

---

# **MACHINE LEARNING PROJECT**

## **POWER SYSTEM FAULT DETECTION AND CLASSIFICATION**

**Presented By:**  
**RAHUL KUMAR SINHA-BIT SINDRI, DHANBAD-  
INFORMATION TECHNOLOGY**

# OUTLINE

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

---

# PROBLEM STATEMENT

Design a machine learning model to detect and classify different types of faults in a power distribution system. Using electrical measurement data (e.g., voltage and current phasors), the model should be able to distinguish between normal operating conditions and various fault conditions (such as line-to-ground, line-to-line, or three-phase faults). The objective is to enable rapid and accurate fault identification, which is crucial for maintaining power grid stability and reliability.

# PROPOSED SOLUTION

## 1 Data Collection

**Gather Data:** Collect high-resolution voltage and current phasor measurements (e.g., from Phasor Measurement Units - PMUs) under different conditions:

Normal operating conditions

Fault conditions: line-to-ground, line-to-line, three-phase faults

**Data Labeling:** Label each data segment with the corresponding state for supervised learning.

## 2. Data Preprocessing

**Filtering & Noise Reduction:** Use filters (e.g., low-pass filter) to eliminate noise.

**Windowing:** Segment data into fixed-length windows (e.g., 100-200 ms) for analysis.

**Normalization:** Scale features for consistent ranges.

## 3. Model Selection

Use robust classification algorithms suitable for Multiclass-Classification:

**Random Forest Classifier:** it provides robustness and also has high accuracy.

## 4. Validation & Testing :-

Simulate various fault scenarios to validate model robustness.

Conduct field tests under controlled conditions to ensure reliability.

# SYSTEM APPROACH

## HARDWARE , SOFTWARE AND STORAGE REQUIREMENT :-

### 1.watsonx.ai Studio:

- It offers a collaborative environment with tools for data preparation, model development, testing, and deployment, all within a cloud-based interface.

### 2. watsonx.ai Runtime:

- It provides 8 virtual CPU and 32 GB of RAM.

### 3. Cloud object Storage:

- In the free lite version, we are provided with 25 GB per month storage

# ALGORITHM & DEPLOYMENT

- In this Algorithm section, we would see the machine learning algorithm chosen for predicting Power fault Detection.
- **Algorithm Selection:**
  - Here Random Forest Classifier is chosen because Random Forest tends to deliver excellent predictive performance by combining multiple decision trees to reduce overfitting and increase generalization which is required here for predicting the power fault type.
- **Data Input:**
  - The inputs taken here are fault type, fault location , voltage, current , power load, Temperature, wind speed, weather , component nature, Duration of fault and Down Time these are the parameters taken here as the inputs.

## Training Process:

- **Input Data:** The Random Forest Classifier uses the extracted features along with labels indicating the fault type.
- **Model Training:** Train the Random Forest classifier on this labeled dataset so it learns to distinguish patterns associated with different faults.
- **Prediction Process:-**

**Prediction:** The Random Forest analyzes the incoming features and predicts whether a fault exists and its specific type.

**Decision-Making:** The system can trigger alerts, isolate faulty sections, or initiate corrective actions based on the prediction.

# RESULT

THE RESULTS OF THE POWER SYSTEM FAULT DETECTION AND CLASSIFICATION BY THE USAGE OF WATSONX.AI STUDIO AS A MACHINE LEARNING MODEL ON IBM CLOUD PLATFORM ARE GIVEN AS THE FOLLOWING:-

# RESULT

The screenshot displays the IBM Watsonx AI Studio web interface. The browser's address bar shows the URL: `eu-gb.dataplatform.cloud.ibm.com/ml/auto-ml/ac1ae12b-8a71-47e1-b5ab-ce3d11f5e7de/configure?projectId=30f7b127-cc6e-426d-8a88-4c3302a7989a&context=cpda...`. The page title is "Configure AutoAI experiment" and the experiment name is "Power\_fault". The interface is divided into two main sections: "Add data source" on the left and "Configure details" on the right.

**Add data source:** This section includes a dashed box with the text "Add files such as tabular data (CSV)." and two buttons: "Browse" and "Select from project". Below this, a file named "fault\_data.csv" is listed with a size of 47.62 KB and 13 columns.

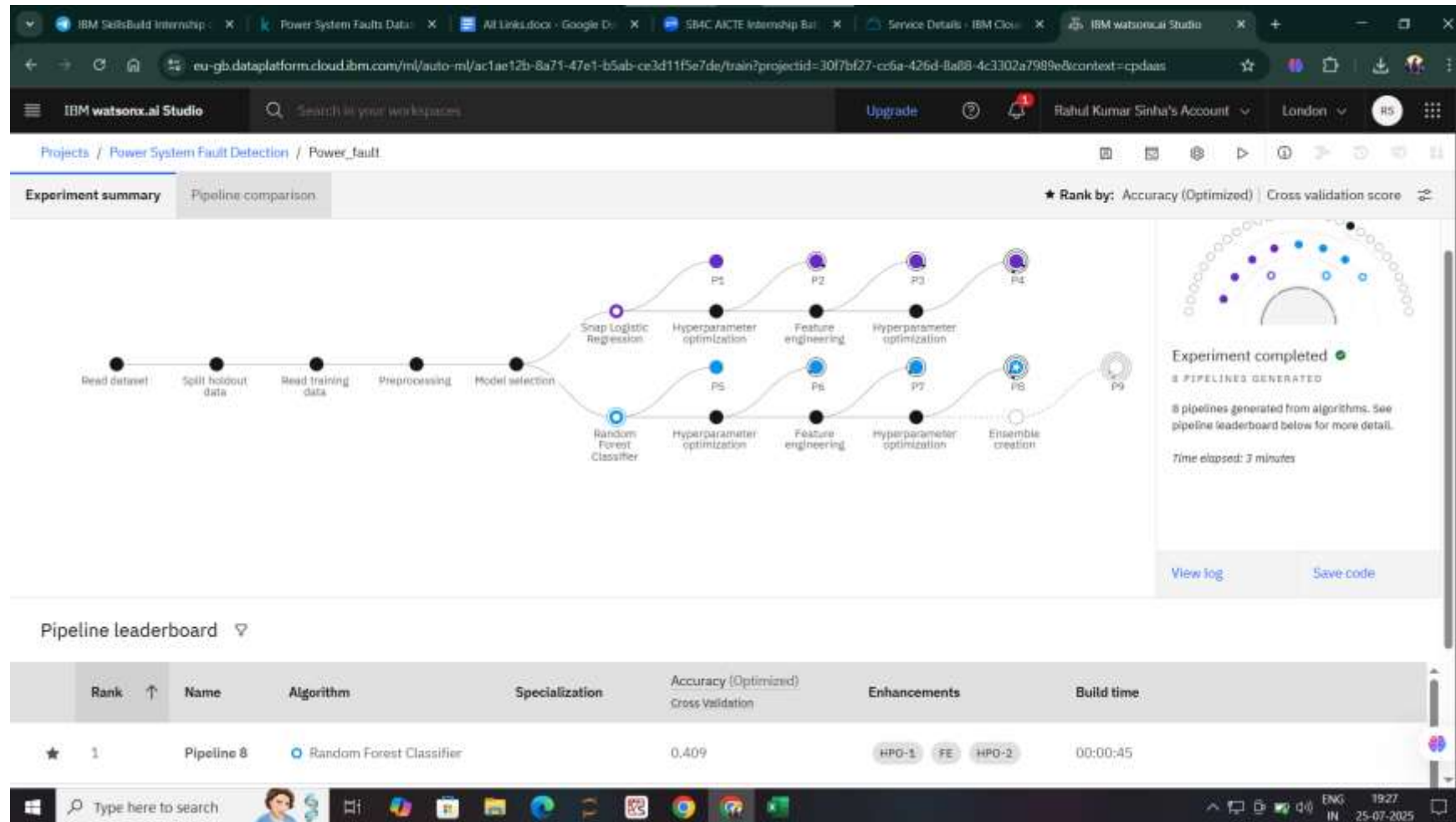
**Configure details:** This section contains several configuration options:

- A toggle switch for "Enable this option to predict future activity over a specified date/time range. Data must be structured and sequential." with "Yes" and "No" buttons.
- A section titled "What do you want to predict?" with a "Prediction column" dropdown menu set to "Fault Type".
- A status indicator "CUH remaining: 14.95 CUH".
- A "PREDICTION TYPE" section showing "Multiclass Classification" and "OPTIMIZED FOR Accuracy & run time".
- A "Run experiment" button at the bottom right.

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the date and time as "25-07-2025 19:21".



# RESULT



# RESULT

The screenshot displays the IBM Watsonx AI Studio interface. The browser address bar shows the URL: `eu-gb.dataplatform.cloud.ibm.com/ml-runtime/deployments/c649aaa7-940c-4813-9691-178ddcb3783d?context=cpdaas&space_id=a7a81c53-3f1a-453b-b3b1-4458ba02d069`. The page title is "PowerFaultDetect\_Deploy" with status "Deployed" and "Online".

**Endpoints for scoring**

- Private endpoint:** `https://private.eu-gb.ml.cloud.ibm.com/ml/v4/deployments/powerfault/predictions?version=2021-05-01`
- Public endpoint:** `https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/powerfault/predictions?version=2021-05-01`

**Code snippets**

Language tabs: cURL, Java, JavaScript, Python, Scala. The cURL tab is active, showing the following commands:

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account (https://eu-gb.dataplatform.cloud.ibm.com/docs/)  
export API_KEY=<your API key>  
  
export IAM_TOKEN=$(curl --insecure -X POST --location "https://iam.cloud.ibm.com/identity/token" \\  
--header "Content-Type: application/x-www-form-urlencoded" \
```

**About this deployment**

- Name:** PowerFaultDetect\_Deploy
- Description:** No description provided.
- Deployment Details:**
  - Deployment ID: c649aaa7-940c-4813-9691-178ddcb3783d
  - Serving name: powerfault
  - Software specification: hybrid\_0.1
  - Hybrid pipeline software specifications: autoai-kb\_rt24.1-py3.11
  - Copies: 1
- Tags:** Add tags to make assets easier to find.
- Associated asset:** PB - Random Forest Classifier: Power\_fault

The Windows taskbar at the bottom shows the search bar and various application icons. The system tray indicates the time is 19:32 on 25-07-2025.

# RESULT

The screenshot displays the IBM Watson Studio interface. At the top, the browser address bar shows the URL: `eu-gb.dataplatform.cloud.ibm.com/ml-runtime/deployments/ce49aea7-940c-4813-9691-178ddcb3783d/test?space_id=a7a81c53-3f1a-453b-b3b1-4458ba02d069&conte...`. The page title is "PowerFaultDetect\_Deploy" with a green "Deployed" status and a "Deploy" button. Below the title, there are tabs for "API reference" and "Test". The "Test" tab is active, showing a section titled "Enter input data". Under this section, there are buttons for "Test" and "Predict". Below these buttons, there is a text input field and a "Predict" button. Below the input field, there is a table with 12 columns and 10 rows. The columns are: Fault ID (other), Fault Location (Latitude, Longitude) (other), Voltage (V) (double), Current (A) (double), Power Load (MW) (double), Temperature (°C) (double), Wind Speed (km/h) (double), Weather Condition (other), Maintenance Status (other), Component Health (other), and Duration. The first row contains the following data: 1, POOL, 22.88, 2200, 300, 60, 38, 33, RAINY, PENDING, NORMAL, 4. The rest of the rows are empty. At the bottom right of the table, there is a "Predict" button. The bottom of the screen shows the Windows taskbar with the search bar and various application icons.

PowerFaultDetect\_Deploy Deployed Deploy

API reference Test

Enter input data

Test Predict

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 60 MB.

[Download CSV template](#) [Browse local file](#) [Search in space](#) Clear all

	Fault ID (other)	Fault Location (Latitude, Longitude) (other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather Condition (other)	Maintenance Status (other)	Component Health (other)	Duration
1	POOL	22.88	2200	300	60	38	33	RAINY	PENDING	NORMAL	4
2											
3											
4											
5											
6											
7											
8											
9											
10											

1 row, 12 columns

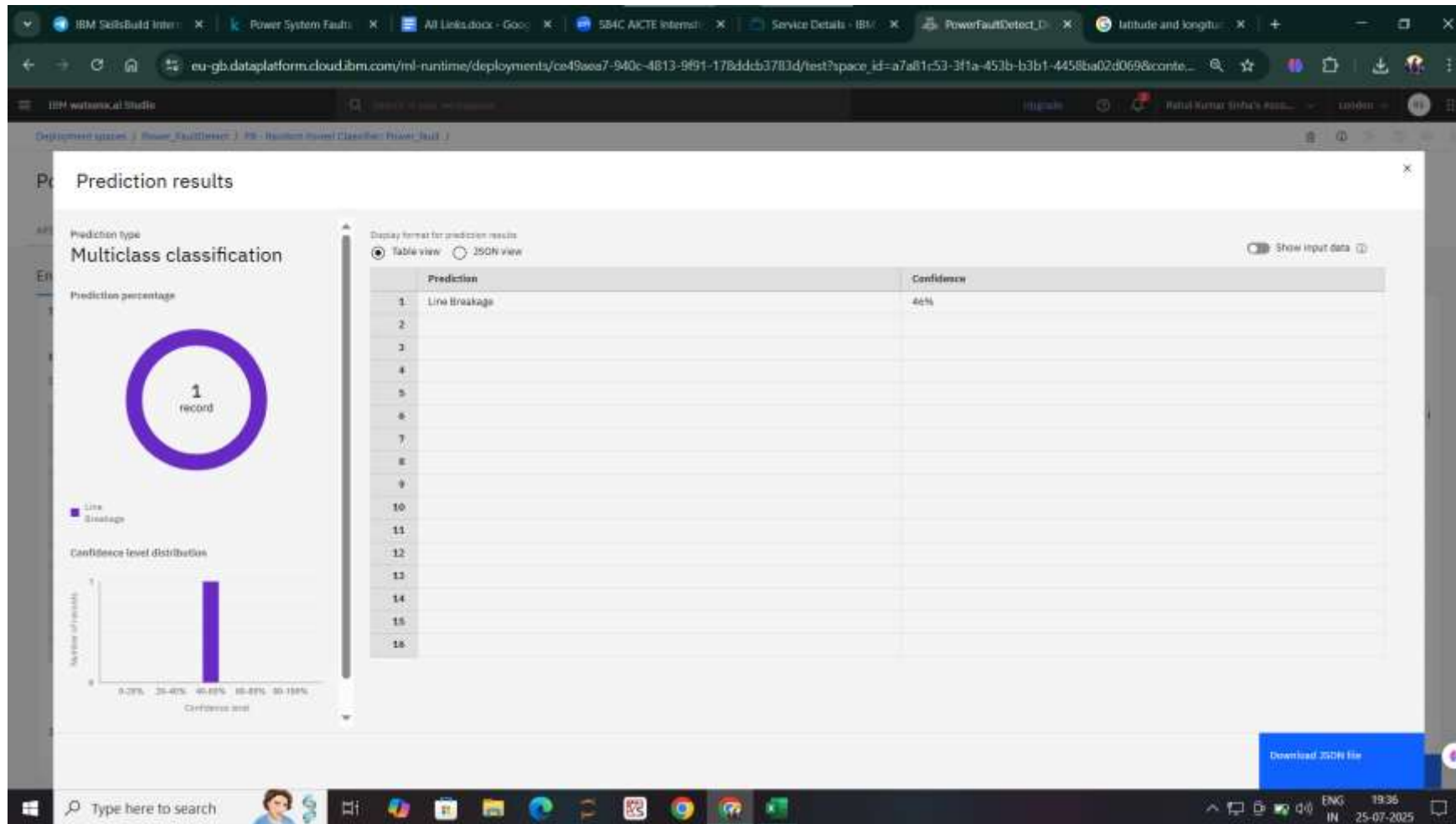
Predict

# RESULT

The screenshot shows the IBM Watson Studio interface for a deployed model named 'PowerFaultDetect\_Deploy'. The 'Test' tab is active, displaying a table for entering input data. The table has 12 columns: (other), Voltage (V) (double), Current (A) (double), Power Load (MW) (double), Temperature (°C) (double), Wind Speed (km/h) (double), Weather Condition (other), Maintenance Status (other), Component Health (other), Duration of Fault (hrs) (double), and Down time (hrs) (double). The first row contains the following values: (other), 2200, 300, 60, 30, 35, RAINY, PENDING, NORMAL, 4, and 5. A 'Predict' button is located at the bottom right of the interface.

(other)	Voltage (V) (double)	Current (A) (double)	Power Load (MW) (double)	Temperature (°C) (double)	Wind Speed (km/h) (double)	Weather Condition (other)	Maintenance Status (other)	Component Health (other)	Duration of Fault (hrs) (double)	Down time (hrs) (double)
1	2200	300	60	30	35	RAINY	PENDING	NORMAL	4	5
2										
3										
4										
5										
6										
7										
8										
9										
10										

# RESULT



# CONCLUSION

- In practical scenarios, utilities might implement this by training a Random Forest with labeled fault data collected during system tests or historical event logs. Once trained, sensors detect anomalies, and the model classifies the faults, enabling rapid and accurate response.
- Implementing a Random Forest classifier for power system faults faces challenges like limited and imbalanced fault data, noise in sensor measurements, selecting relevant features, ensuring real-time processing, adapting to system changes, integration with existing systems, maintaining model accuracy over time, and explaining model decisions. Overcoming these requires good data management, feature engineering, computational resources, and ongoing model updates.

# FUTURE SCOPE

This System can be further modified in future by the implementing the following standards:-

## 1. Integration with IoT and Smart Grids

- Combining fault detection with IoT sensors for real-time, high-resolution monitoring.
- Enabling more accurate and faster fault diagnosis across extensive smart grid networks.

## 2. Hybrid Models

- Integrating Random Forest with other machine learning algorithms (like deep learning) to improve accuracy and handle complex fault scenarios.
- Using ensemble techniques for more robust fault classification.

## 3. Automated Data Collection and Self-Training

- Developing systems that automatically collect and label data, allowing continuous model learning and adaptation to evolving system conditions.

## 4. Enhanced Interpretability

- Improving explainability of model decisions for better trust, regulatory compliance, and system diagnostics.

## 5. Predictive Maintenance

- Transitioning from fault detection to fault prediction, enabling preventive maintenance and reducing downtime.

---

# REFERENCES

## 1.Websites :-

- [www.google.com](http://www.google.com)
- [www.Wikipedia.com](http://www.Wikipedia.com)
- [www.Kaggle.com](http://www.Kaggle.com)

## 2.Books:-

- “Mathematics for Machine Learning” by Deisenroth, Faisal, and Ong



# IBM CERTIFICATIONS



# IBM CERTIFICATIONS



# IBM CERTIFICATIONS



IBM **SkillsBuild**

Completion Certificate

This certificate is presented to

**RAHUL KUMAR SINHA**

for the completion of

**Lab: Retrieval Augmented Generation with LangChain**

(ALM-COURSE\_3824998)



**THANK YOU !!**