cloud-based API** for scalable use across multiple networks.
- Hybrid and Ensemble Approaches:
  - Combine AutoAI's supervised model with **unsupervised anomaly detection techniques** (Autoencoders, Isolation Forest) for unknown attack patterns.
- Continuous Learning:
  - Implement **online learning or periodic retraining** so the model adapts to **new cyber threats** without manual intervention.
- Integration with Security Tools:
  - Connect with **firewalls, SIEM tools (e.g., Splunk, QRadar)** for automatic threat response and mitigation.

;

REFERENCES

1. Kaggle Dataset:

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

2. IBM Cloud Documentation:

- IBM Watsonx.ai and AutoAI Official Guide
- https://www.ibm.com/cloud/watsonx

3. 4. 5. AutoAI Model Evaluation Guide:

- “Model Evaluation Metrics in IBM AutoAI” – IBM Developer Documentation
- https://cloud.ibm.com/docs/autoai?topic=autoai-evaluation

REFERENCES

6. JSON Format Testing for ML APIs:

- IBM Watson Machine Learning API Reference
- https://cloud.ibm.com/apidocs/machine-learning

7. Book Reference (optional):

- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013).
- *An Introduction to Statistical Learning: With Applications in R*. Springer.

IBM CERTIFICATIONS

- Screenshot/ credly certificate(getting started with AI)

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

Karan Kashyap

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 15, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/451f5d3f-8da1-4f67-aaeb-029dce036c80

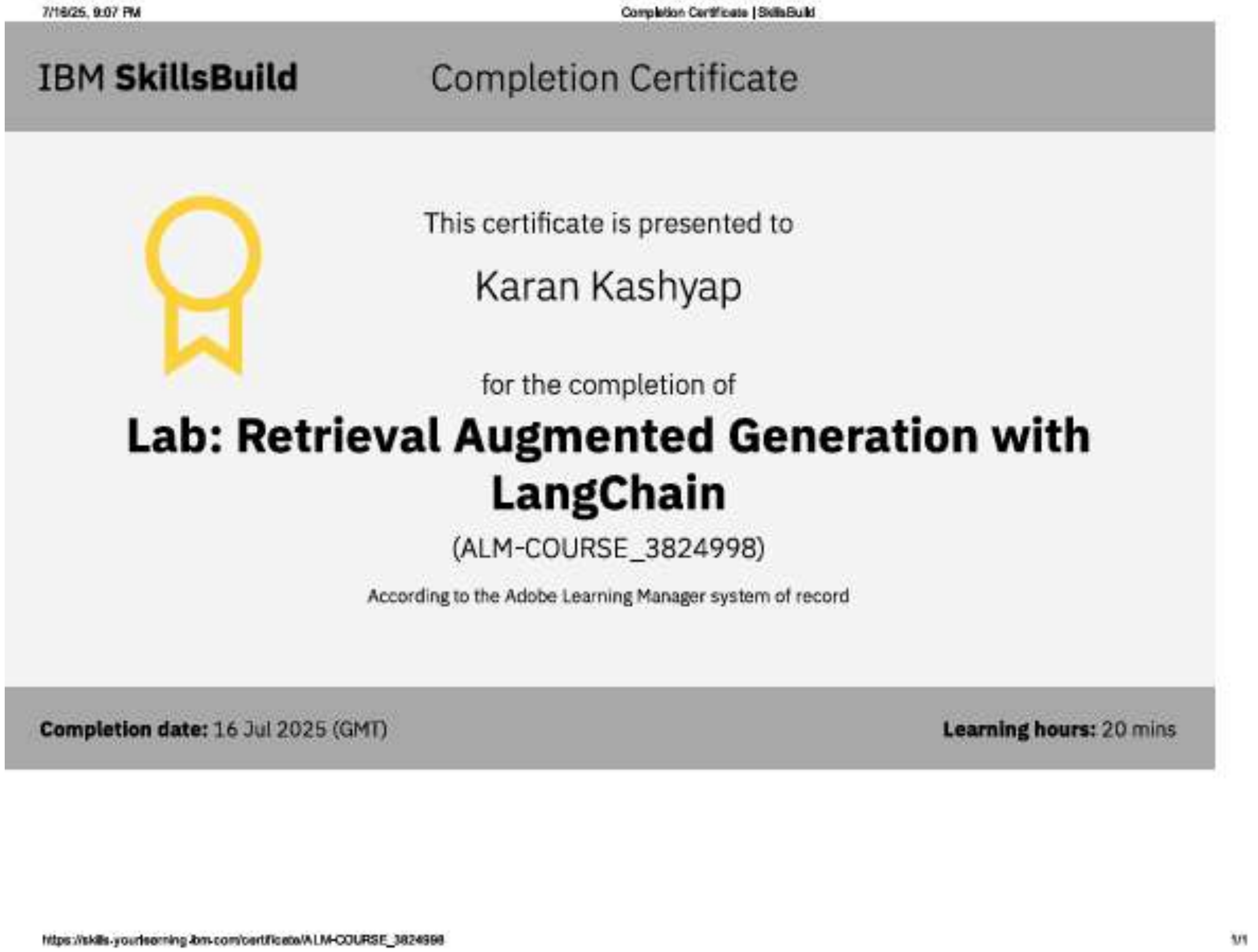
IBM

IBM CERTIFICATIONS

- Screenshot/ credly certificate(Journey to Cloud)

IBM CERTIFICATIONS

- Screenshot/ credly certificate(RAG Lab)



7/16/25, 9:07 PM Completion Certificate | SkillsBuild

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Karan Kashyap

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 16 Jul 2025 (GMT) **Learning hours:** 20 mins

https://skills.yourlearning.ibm.com/certificate/ALM-COURSE_3824998 1/1

edunet foundation

GITHUB

- GitHub Linkt:-
  - https://github.com/opmk1234/Network-Intrusion-Detection.git

edunet
foundation

THANK YOU