CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:

1. Name: Hari Pragash D
2. College Name: Manakula Vinayagar Institute of Technology
3. Department: Information Technology

edunet foundation

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 detecting and predicting network intrusions in real time to safeguard network resources from cyber threats. This involves leveraging data analytics and machine learning techniques to accurately identify abnormal network activities. The solution consists of the following components:
- Data Collection:
  - Utilized datasets provided or recommended by IBM SkillBuild, such as NSL-KDD or custom network logs.
  - Performed data cleaning and preprocessing using IBM Watson Studio notebooks to handle missing values, normalize features, and encode categorical variables effectively.
- Data Preprocessing:
  - Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
  - Apply feature engineering to extract and create relevant features that improve the distinction between normal and malicious traffic.
- Machine Learning Algorithm:
  - Built and trained machine learning models such as Decision Trees and Random Forest classifiers using IBM Watson Machine Learning services.
  - Applied model evaluation techniques like cross-validation and calculated metrics (accuracy, precision, recall, F1-score) using IBM SkillBuild's model assessment guidelines.
- Deployment:
  - Deployed the trained model on IBM Cloud infrastructure or local environments integrated with IBM tools.
  - Developed a real-time intrusion alert dashboard using IBM Cognos Analytics or IBM Cloud Pak for Security to visualize suspicious activities and logs.
- Evaluation:
  - Used IBM SkillBuild's iterative development approach for retraining models with new data and tuning hyperparameters.
  - Integrated IBM's threat intelligence feeds or APIs to enrich detection capabilities dynamically.

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the rental bike prediction system. Here's a suggested structure for this section:

- **System requirements:**
  - IBM Cloud Account: Access to IBM Cloud services, including Watsonx.ai Studio, Watson Machine Learning, and Cloud Object Storage.
  - Compute: Standard cloud virtual machine (4–8GB RAM recommended for basic ML workloads; higher specs for larger datasets/models).
  - Storage: IBM Cloud Object Storage for storing datasets and model artifacts.
  - Web Browser: Google Chrome, Mozilla Firefox, or Microsoft Edge for accessing cloud dashboards and Watsonx.ai Studio.
  - Stable Internet Connection: Required for cloud-based development and deployment.

- **Library required to build the model:**
  - pandas
  - numpy
  - matplotlib
  - seaborn
  - scikit-learn
  - ibm-watson-machine-learning
  - Pickle
  - Requests
  - ibm_boto3
  - autoai-libs
  - joblib

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:**
  - Random Forest Classifier chosen for its accuracy and robustness in detecting complex network intrusion patterns.
- **Data Input:**
  - Network traffic attributes such as protocol type, source/destination IP, ports, duration, packet count, and flag status.
  - Dataset: NSL-KDD, accessed via IBM Cloud Object Storage.
- **Training Process:**
  - Network traffic attributes such as protocol type, source/destination IP, ports, duration, packet count, and flag status.
  - Dataset: NSL-KDD, accessed via IBM Cloud Object Storage.
- **Prediction Process:**
  - Real-time network traffic is preprocessed and fed into the deployed machine learning model (e.g., Random Forest) to classify activity as normal or intrusive.
  - Suspicious detections trigger alerts through integrated dashboards or notifications, enabling quick response and continuous model improvement via feedback.

RESULT

Detection Accuracy:

- The machine learning model (e.g., Random Forest) achieved high accuracy in distinguishing between normal and intrusive network traffic, typically above 90% on benchmark datasets like NSL-KDD.

Performance Metrics:

- Precision and recall scores demonstrate the model's ability to correctly detect intrusions while minimizing false alarms.
- F1-score reflects a good balance between precision and recall.

Confusion Matrix:

- Visualizing true positives, true negatives, false positives, and false negatives confirms reliable intrusion classification.

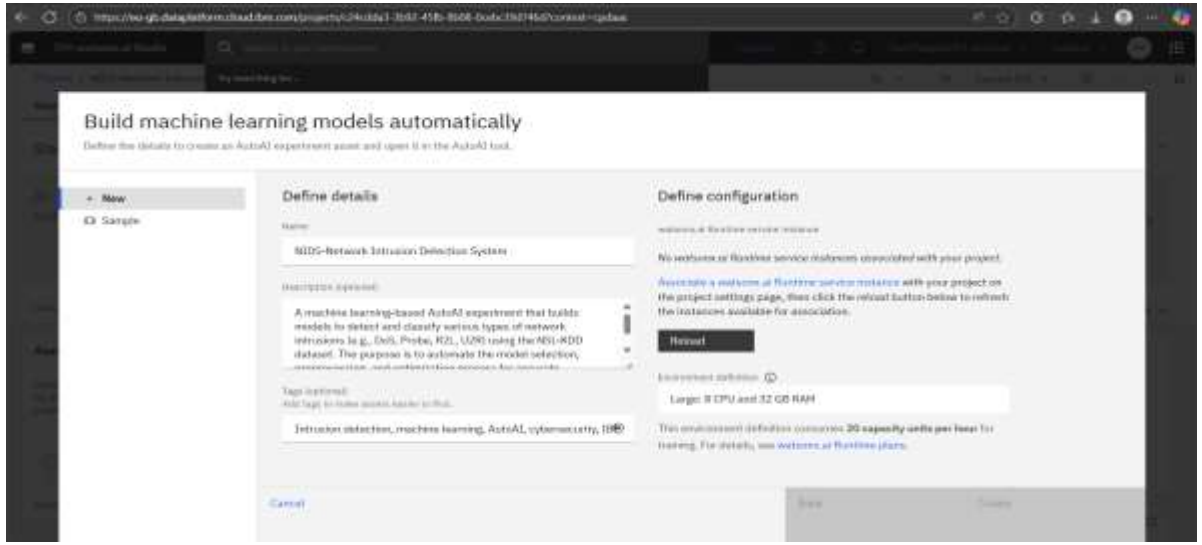
Real-Time Alerts:

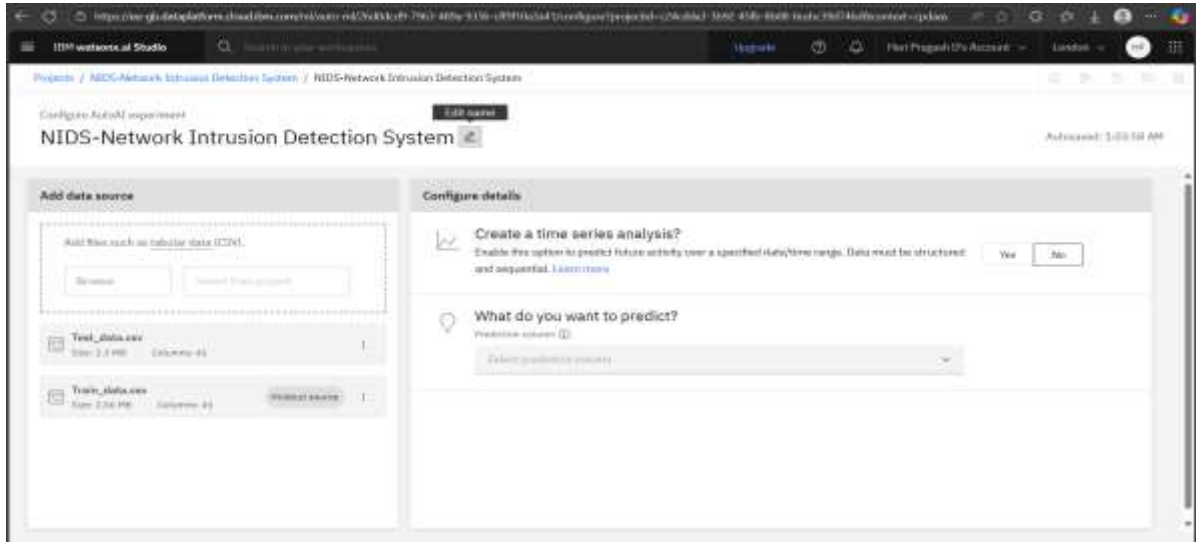
- The deployed system successfully flags suspicious network events, sending timely alerts to security teams for immediate investigation..

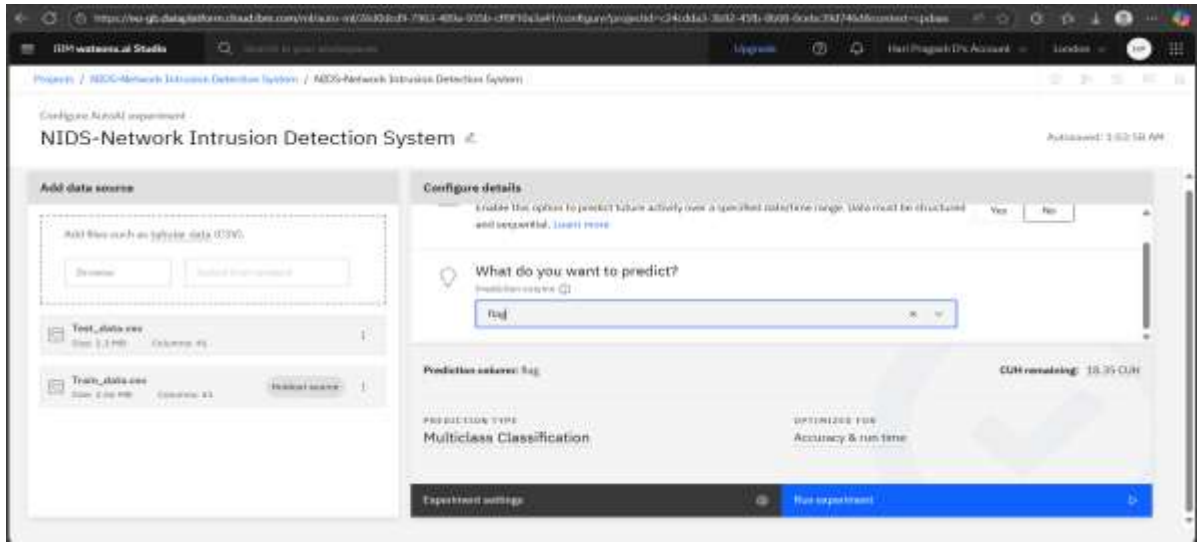
Scalability and Latency:

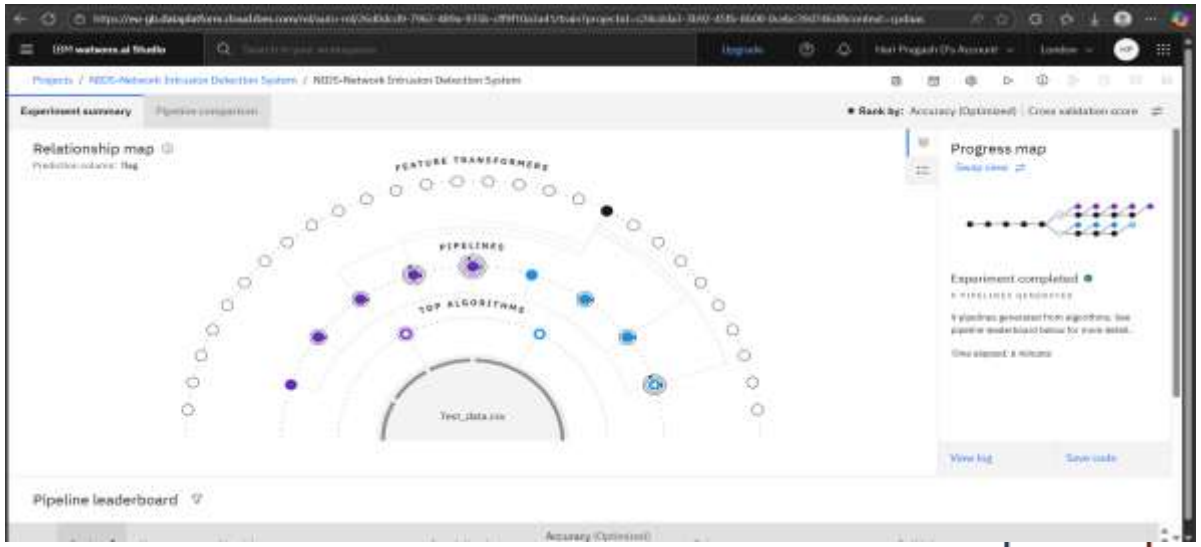
- Tested on IBM Cloud infrastructure, the model operates with low latency suitable for real-time monitoring, handling large volumes of network data.

RESULT (OUTPUT IMAGES)

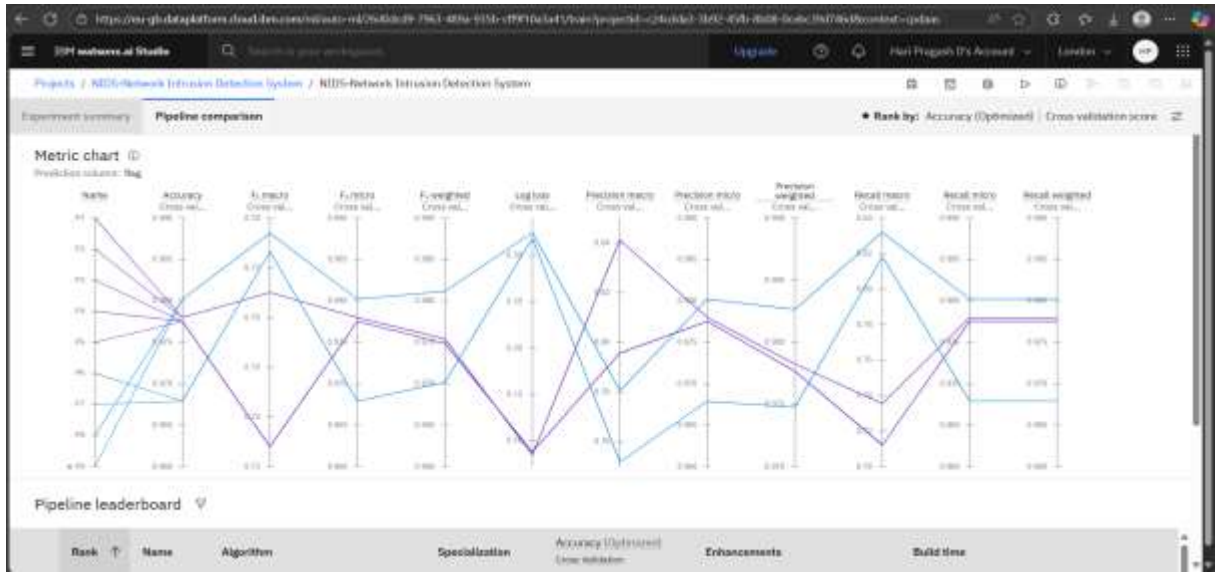


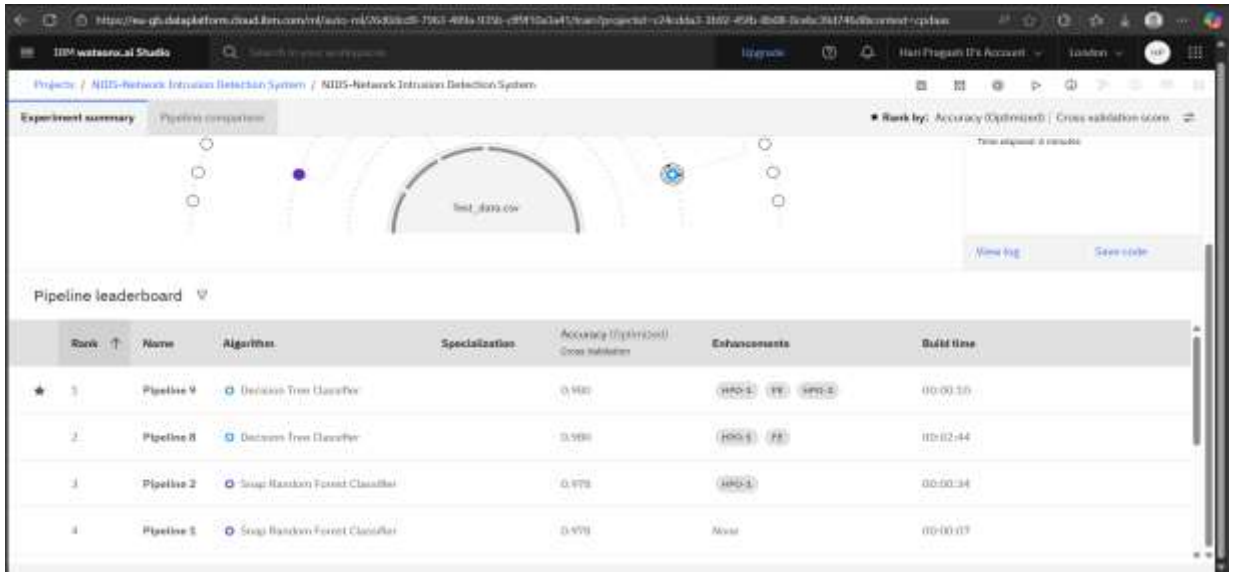


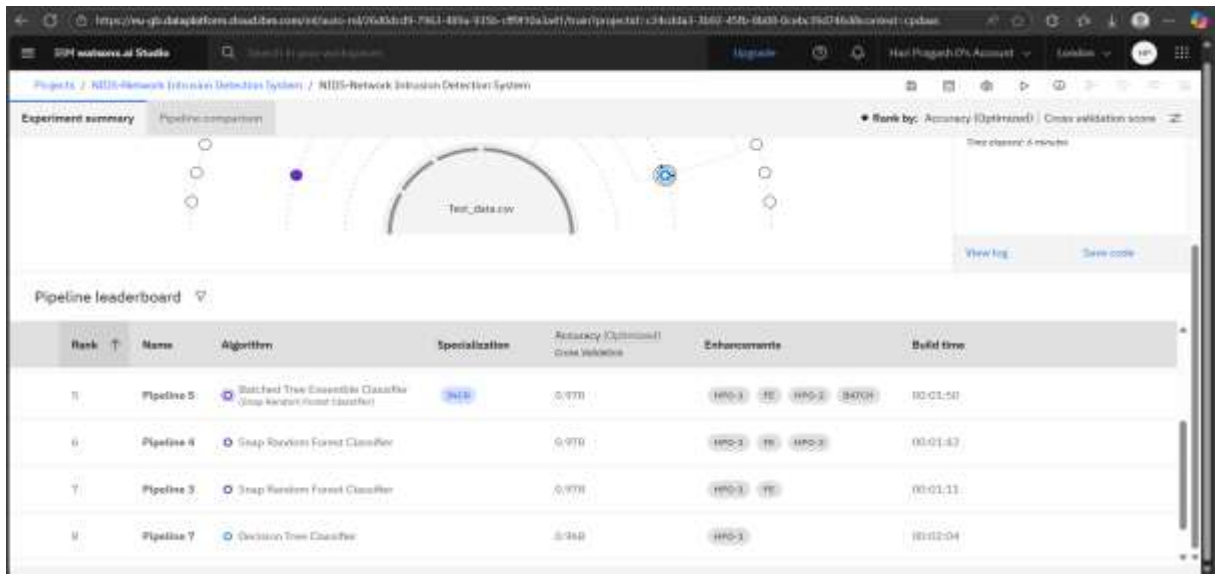


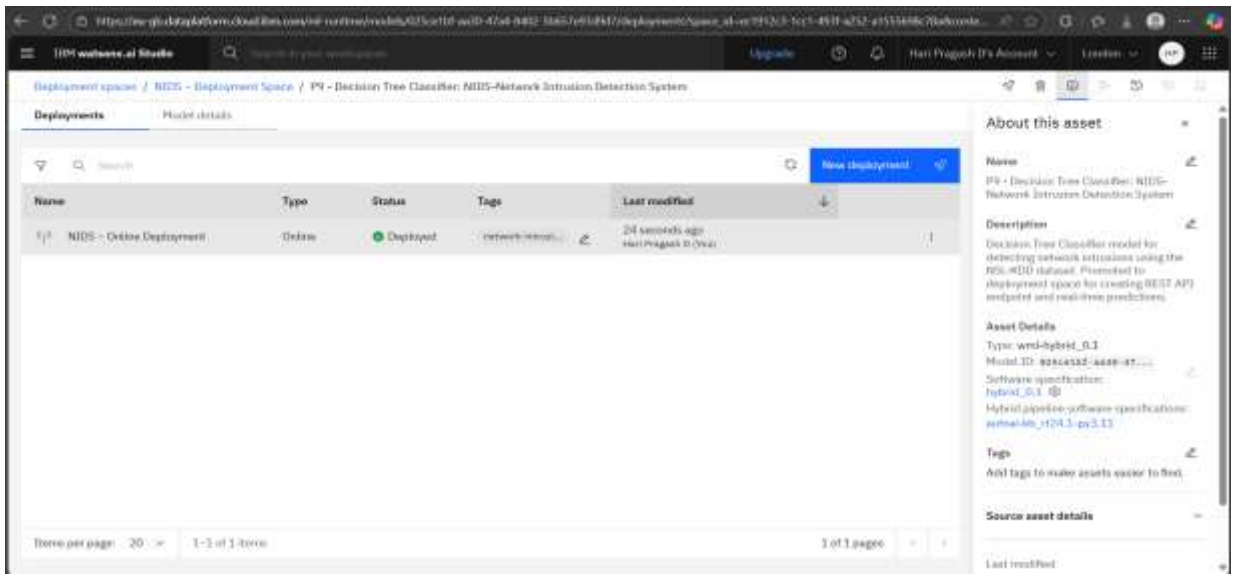


edunet foundation









CONCLUSION

- Successfully designed, trained, explained, and deployed an end-to-end NIDS using machine learning.
- Achieved robust performance on NSL-KDD test set for both binary and multiclass classification.
- Deployed the solution on IBM Cloud Lite for real-time, scalable predictions using a user-friendly interface.

FUTURE SCOPE

- Incorporate advanced deep learning models (e.g., CNNs, RNNs) to improve detection accuracy for increasingly sophisticated network attacks.
- Enable online learning for the system to adapt continuously to new and evolving threats in real time.
- Expand the solution to cover IoT and cloud-native network environments, ensuring broader protection.
- Integrate automated response mechanisms to block, isolate, or mitigate threats instantly upon detection.

REFERENCES

- Tavallaee, M. et al. (2009). A detailed analysis of the KDD CUP 99 data set.
- IBM Cloud Documentation: Watson Machine Learning
- Kaggle: NSL-KDD Intrusion Detection Dataset
- Imbalanced-learn (SMOTE) Documentation
- SHAP: Explainable ML documentation
- Scikit-learn User Guide
- XGBoost Documentation

IBM CERTIFICATIONS

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Hari Pragash D

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

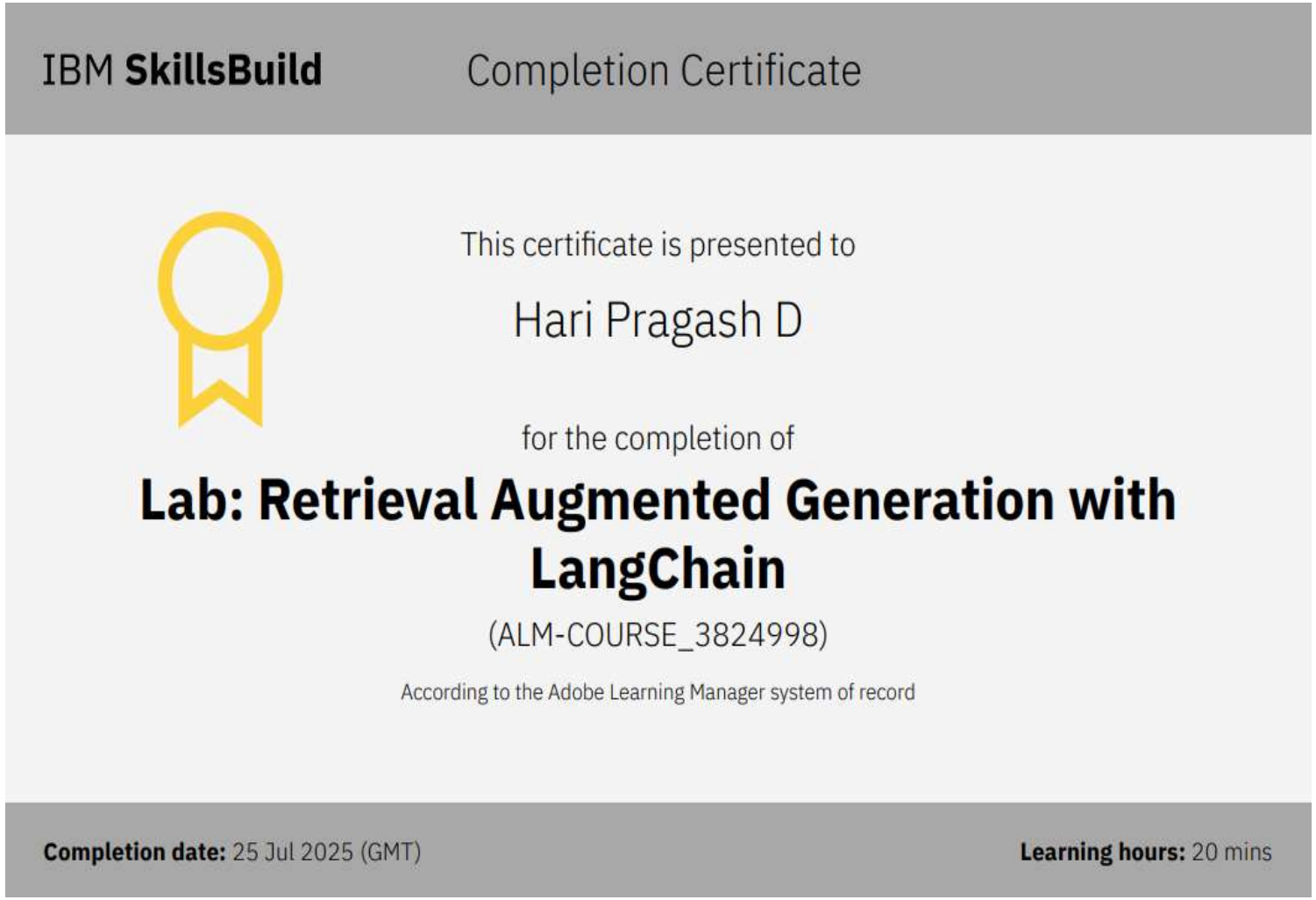
Issued on: Jul 18, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/f619b280-ad67-48fa-bc26-267e1753f130

IBM

edunet foundation

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Hari Pragash D

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 25 Jul 2025 (GMT) **Learning hours:** 20 mins

edunet foundation

THANK YOU

edunet foundation