# CAPSTONE PROJECT

## POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
1. Vivek Chauhan-Graphic Era Deemed To Be University-Computer Science Engineering(CSE)

edunet foundation

# OUTLINE

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

# PROBLEM STATEMENT

- Modern power distribution systems are critical for uninterrupted electricity delivery but are prone to various faults such as line-to-ground, line-to-line, and three-phase faults. These faults, if undetected, can lead to equipment damage, service disruption, and system instability. There is a growing need for an intelligent system that can analyze real-time electrical measurements (like voltage and current phasors) and accurately classify fault types to ensure rapid response and enhance grid reliability.

# PROPOSED SOLUTION

- The objective is to develop an intelligent fault detection and classification system for power distribution networks using machine learning. The approach leverages voltage and current phasor data to distinguish between normal operation and multiple fault conditions.

- Key Components:

- Data Acquisition: Collect electrical measurement data under normal and fault conditions.

- Feature Extraction: Process phasor data to derive meaningful input features.

- Model Training: Train a classification model(e.g. Decision Tree, Random Forest or SVM).

- Fault Classification: Automatically identify the type of fault (e.g., L-G, L-L, L-L-L).

- Evaluation: Assess performance using classification accuracy, F-1 score, and precision-recall metrics.

# SYSTEM APPROACH

System Requirements:

- IBM Watson Studio / IBM Cloud Pak for Data
- IBM Cloud Object Storage
- IBM AutoAI (optional for model training assistance)
- Python environment with scikit-learn, pandas, numpy, etc.

Development Workflow:

- **Data Ingestion:**
  Upload phasor data (voltage/current measurements) to **IBM Cloud Object Storage**.
- **Data Preprocessing & Feature Engineering:**
  Use **IBM Watson Studio** notebooks to clean, transform, and prepare the data.
- **Model Development:**
  Train a **Random Forest Classifier** using Jupyter notebooks in Watson Studio or use **AutoAI** for automated model building.
- **Model Evaluation:**
  Evaluate accuracy, precision, recall, and confusion matrix within the notebook environment.
- **Model Deployment:**
  Deploy the trained model as a REST API using **Watson Machine Learning (WML)** for real-time fault classification.
- **Integration & Monitoring:**
  Integrate the deployed API with a fault monitoring dashboard or SCADA interface. Use **IBM Cloud Monitoring** for usage and performance insights.

# ALGORITHM & DEPLOYMENT

**Algorithm Selection:**

The **Random Forest Classifier** is chosen for its robustness, high accuracy, and ability to handle multiclass classification problems effectively. In the context of power system fault detection, Random Forest provides reliable fault classification (e.g., line-to-ground, line-to-line, three-phase) by aggregating decisions from multiple decision trees, reducing the risk of overfitting and improving generalization on unseen fault patterns.

**Data Input:**

The model uses labeled electrical measurement data, including:

- Voltage phasors (magnitude and phase angle)
- Current phasors (magnitude and phase angle)
- System status labels (e.g., Normal, L-G, L-L, L-L-L)
  These features are extracted under different system operating and fault conditions.

**Training Process:**

The data is preprocessed and split into training and testing sets. The Random Forest model is trained using scikit-learn in IBM Watson Studio. Key steps include:

- Feature normalization
- Cross-validation for generalization
- Hyperparameter tuning (number of trees, depth, etc.) using grid search or AutoAI
  The model is evaluated using metrics such as accuracy, precision, recall, and F1-score.

**Prediction Process:**

Once trained, the model is deployed on **IBM Watson Machine Learning** as a REST API. During runtime:

- Real-time phasor data is sent to the model endpoint
- The API returns a predicted fault type instantly

# RESULT

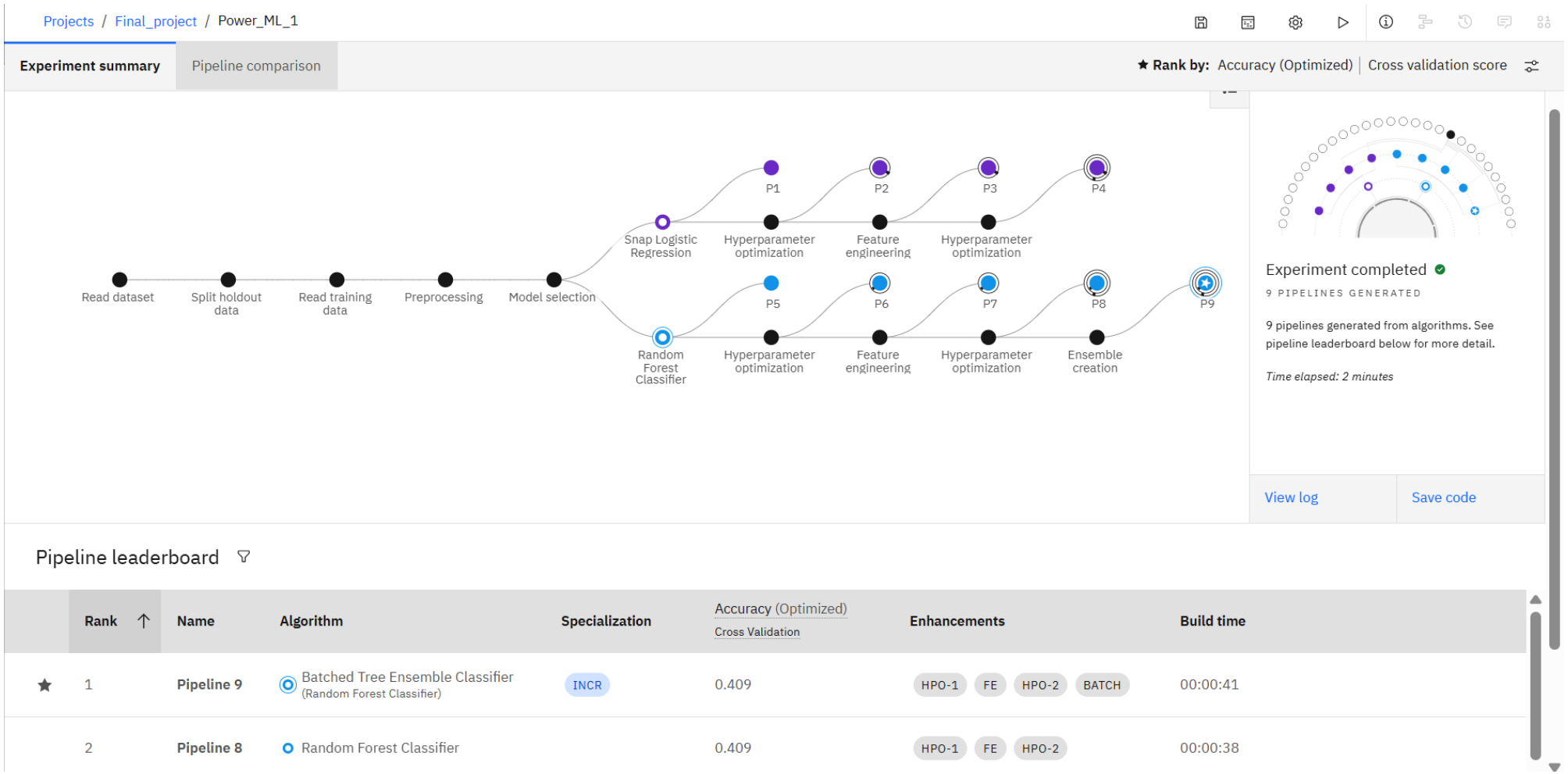


Fig 1: Progress Map

edunet foundation

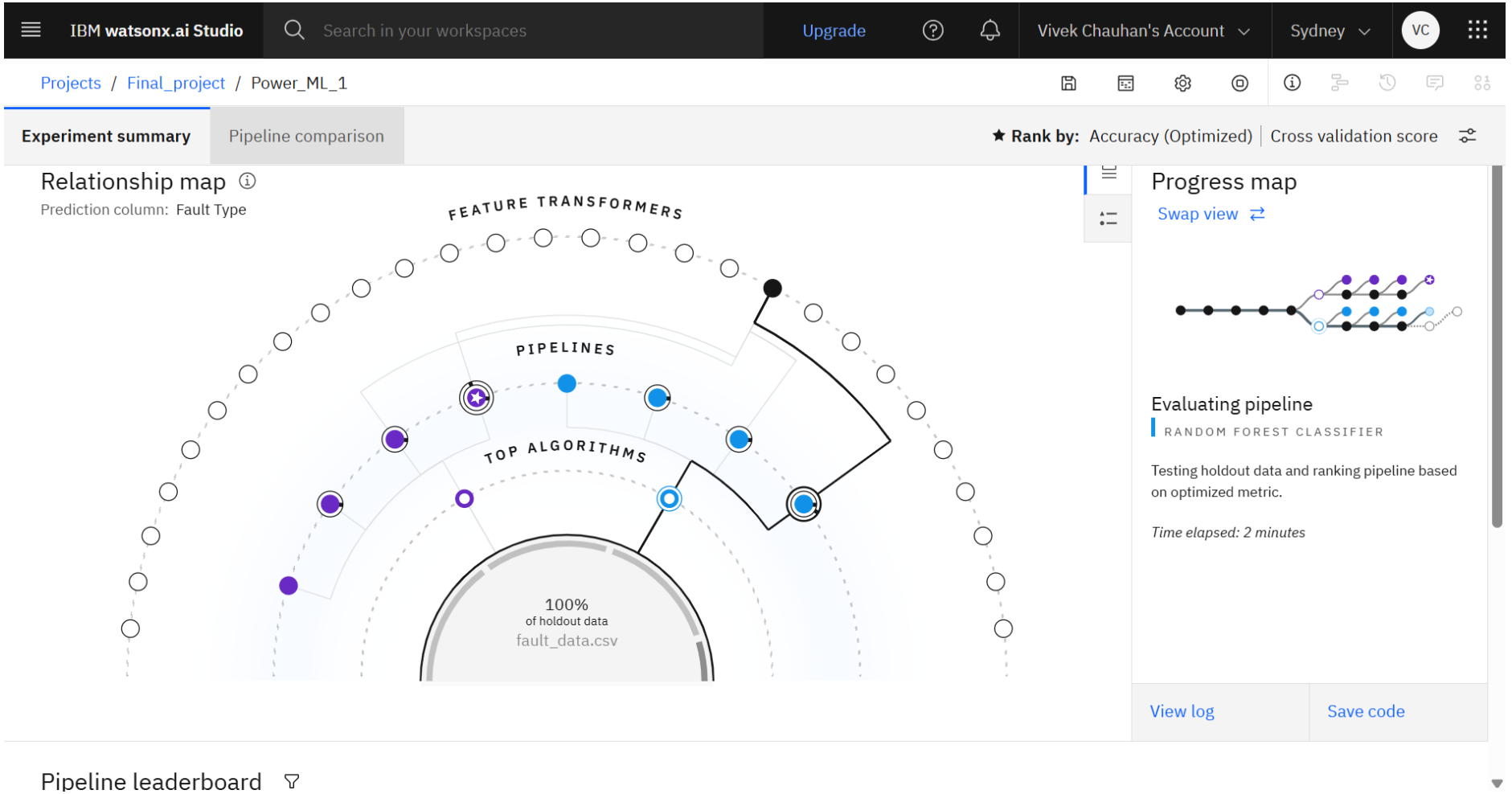


Fig 2: Relationship Map

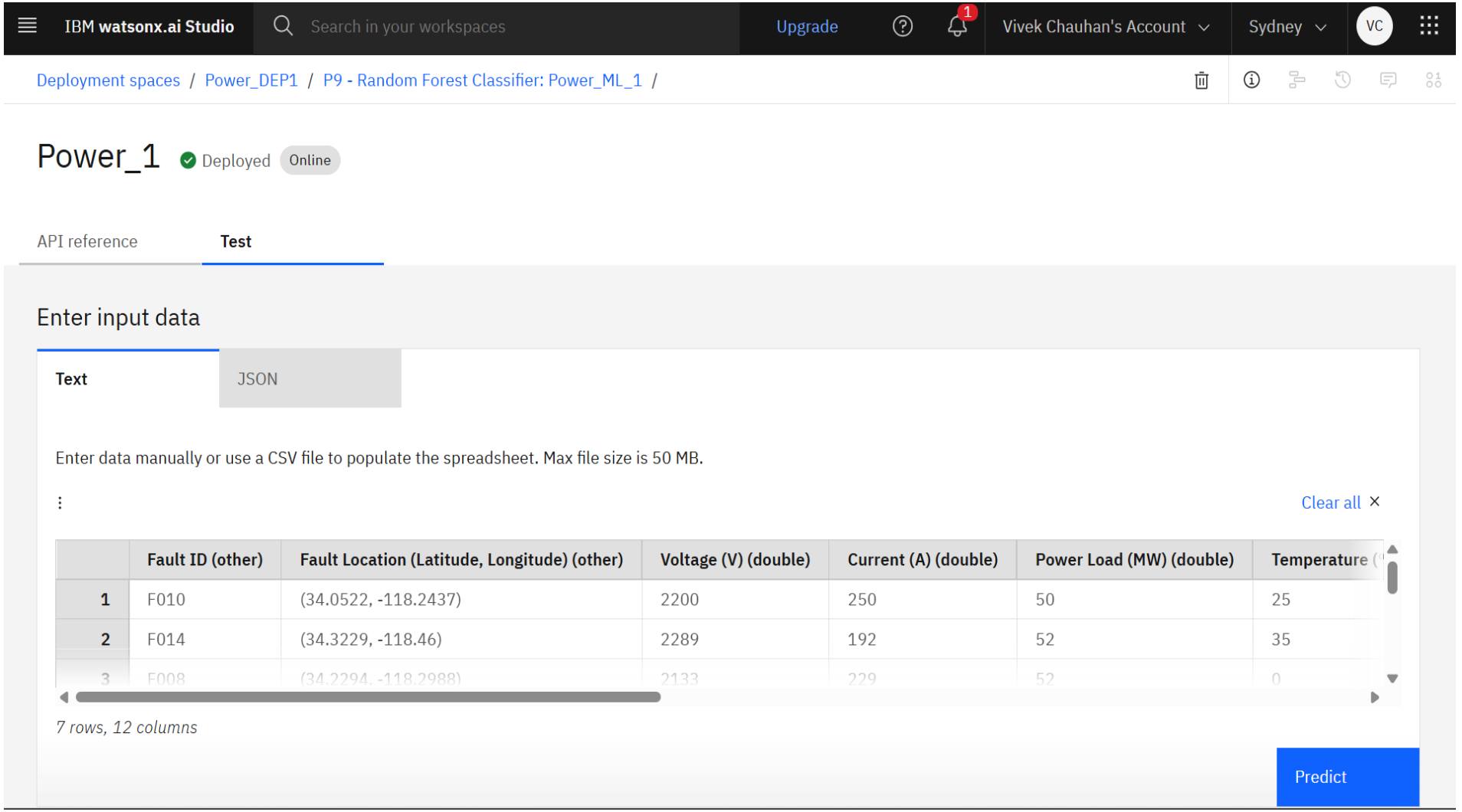


Fig 2: Inputting data that needs to be predicted

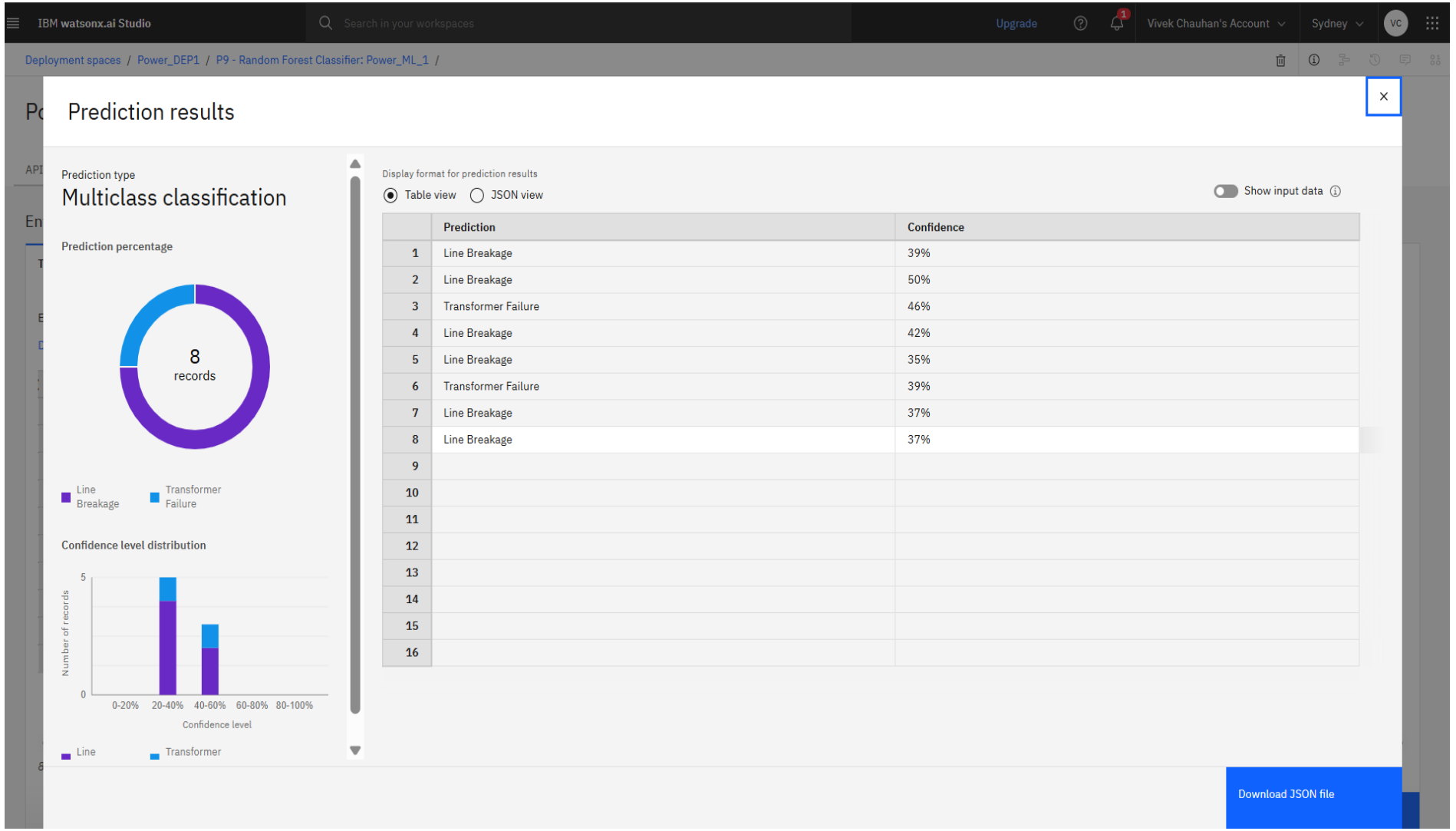


Fig 3: Prediction result of the Random Forest model

# CONCLUSION

- The project successfully demonstrates the use of machine learning—specifically the Random Forest Classifier—for accurately detecting and classifying different types of faults in a power distribution system. By leveraging electrical phasor data (voltage and current), the model can differentiate between normal operations and various fault types such as line-to-ground, line-to-line, and three-phase faults.
- Integration with **IBM Cloud services** (Watson Studio, WML, and Cloud Object Storage) facilitated seamless model development, training, and deployment. The system ensures real-time fault identification, enabling faster response times, reduced downtime, and improved grid reliability.
- This solution highlights the potential of AI-powered automation in modernizing the fault management process in power systems.

# FUTURE SCOPE

- **Edge Deployment:** Implementing the trained model on edge devices for ultra-fast local fault detection without relying on cloud latency.
- **Real-time Streaming Integration:** Using IBM Cloud services like **IBM Streams** to process live sensor data continuously.
- **Deep Learning Models:** Exploring neural networks (e.g., LSTM, CNN) for more complex fault pattern recognition and predictive maintenance.
- **Expanded Dataset:** Incorporating more types of faults, multiple voltage levels, and seasonal/environmental conditions for improved accuracy.
- **SCADA Integration:** Embedding the model directly into SCADA systems for automated control and protection.
- **Cybersecurity Layer:** Integrating fault detection with security models to identify abnormal behavior due to cyber threats.

# REFERENCES


- Scikit-learn: Machine Learning in Python
  https://scikit-learn.org/stable/modules/ensemble.html#forest
- IBM Watson Studio Overview
  https://www.ibm.com/cloud/watson-studio
- IBM Cloud Machine Learning Deployment Guide
  https://www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=projects-deploying-models
- Towards Data Science: *Understanding Random Forests*
  https://towardsdatascience.com/understanding-random-forest-58381e0602d2
- Medium: *Fault Detection in Power Systems Using ML*
  https://medium.com/@powergridai/fault-detection-ml

# IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence
IBM SkillsBuild
IBM

Vivek Chauhan

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 17, 2025
Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/0091e8a3-7493-42eb-9d39-07706b268d2b

IBM

# IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution
IBM SkillsBuild

Vivek Chauhan

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/b41af5dc-edba-49be-9782-f86d762341b3

IBM

# IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Vivek Chauhan

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

edunet foundation