CAPSTONE PROJECT

NETWORK INTRUSION DETECTION

Presented By:
1. B. V. D. S. Karthikeyan- Velagapudi Ramakrishna Siddhartha Engineering College- CSE

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

In today’s digitally connected world, securing communication networks against malicious activities is more critical than ever. Modern networks are frequently targeted by a variety of cyber-attacks that aim to compromise data, disrupt services, or gain unauthorized access to sensitive information. These attacks can range from Denial of Service (DoS), which overwhelms a system to make it unavailable, to more covert threats like Probing attacks that scan for vulnerabilities, Remote to Local (R2L) attacks where external users attempt unauthorized access, and User to Root (U2R) attacks that aim to gain privileged access within a system. Traditional security mechanisms often struggle to detect these increasingly sophisticated and evolving threats in real-time. As the complexity and volume of network traffic grow, so does the difficulty in distinguishing between normal activity and malicious behavior. This presents a significant challenge in the domain of network security—how to develop a system that can efficiently analyze traffic patterns and reliably identify potential intrusions. The goal is to enhance the detection and classification of abnormal activities within network environments to protect critical infrastructure and maintain data integrity.

PROPOSED SOLUTION

The proposed system focuses on building a machine learning-based intrusion detection model that classifies network traffic as either normal or anomalous. Leveraging IBM Watsonx.ai's AutoAI service, the system automates data preprocessing, model selection, training, and deployment, enabling faster and more reliable security analytics in communication networks. The solution will consist of the following components:

Data Collection:

- The dataset used is sourced from Kaggle, based on the widely used NSL-KDD network intrusion detection format.
- It contains network traffic records, each labeled as either "normal" or "anomaly".
- The dataset includes 41 features that describe:
- Connection properties (e.g., duration, src_bytes, dst_bytes)
- Protocol details (e.g., protocol_type, service, flag)
- Traffic behavior metrics (e.g., serror_rate, rerror_rate, count, srv_count)
- Each data point represents a snapshot of a network connection captured for analysis.

Data Preprocessing:

- The dataset was uploaded to IBM Cloud Object Storage and linked to Watsonx.ai using DataConnection, enabling seamless integration with AutoAI.
- AutoAI automatically handled encoding of categorical features like protocol_type, service, and flag, and applied scaling/normalization to numerical features.
- Duplicate records were removed by enabling drop_duplicates=True in the experiment metadata; missing values were not present in the dataset.
- The data was split into training and holdout sets (90/10), and feature selection was performed automatically to retain only the most relevant inputs for classification.

PROPOSED SOLUTION

Machine Learning Algorithm:

- IBM Watsonx.ai AutoAI was used to automate model selection, training, and evaluation for classifying network traffic as either normal or anomaly.
- Multiple machine learning algorithms were explored, including Decision Tree and Random Forest, as part of AutoAI's automated pipeline generation.
- Each algorithm was evaluated using cross-validation, and performance was measured using the accuracy metric on a holdout test set (10% of the data).
- While both models performed well, the Random Forest Classifier achieved the highest accuracy and was selected as the final model.
- Random Forest, being an ensemble learning algorithm, constructs multiple decision trees and aggregates their predictions, resulting in greater robustness and reduced overfitting.
- AutoAI also handled hyperparameter tuning (e.g., number of trees, max depth) internally and saved the best pipeline, which was later deployed as a web service for real-time prediction and scoring.

Deployment:

- The best-performing model (Random Forest Classifier) was promoted to a deployment space in IBM Watsonx.ai after automatic selection by AutoAI.
- It was then deployed as a web service, allowing real-time classification of new network traffic as either "normal" or "anomaly".
- This deployment enables easy integration with external systems and supports scalable, cloud-based prediction for intrusion detection.

SYSTEM APPROACH

System Requirements:

- **Hardware Requirements:**
  - Processor: Intel Core i5/i7 or equivalent (for local runs; IBM Cloud handles this in Watsonx.ai)
  - RAM: Minimum 8 GB (16 GB recommended)
  - Storage: At least 1 GB of free space (if working locally)
  - Internet: Stable broadband connection (for accessing IBM Watson Cloud)
- **Software Requirements:**
  - Operating System: Windows 10 / 11, Linux, or macOS
  - Browser: Latest version of Chrome/Firefox (for accessing IBM Watsonx.ai dashboard)
  - Python Version: 3.11 (used in AutoAI Notebook)

SYSTEM APPROACH

Libraries Required to Build the Model:

These libraries were either preinstalled in Watsonx.ai or installed via pip in the notebook:

- ibm-watsonx-ai: To interact with Watsonx.ai services
- autoai-libs: Required for AutoAI pipelines and helpers
- lale: For pipeline composition and operator tuning
- scikit-learn==1.3.*: Core ML algorithms and model evaluation
- xgboost==2.0.*: Gradient boosting model
- lightgbm==4.2.*: Light Gradient Boosting Machine
- snapml==1.14.*: IBM Snap ML for accelerated training
- pandas, numpy: For data manipulation and analysis
- matplotlib, seaborn: For visualization (optional but useful)
- getpass: To securely handle API key input

ALGORITHM & DEPLOYMENT

Algorithm Selection:

The project required a machine learning model capable of distinguishing between normal and anomalous network behavior across complex and high-dimensional data. Several classification algorithms were evaluated using IBM Watsonx AutoAI, including Decision Tree, Logistic Regression, and Gradient Boosting Classifiers. Ultimately, Random Forest Classifier was selected due to its superior performance metrics, especially its ability to handle imbalanced data, reduce variance, and deliver consistent results in multi-feature environments. It operates as an ensemble method, building multiple decision trees and combining their predictions for more accurate and generalized classification. The algorithm also provides feature importance scores, which is valuable for understanding which attributes (such as src_bytes, service, or count) contribute most to detecting anomalies. Its ability to resist overfitting while maintaining high accuracy made it the ideal choice for this intrusion detection application.

Data Input:

- The input features used by the model consist of various network traffic attributes, including:
- Basic features like duration, protocol_type, service, and flag
- Traffic volume metrics like src_bytes and dst_bytes
- Statistical features such as count, serror_rate, and srv_diff_host_rate
- These features were selected automatically by the AutoAI engine and are crucial for identifying suspicious patterns in data flow that signify intrusions.

ALGORITHM & DEPLOYMENT

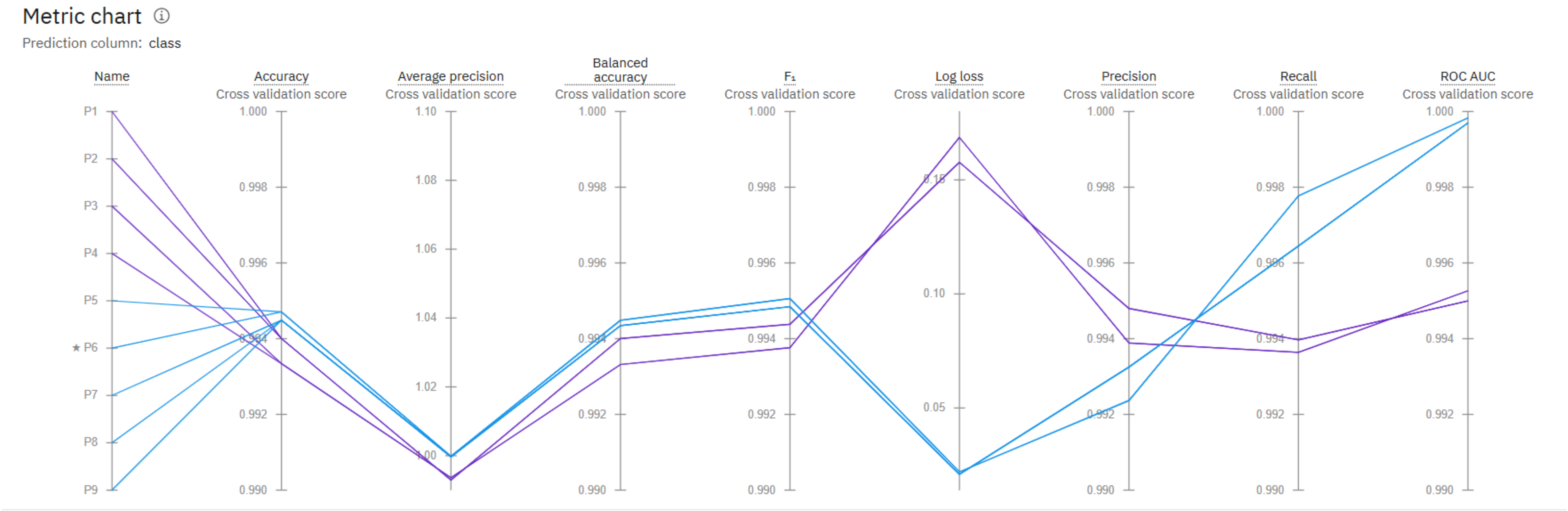
Training Process:

- The training process was initiated within the IBM Watsonx.ai AutoAI environment, which automatically handled data preprocessing, algorithm selection, and hyperparameter optimization. The dataset used contained labeled network traffic records with features such as protocol_type, flag, serror_rate, dst_host_same_src_port_rate, and more. The target column was binary: either normal or anomaly.
- Before training, duplicate entries were dropped and missing values were handled. The dataset was split using a 90/10 training/holdout strategy, ensuring the model was evaluated on unseen data. AutoAI applied internal k-fold cross-validation to prevent overfitting and to ensure stable model generalization. The training pipeline automatically tuned hyperparameters, such as the number of trees, tree depth, and feature splits, using grid search or random search strategies.
- Once training was complete, AutoAI generated a leaderboard of model candidates. The best-performing pipeline, which was a Random Forest model, was retained, and further evaluated using accuracy, precision, recall, and F1-score metrics. This model was then exported and deployed for real-time scoring.

Prediction Process:

- After deployment, the trained Random Forest model predicts whether incoming network traffic is normal or anomalous. It uses the same input feature set as during training. The model can be accessed as a web service in IBM Watsonx.ai, enabling real-time detection of intrusion attempts. This classification can help in issuing alerts or blocking malicious requests, thus strengthening network security in practical scenarios.

RESULT

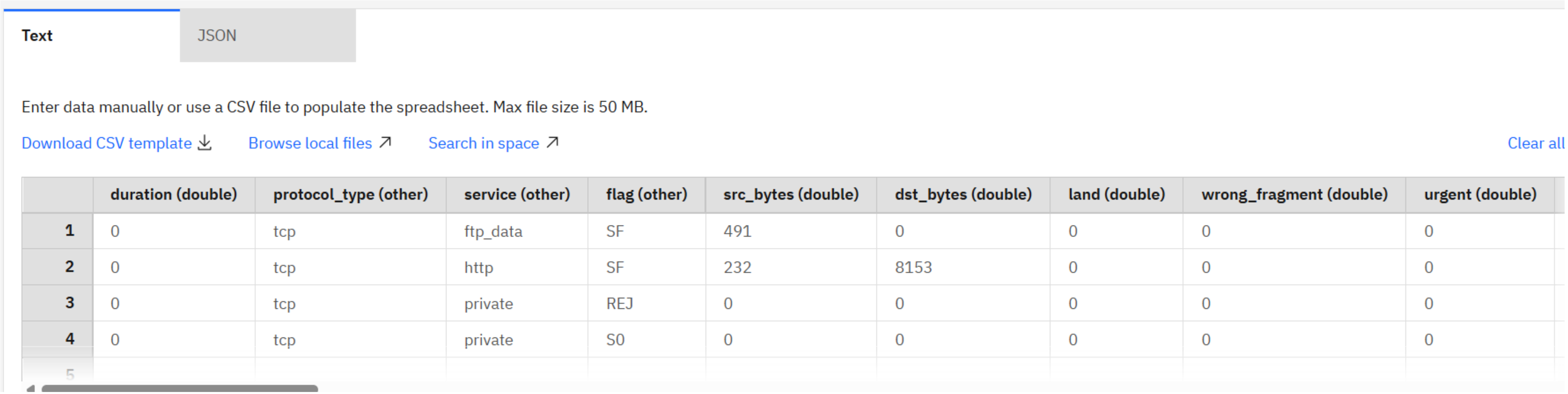


Metric chart

Prediction column: class

RESULT

Tested on fixed data

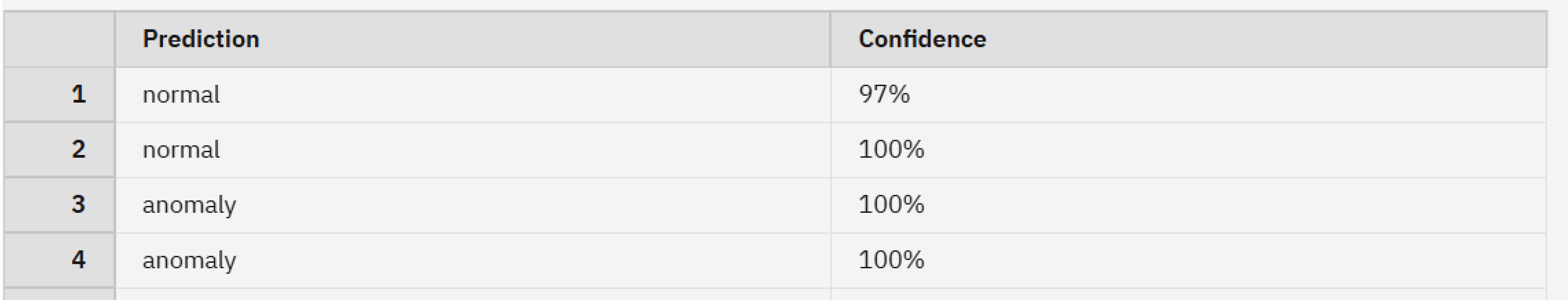


Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)
1	0	tcp	ftp_data	SF	491	0	0	0	0
2	0	tcp	http	SF	232	8153	0	0	0
3	0	tcp	private	REJ	0	0	0	0	0
4	0	tcp	private	S0	0	0	0	0	0
5									

RESULT

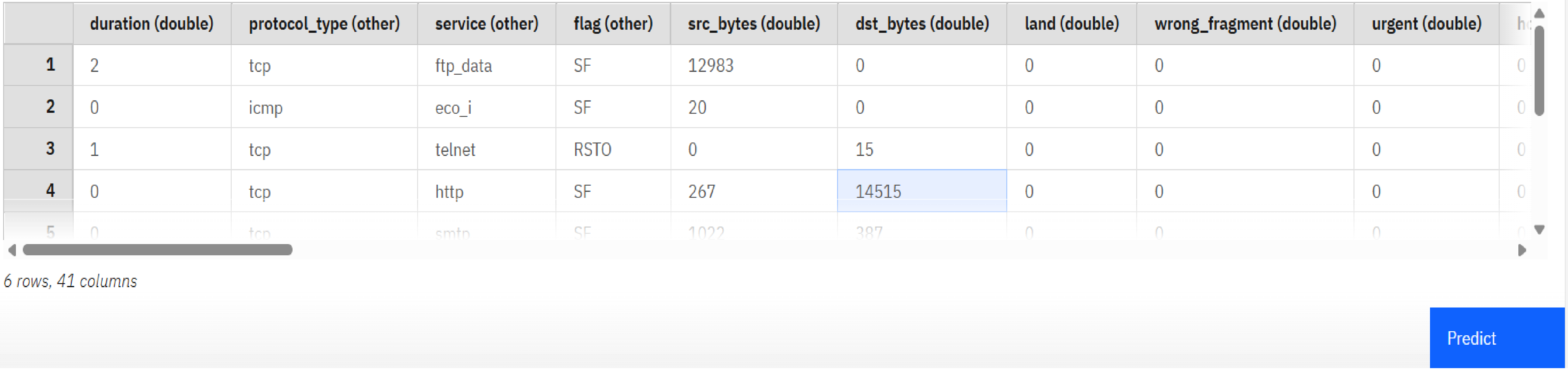
Results of the fixed data



	Prediction	Confidence
1	normal	97%
2	normal	100%
3	anomaly	100%
4	anomaly	100%

RESULT

Tested on different inputs

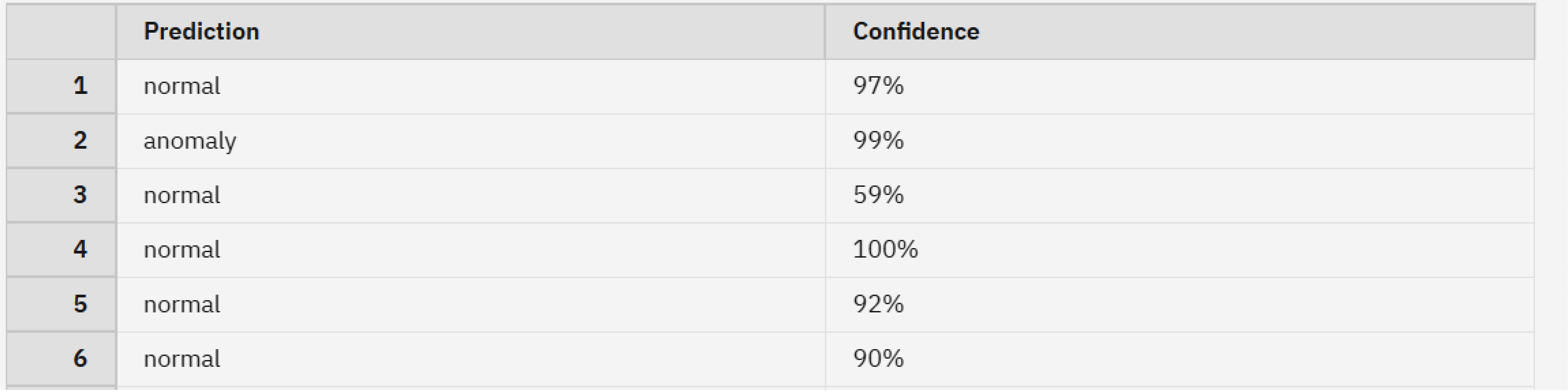


	duration (double)	protocol_type (other)	service (other)	flag (other)	src_bytes (double)	dst_bytes (double)	land (double)	wrong_fragment (double)	urgent (double)	h
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0
2	0	icmp	eco_i	SF	20	0	0	0	0	0
3	1	tcp	telnet	RSTO	0	15	0	0	0	0
4	0	tcp	http	SF	267	14515	0	0	0	0
5	0	tcp	smtp	SF	1022	387	0	0	0	0

6 rows, 41 columns

RESULT

Results of own input records



	Prediction	Confidence
1	normal	97%
2	anomaly	99%
3	normal	59%
4	normal	100%
5	normal	92%
6	normal	90%

CONCLUSION

- The proposed machine learning-based Network Intrusion Detection System successfully classified network traffic as either normal or anomalous with an impressive accuracy of 99%. Leveraging the capabilities of IBM Watsonx AutoAI, the system effectively handled data preprocessing, model selection, and hyperparameter optimization with minimal manual intervention. The Random Forest algorithm emerged as the best-performing model, demonstrating high reliability in identifying malicious patterns.

- Throughout the implementation, one of the key challenges was ensuring the dataset was properly cleaned and structured for training, especially given the diversity of network features. However, Watsonx's automated pipeline significantly simplified this process. The deployment phase was seamless, with the model successfully integrated into a web service capable of making real-time predictions.

- This project underscores the critical role of accurate intrusion detection in safeguarding digital infrastructure. With continued updates and expanded datasets, the system has the potential to adapt to evolving threats, making it a scalable and dependable security solution for communication networks.

FUTURE SCOPE

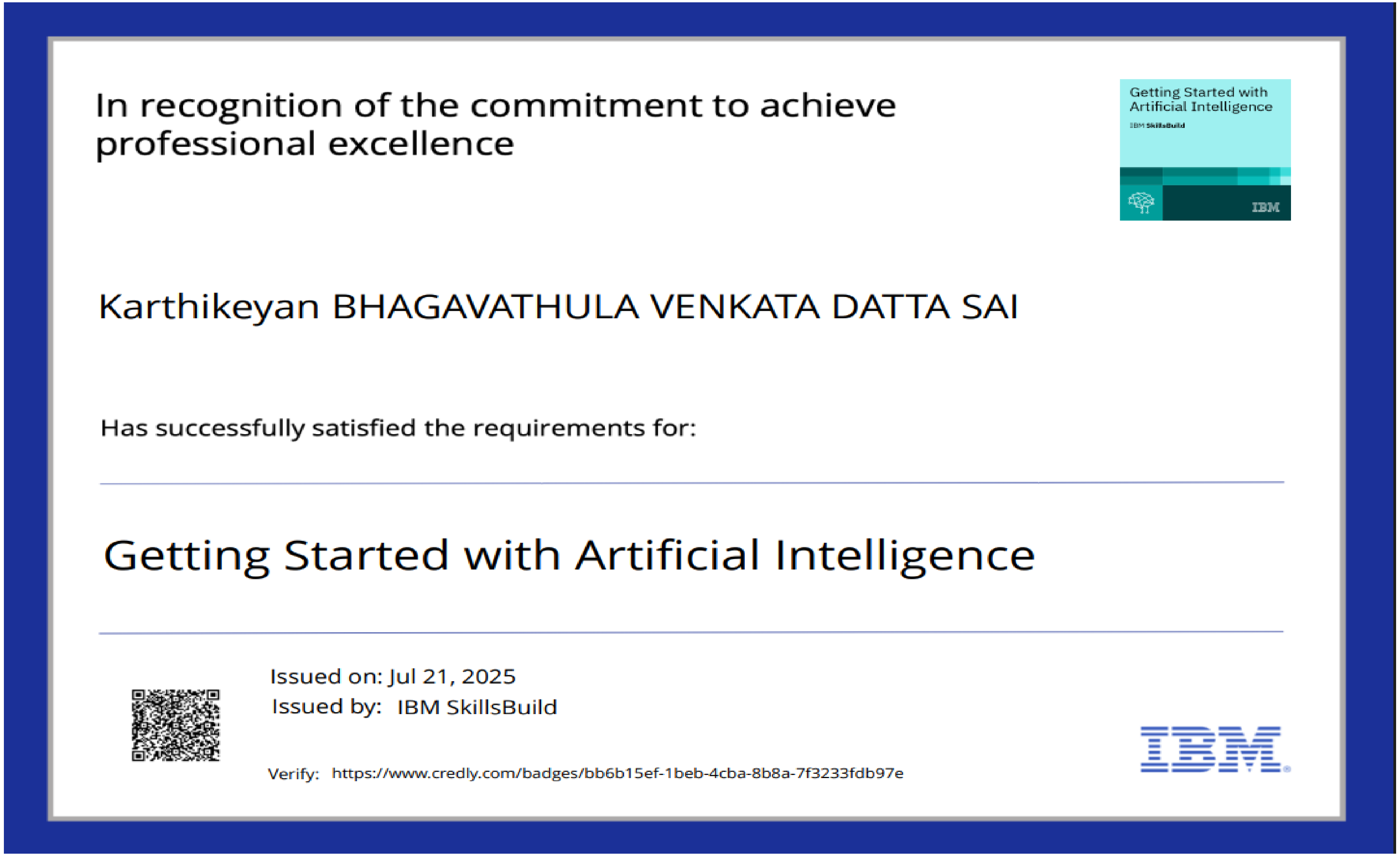
The current intrusion detection system demonstrates strong classification performance, but there are several areas for enhancement and expansion:

- **Integration of Real-Time Data Sources**: Future versions of the system can incorporate live traffic streams, enabling proactive detection and response to intrusions in real-time environments.
- **Algorithm Optimization**: Although Random Forest performed well, exploring advanced techniques like deep learning (e.g., LSTM for sequential pattern analysis) or hybrid models may further boost detection rates, especially for rare or evolving attack types.
- **Scalability Across Networks**: The model can be adapted to monitor and protect larger and more complex networks, such as enterprise systems or multiple interconnected city infrastructures.
- **Edge Computing Integration**: Deploying the intrusion detection logic on edge devices can help process traffic data locally, reducing latency and improving real-time threat mitigation.
- **Adaptive Learning Capabilities**: Incorporating online learning methods will allow the model to continuously learn from new threats, improving its ability to detect zero-day or previously unseen attacks.
- **Visualization Dashboards**: Building comprehensive dashboards can assist security analysts in monitoring system activity and intrusion alerts more effectively.

REFERENCES

1. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. In IEEE Symposium on Computational Intelligence for Security and Defense Applications. https://doi.org/10.1109/CISDA.2009.5356528
2. Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 446–452. https://www.ijarcce.com/
3. IBM Watsonx.ai Documentation. AutoAI Experiments and Model Deployment Guide. https://www.ibm.com/docs/en/watsonx
4. Scikit-learn Developers. (2023). Machine Learning in Python. https://scikit-learn.org/stable/
5. Kaggle. (2021). Network Intrusion Detection Dataset. https://www.kaggle.com/datasets/sampadab17/networkintrusion-detection
6. Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

IBM CERTIFICATIONS



Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

In recognition of the commitment to achieve professional excellence

Karthikeyan BHAGAVATHULA VENKATA DATTA SAI

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

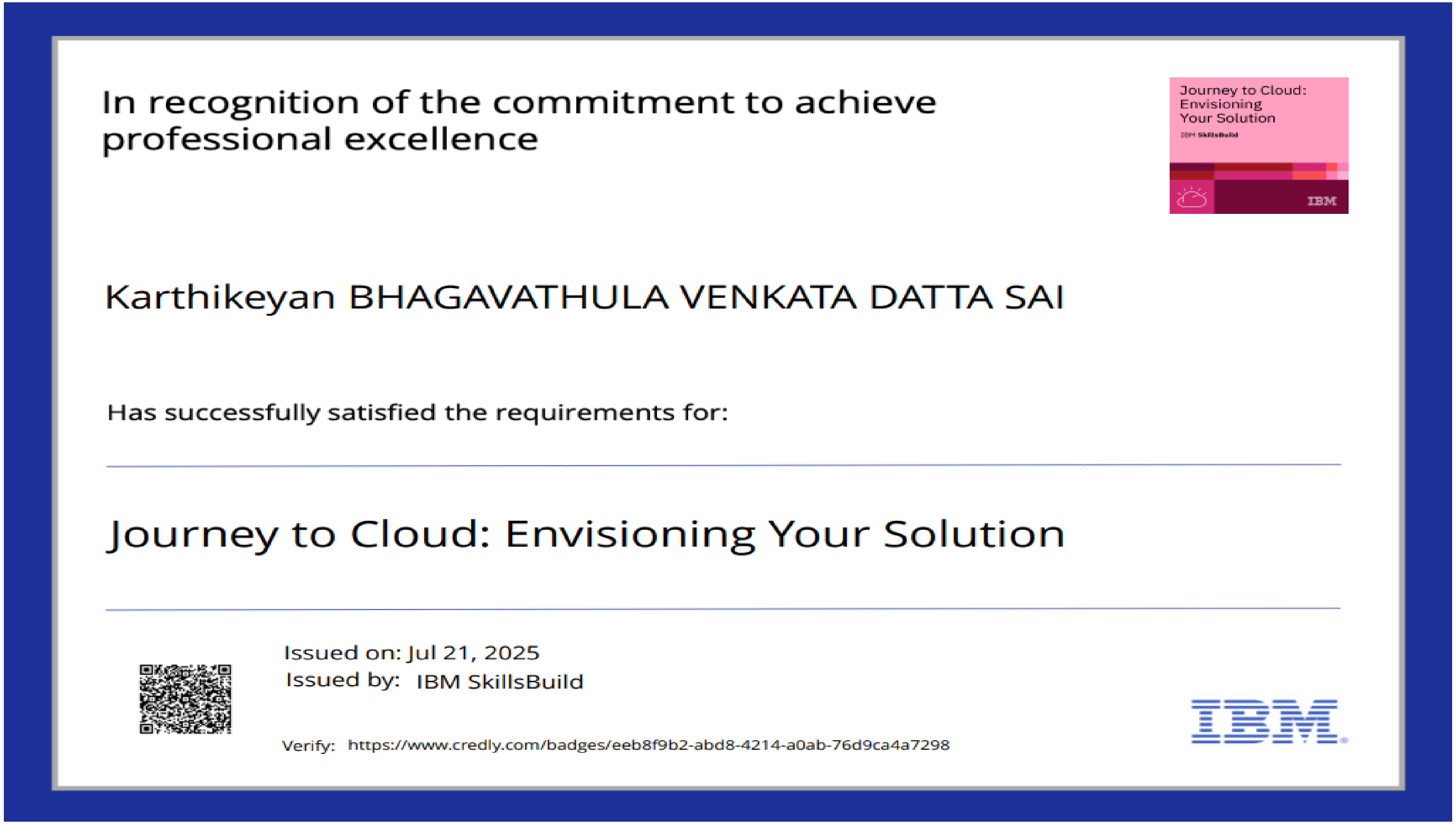
Issued on: Jul 21, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/bb6b15ef-1beb-4cba-8b8a-7f3233fdb97e

IBM

IBM CERTIFICATIONS



Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild
IBM

In recognition of the commitment to achieve professional excellence

Karthikeyan BHAGAVATHULA VENKATA DATTA SAI

Has successfully satisfied the requirements for:

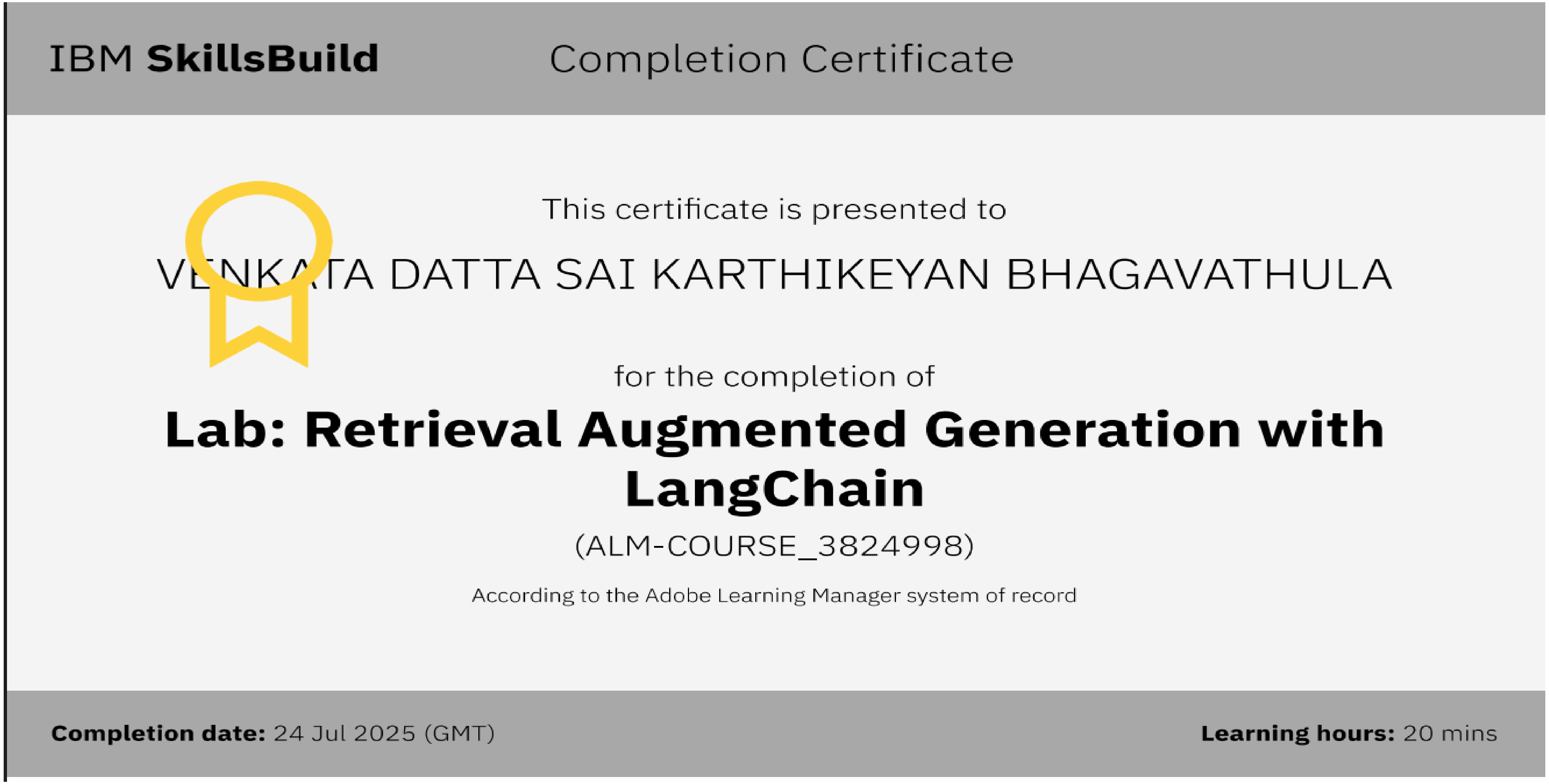
Journey to Cloud: Envisioning Your Solution

Issued on: Jul 21, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/eeb8f9b2-abd8-4214-a0ab-76d9ca4a7298

IBM

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

VENKATA DATTA SAI KARTHIKEYAN BHAGAVATHULA

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