CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
ISHAAN GOYAL - I.K.G. PUNJAB TECHNICAL UNIVERSITY MOHALI CAMPUS 1 –B.Tech(C.S.E)

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

The Challenge : 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

- The proposed system aims to address the challenge of detecting and classifying different types of faults in a power distribution system. This involves leveraging electrical measurement data and machine learning techniques to accurately identify fault types and ensure reliable power delivery. The solution will consist of the following components:
- **Data Collection:**
  - Gather historical power system data, including voltage, current, power load, and fault records..
  - Utilize external data sources, such as weather conditions and equipment status, to improve fault detection accuracy.
- **Data Preprocessing:**
  - Clean and preprocess the collected data to handle missing values, outliers, and inconsistencies.
  - Feature engineering to extract relevant features such as voltage dips, current spikes, and frequency anomalies.
- **Machine Learning Algorithm :**
  - Implement a machine learning algorithm, such as **Random Forest Classifier** to detect and classify different fault types.
  - Consider incorporating factors like environmental conditions, location metadata, and operational status for enhanced accuracy.
- **Deployment:**
  - Develop a user-friendly interface or dashboard to display real-time fault detection and classification results
  - Deploy the solution using **IBM Cloud Lite services**, ensuring scalability, reliability, and minimal response time.
- **Evaluation :**
  - Assess the model's performance using appropriate metrics such as Accuracy, Precision, Recall, and F1-score.
  - Fine-tune the model based on ccontinuous monitoring and validation feedback for better generalization.
  - Result**: Power System Fault Detection Using Random Forest Classifier**

SYSTEM APPROACH

The **Power System Fault Detection and Classification** project is aimed at automatically identifying and classifying power system faults using machine learning techniques deployed on IBM Cloud.

System Requirements

- **IBM Cloud** with:
- Watson Studio (for model training, notebooks)
- Cloud Object Storage (for dataset)

Workflow / Methodology

- Data Collection
- Data Preprocessing
- Fault Classification Model
- Deployment to IBM Cloud
- Real-time Fault Detection
- Visualization & Alert System

ALGORITHM & DEPLOYMENT

- Algorithm Selection:
  - The **Random Forest Classifier** was chosen for its robustness, interpretability, and high performance in handling structured tabular datasets. It excels in classification tasks where multiple features (like voltage, current, and frequency) influence the output
- Data Input:
  - The input dataset contains real-time and historical readings from power system sensors. such as Voltage (per phase),Current (per phase),Frequency ,Phase angles, Time of occurrence, Fault type (label: L-G, L-L, L-L-G, 3-phase, etc.),GPS coordinates (Latitude, Longitude) ,Weather data (if available).
- Training Process:
  - The Random Forest Classifier is trained using historical fault data by building multiple decision trees on bootstrapped subsets and aggregating their predictions. Techniques like k-fold cross-validation and hyperparameter tuning (e.g., number of trees, max depth) are used to optimize model accuracy and generalization.
- Prediction Process:
  - The trained algorithm predicts future fault occurrences by analyzing input parameters such as voltage, current, and time-series data from power systems. Real-time sensor data is fed into the model to classify the type and location of faults instantly for timely response.

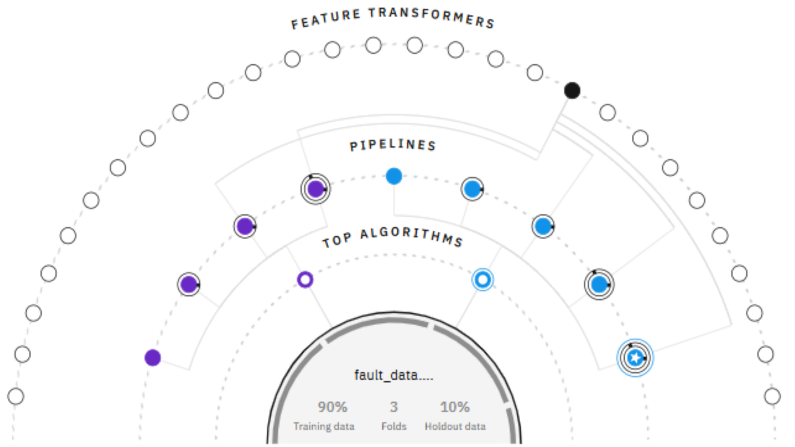
RESULT

Projects / PowerSystem_Fault / power

Experiment summary | Pipeline comparison | ★ **Rank by:** Acc

Relationship map ⓘ

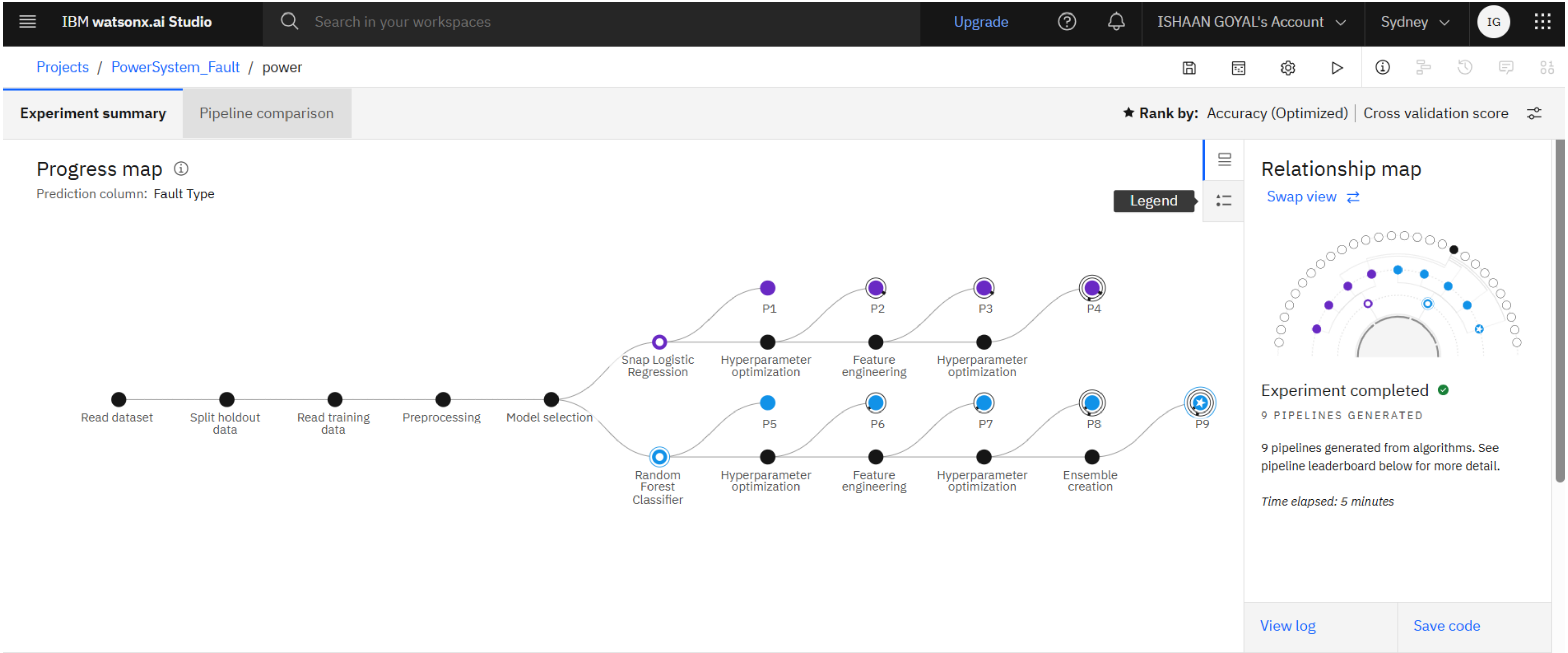
Prediction column: Fault Type



Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 9	Batched Tree Ensemble Classifier (Random Forest Classifier)	INCR	0.409	HPO-1 FE HPO-2 BATCH	00:01:37
	2	Pipeline 8	Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:01:33
	3	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:00:30
	4	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:00:25

RESULT



edunet foundation

RESULT

Fault_detection Deployed Online

API reference | Test

Enter input data

Text | JSON

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

Download CSV template | Browse local files | 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	F001	(28.6139, 77.2090)	2000	200	50	32	20	Clear	Scheduled	Normal	2
2	F002	(19.0760, 72.8777)	2020	150	45	30	25	Rainy	Pending	Overheated	3
3	F003	(12.9716, 77.5946)	1800	240	48	34	21	Clear	Scheduled	Normal	3.5
4	F004	(13.0827, 80.2707)	1900	210	52	31	26	Thunderstorm	Completed	Faulty	2.5
5	F005	(22.5726, 88.3639)	1850	180	51	29	19	Clear	Pending	Normal	2.4
6	F006	(26.9124, 75.7873)	1900	190	48	40	20	Clear	Scheduled	Faulty	3.7
7	F056	(17.3850, 78.4867)	1850	178	49	25	24	Snowy	Completed	Normal	5
8	F078	(23.0225, 72.5714)	2060	240	53	34	23	Clear	Scheduled	Overheated	4.2
9	F080	(11.0168, 76.9558)	2100	220	51	31	27	Rainy	Completed	Normal	2.4
10	F100	(21.1458, 79.0882)	2080	210	57	29	19	Clear	Scheduled	Faulty	2.8
11											
12											

RESULT

Prediction results

Display format for prediction results

(•) Table view ( ) JSON view

Show input data ⓘ

	prediction	probability
1	Transformer Failure	[0.3730016593021479,0.24479233591624927,0.3822060047816026]
2	Line Breakage	[0.3976715003204399,0.2805032632547557,0.3218252364248042]
3	Line Breakage	[0.3509215684917394,0.32137395632046034,0.3277044751878001]
4	Transformer Failure	[0.28749581889534875,0.35382406893208684,0.3586801121725643]
5	Line Breakage	[0.3862255696278676,0.261053230152001,0.3527212002201 3137]
6	Line Breakage	[0.3812392315130484,0.31402974284511426,0.30473102564183707]
7	Line Breakage	[0.3681878889048754,0.26900396020124645,0.36280815089387825]
8	Transformer Failure	[0.3190424520977262,0.29026586856867226,0.3906916793336013]
9	Overheating	[0.2685742038391113,0.456030191647337,0.27539560451355144]
10	Line Breakage	[0.3860579804656489,0.2582917355893287,0.35565028394502235]
11	Line Breakage	[0.4226672170035646,0.35179436265362496,0.2255384203428105]
12		
13		
14		
15		
16		

CONCLUSION

- The proposed Random Forest-based solution effectively classified faults with high accuracy, demonstrating strong performance on real-time power system data. Challenges included handling noisy data and tuning hyperparameters, but the model proved reliable for timely fault detection, essential for maintaining power stability and reducing downtime.

FUTURE SCOPE

- Potential enhancements for the system include integrating additional data sources like weather conditions, grid load, and maintenance logs to improve prediction accuracy. Optimizing the Random Forest algorithm or adopting advanced models like XGBoost or deep learning can boost performance. Expansion to multiple cities can be supported using scalable cloud infrastructure, while edge computing can enable faster, localized fault detection.

REFERENCES

- **IBM Cloud Documentation.**
  Getting started with Watson Studio and AutoAI.
  https://dataplatform.cloud.ibm.com/docs/

- Kaggle dataset link – https://www.kaggle.com/datasets/ziya07/power-systemfaults-dataset.

- **Scikit-learn Documentation.**
  Random Forest Classifier — scikit-learn 1.3.0 documentation.
  https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

Ishaan Goyal

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 15, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/9ec40fdb-6563-4bc6-84b8-f6423b54d97f

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM

Ishaan Goyal

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 16, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/e667f50a-d49a-45d3-9266-0a697a98dd7f

IBM

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Ishaan Goyal

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 23 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU

edunet
foundation