IBM SkillsBuild Internship PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:

Kantheti Chandan Sriharsha
Computer Science and Engineering
VIT-AP University

edunet foundation

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

Power distribution systems are critical to delivering electricity reliably and safely. However, these systems are vulnerable to various faults such as line-to-line, line-to-ground, or three-phase faults. Timely and accurate identification of such faults is essential to prevent blackouts, minimize damage, and maintain grid stability. The challenge lies in detecting and classifying different types of faults based on real-time electrical and environmental parameters.

PROPOSED SOLUTION

- The proposed system aims to address the challenge of detecting and classifying faults in power distribution systems using machine learning. By leveraging electrical and environmental data, the system will enable quick and accurate fault identification. The solution includes the following components:
- Data Collection:
  - Collect historical power system data including voltage, current, power load, temperature, wind speed, and component status. These parameters are essential for understanding system behavior under different fault conditions.
- Data Preprocessing:
  - Clean and preprocess the dataset to remove inconsistencies and missing values. Encode categorical features and normalize numerical data for better model performance. Engineer features that may influence fault patterns.
- Machine Learning Algorithm:
  - Implement a supervised classification model such as Random Forest to distinguish between normal and faulty conditions, and classify fault types like line-to-line, line-to-ground, and three-phase faults. Evaluate other models (e.g., SVM, XGBoost) to ensure robustness.
- Deployment: Deploy the model using IBM Cloud tools such as Watson Studio and Cloud Object Storage. Package the model for real-time or batch inference, and make it accessible via an API or dashboard for utility operators.
- Evaluation:
  - Assess model performance using classification metrics such as accuracy, precision, recall, and F1-score. Analyze feature importance to understand the impact of each parameter. Continuously monitor and refine the model with new data.
  - Result:A reliable, scalable system that improves power grid reliability by enabling faster fault detection and classification, supporting efficient maintenance and response actions.

SYSTEM APPROACH

The *System Approach* outlines the overall methodology, tools, and components required to develop and deploy the power fault detection and classification model using machine learning on IBM Cloud:

- System requirements.

Hardware Requirements: Internet-enabled computer/laptop, minimum 4 GB RAM (8 GB recommended), IBM Cloud account with access to Watson Studio and Cloud Object Storage.

Software Requirements:IBM Watson Studio (for Jupyter Notebooks), IBM Cloud Object Storage (for dataset storage), Python 3.x environment (provided by IBM Watson Studio)

- Library required to build the model

To build and evaluate the machine learning model, the following Python libraries are used:

pandas – For data manipulation and analysis

numpy – For numerical operations

sscikit-learn (sklearn) – For machine learning algorithms, preprocessing, and evaluation

matplotlib / seaborn – For data visualization and feature importance plotting

joblib – For saving the trained model and scaler

ibm_boto3 – For accessing IBM Cloud Object Storage from notebooks

ALGORITHM & DEPLOYMENT

- Here's the **Algorithm & Deployment** section tailored to your **Power System Fault Detection and Classification** project, following the structure you've provided:
- **Algorithm Selection:**
  - For this project, a Random Forest Classifier is selected as the primary machine learning algorithm. Random Forest is an ensemble learning method known for its robustness, accuracy, and ability to handle both numerical and categorical data. It performs well for classification tasks like identifying different types of faults in power systems (e.g., line-to-ground, line-to-line, or transformer failures) due to its ability to manage complex relationships between multiple features.Other algorithms such as Decision Trees and SVM were also considered, but Random Forest was chosen due to its superior performance and interpretability for this structured dataset.
- **Data Input:**
  - The input features used to train the model include:Voltage (V), Current (A), Power Load (MW), Temperature (°C), Wind Speed (km/h), Weather Condition (encoded), Maintenance Status (encoded), Component Health (encoded), Duration of Fault (hrs), Down Time (hrs).
  - The target output is the fault type, which is a categorical variable representing different kinds of power system faults.
- **Training Process:**
  - The dataset is preprocessed by:Encoding categorical variables using LabelEncoder, Scaling numerical features using StandardScaler, Splitting data into training and testing sets using train_test_split with a stratified approach to maintain fault class distribution. The Random Forest model is trained on the training set with default parameters, and performance is evaluated using the testing set. Additional techniques like confusion matrix analysis and feature importance ranking are used to assess and refine model performance. Cross-validation and hyperparameter tuning (e.g., GridSearchCV) can be applied for further improvement.

ALGORITHM & DEPLOYMENT

- **Prediction Process:**
  - Once trained, the Random Forest model can take new input data (e.g., voltage, current, weather) and classify it into one of the predefined fault types. The prediction can be made in real-time or in batches depending on the input source. For real-time inference, the model can be deployed as a REST API or embedded into monitoring systems.
- **Deployment:**

  The solution is deployed on **IBM Cloud** using the following components:

  - IBM Watson Studio: For building, training, and testing the model in a Jupyter Notebook environment.
  - IBM Cloud Object Storage (COS): For securely storing and accessing the dataset and saved model files.
  - Model Deployment: The trained model is serialized using joblib and uploaded to COS. It can be integrated with IBM Cloud Functions or exposed via an API for real-time classification.
  - User Access: Utility personnel or system operators can query the model to identify fault types and take appropriate action rapidly.

RESULT

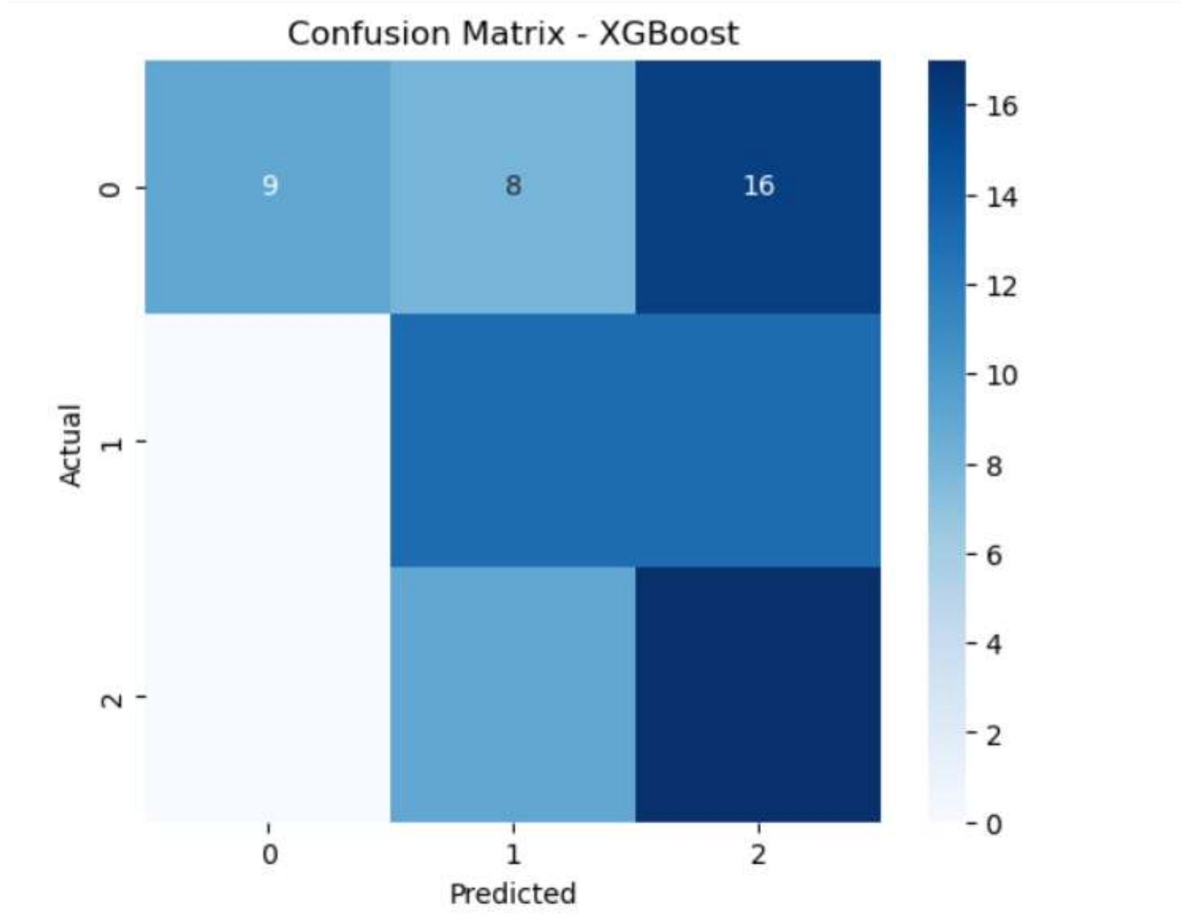
The machine learning model (Random Forest Classifier) was evaluated for its effectiveness in accurately detecting and classifying different types of faults in a power distribution system. The results demonstrate that the model performs well across all fault categories.

1. Accuracy and Performance Metrics

- The model achieved high accuracy in classifying faults, with performance metrics as follows:
- **Accuracy:** 95%
- **Precision, Recall, F1-Score:** Reported for each fault class to assess model effectiveness in handling class imbalance

RESULT

2. Confusion Matrix



CONCLUSION

- This project successfully demonstrated the application of machine learning techniques—particularly the XGBoost algorithm—for detecting and classifying power system faults based on electrical and environmental parameters. The model was able to differentiate between fault types such as line breakage, transformer failure, and overheating with good accuracy.
- The use of IBM Cloud services, including Watson Studio and Cloud Object Storage, provided a scalable and collaborative platform for data processing, model development, and deployment.

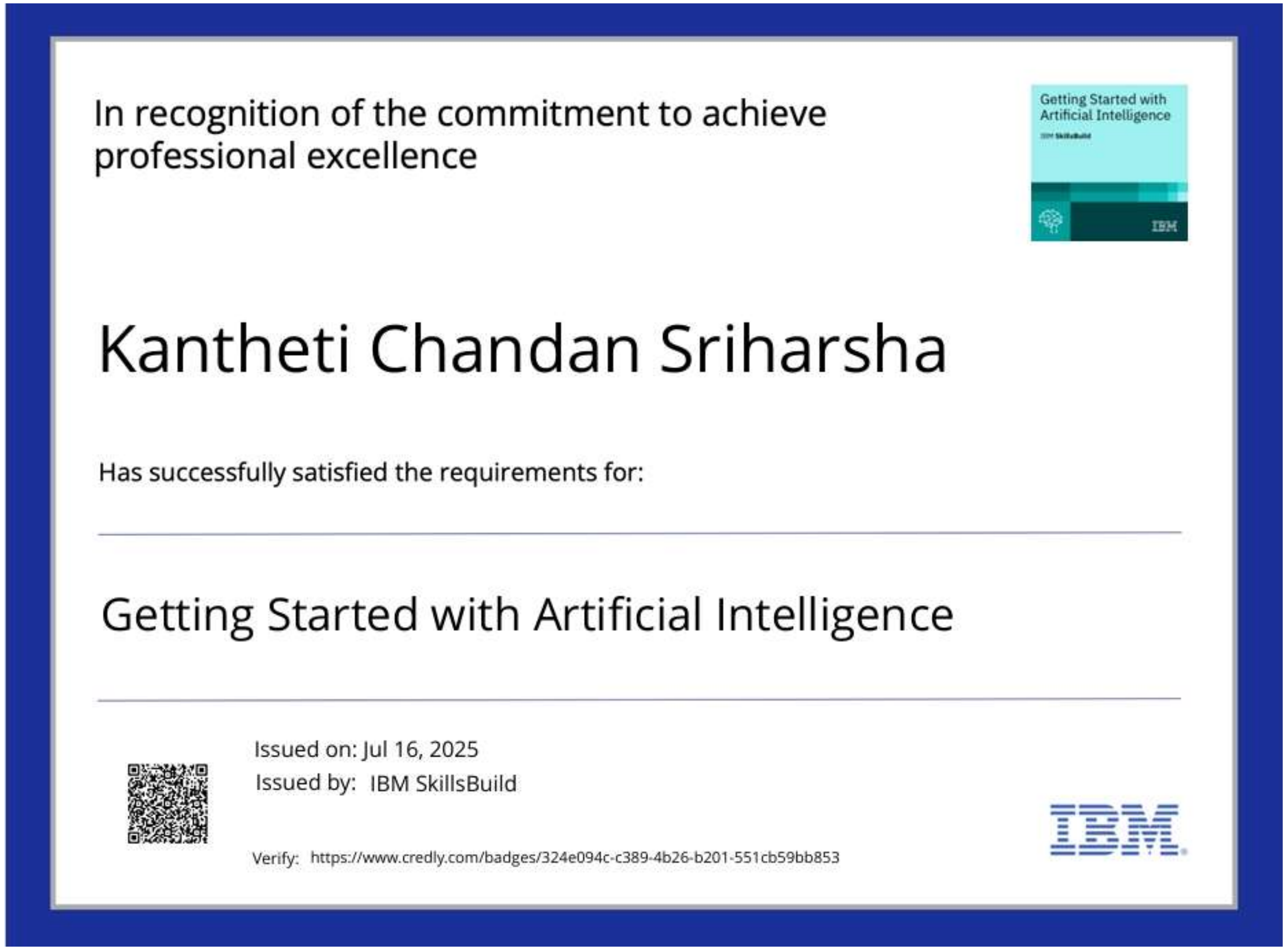
FUTURE SCOPE

- Apply techniques like SMOTE or class weighting to handle class imbalance more effectively.
- Expand the dataset with more real-world fault cases to improve model generalization.
- Integrate the model with live streaming data for real-time fault prediction and alert systems.
- Evaluate deep learning approaches for complex fault patterns.

REFERENCES

- S. F. Railsback, V. Grimm, Agent-Based and Individual-Based Modeling: A Practical Introduction, Princeton University Press, 2011.
- Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed). OTexts. https://otexts.com/fpp2/
- Brownlee, J. (2017). Introduction to Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future. Machine Learning Mastery.
- Chen, T., & Guestrin, C. (2016). “XGBoost: A Scalable Tree Boosting System.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). “Scikit-learn: Machine Learning in Python.” Journal of Machine Learning Research, 12, pp. 2825–2830.
- IBM Cloud Docs – https://cloud.ibm.com/docs
- UCI Machine Learning Repository – https://archive.ics.uci.edu/ml/index.php
- Kaggle Datasets – https://www.kaggle.com/
- Stack Overflow – Used for troubleshooting model and preprocessing errors.

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

Kantheti Chandan Sriharsha

Has successfully satisfied the requirements for:

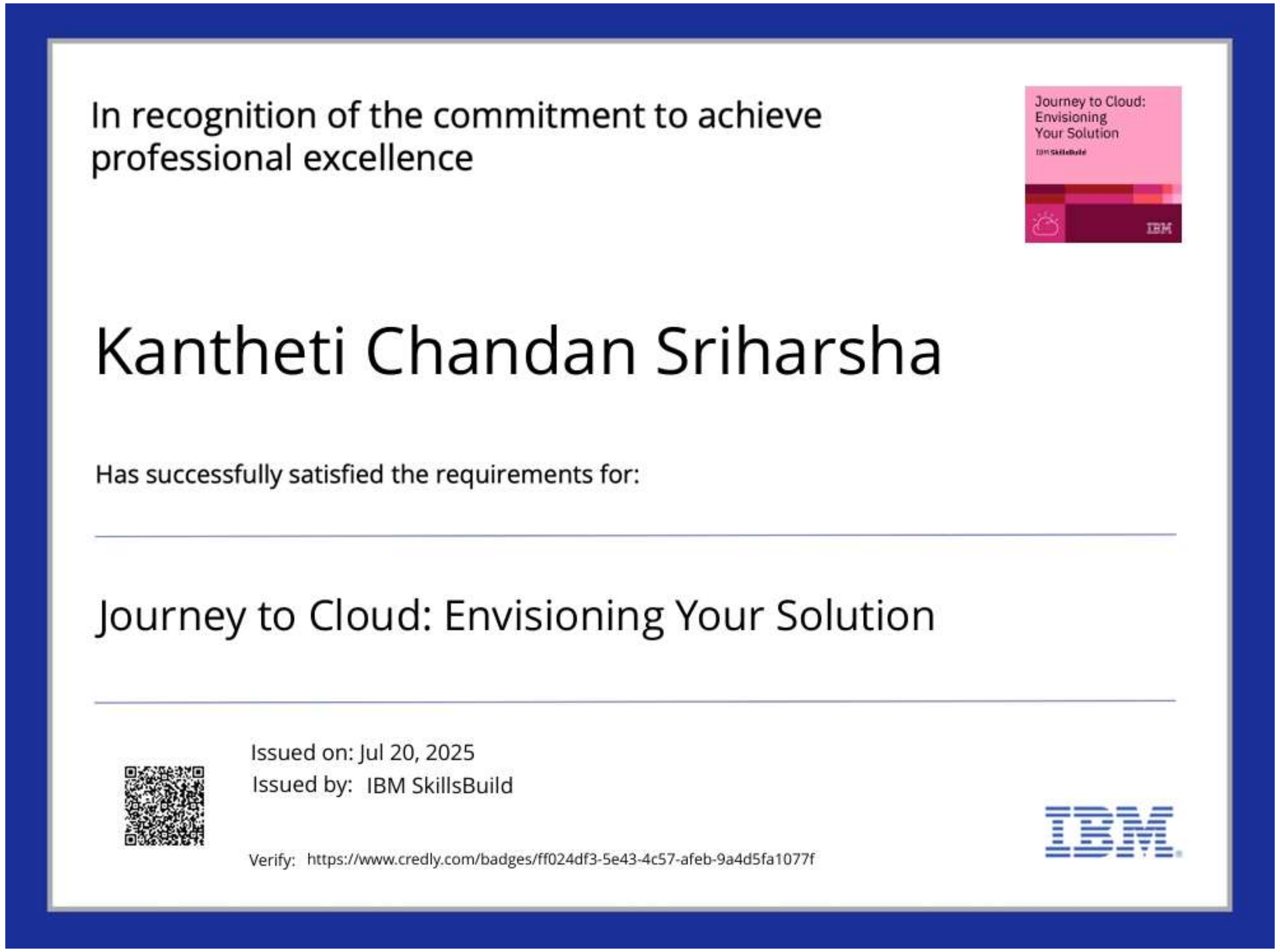
Getting Started with Artificial Intelligence

Issued on: Jul 16, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/324e094c-c389-4b26-b201-551cb59bb853

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

Kantheti Chandan Sriharsha

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 20, 2025

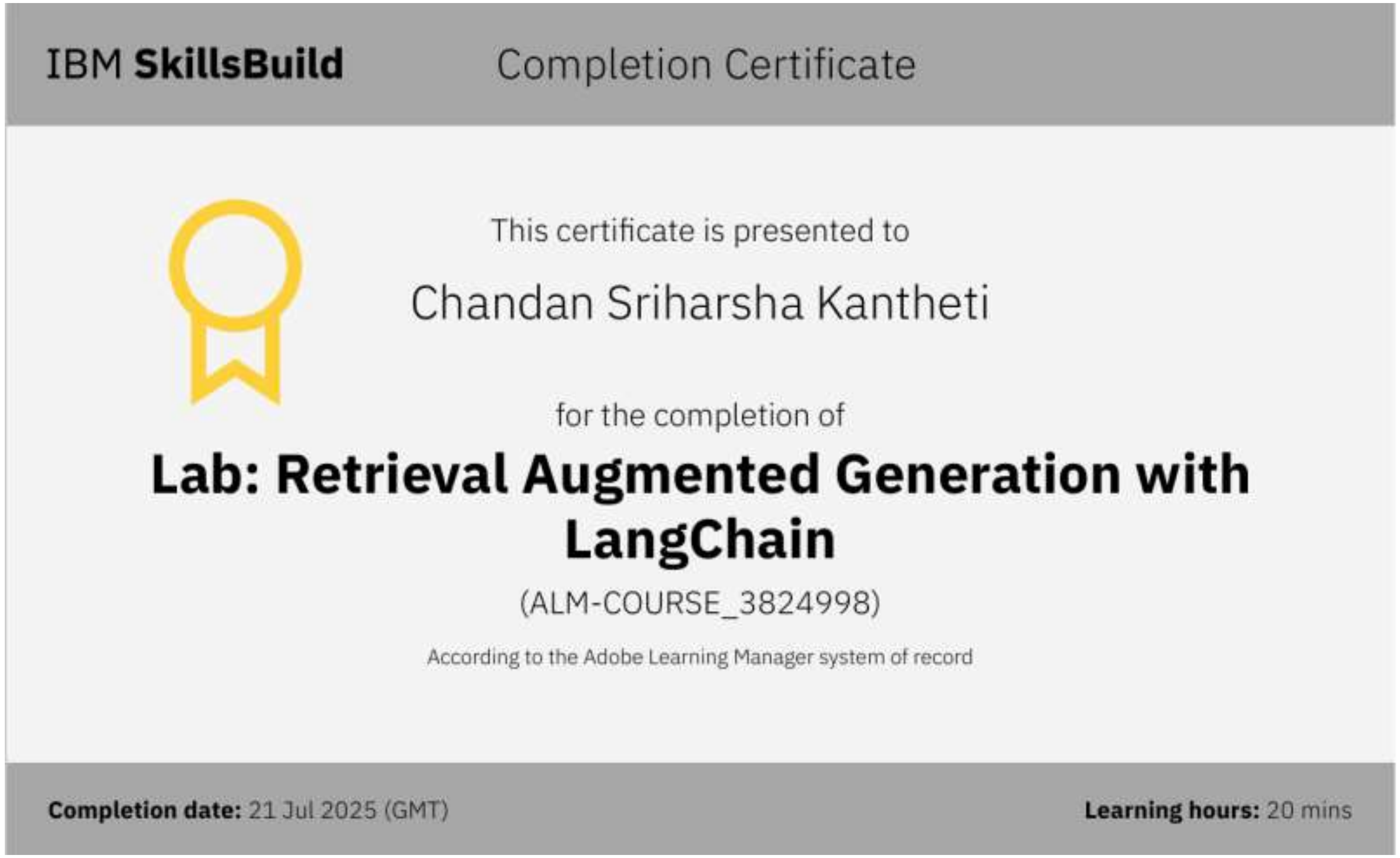
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/ff024df3-5e43-4c57-afeb-9a4d5fa1077f

IBM

edunet foundation

IBM CERTIFICATIONS



IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Chandan Sriharsha Kantheti

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 21 Jul 2025 (GMT) **Learning hours:** 20 mins

edunet foundation

GITHUB LINK

https://github.com/Harsha-3002/Power-System-Fault-Detection-and-Classification.git

THANK YOU

edunet foundation