CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:

1. Student Name: Rohan Lotan Ahire
2. College Name: DR.B.V.HIRAY COLLEGE OF MANAGEMENT AND RESEARCH CENTRE
3. Department: AIML

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

- 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

- Rapid identification and classification of electrical faults (such as line-to-ground, line-to-line, and three-phase faults) in power distribution systems is essential to prevent blackouts, reduce damage, and ensure grid stability. Manual or rule-based systems are often slow or inaccurate.
- The solution will consist of the following components:
- Data Collection:
  - Download dataset from kraggle on power system faults,
  - Upload it to IBM Cloud Object Storage (Lite Tier).
  - Utilize real-time data sources, such as weather conditions, events, and holidays, to enhance prediction accuracy.
- Data Preprocessing:
  - Create a Wotson Studio Project.
  - Clean and normalize the dataset.
- Model Building:
- Use supervised ML classification algorithms:
  - Random Forest.
  - Support Vector Machine (SVM)
- Evaluation:
  - To evaluate the accuracy and effectiveness of the machine learning model trained to classify faults (e.g., LG, LL, LLG, 3-phase) based on voltage and current phasor data using IBM Cloud.

SYSTEM APPROACH

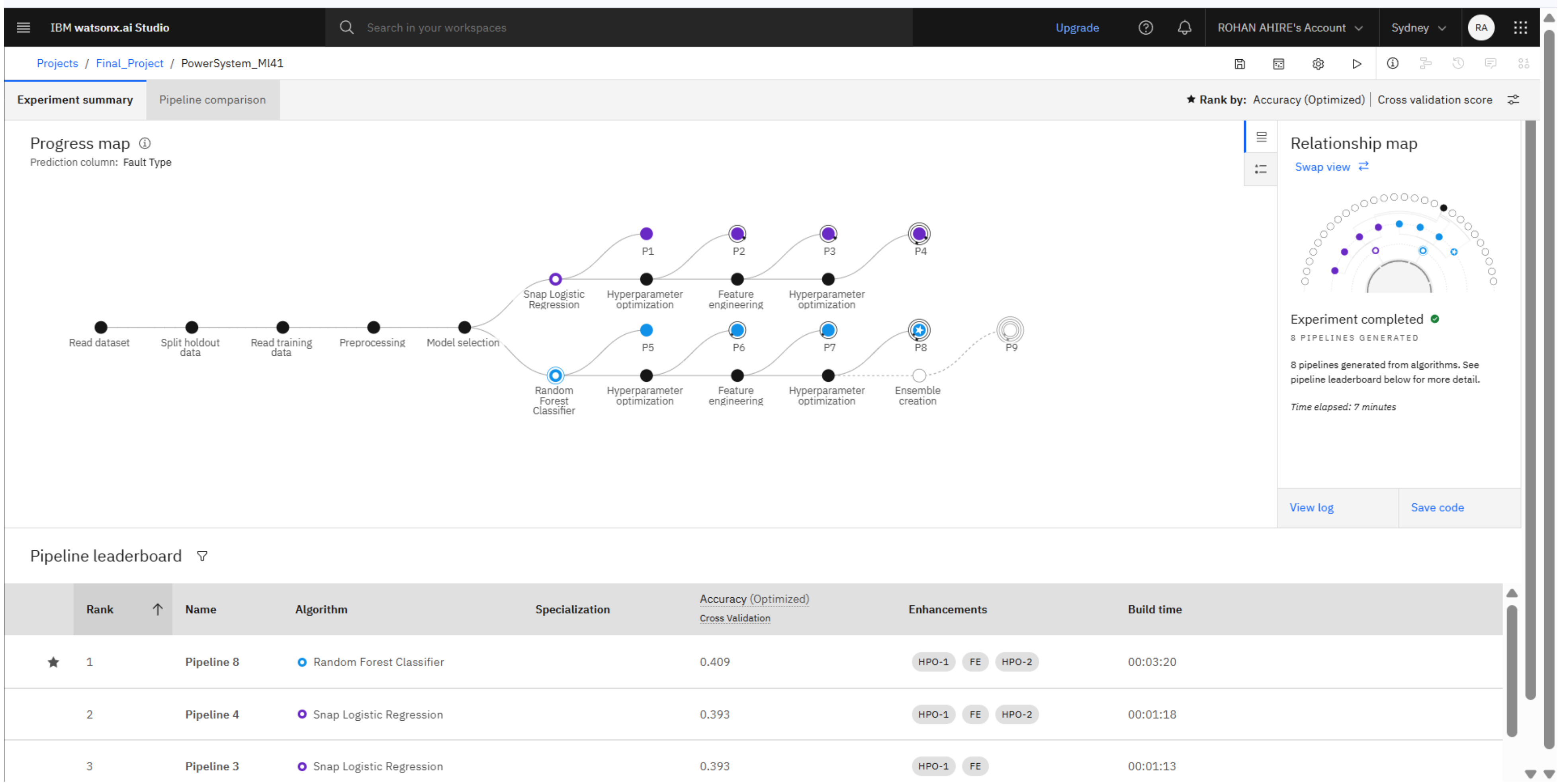
The "System Approach" section outlines the overall strategy and methodology for developing and implementing the rental bike prediction system. Here's a suggested structure for this section:

- System requirements:
  - IBM Cloud (mandatory)
  - IBM Watson studio for model development and deployment
  - IBM cloud object storage for dataset handling

ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting bike counts. Here's an example structure for this section:
- **Algorithm Selection:**
- Random Forest Classifier
- **Data Input:**
- Voltage, current, and phasor measurements from the dataset
- **Training Process:**
- Supervised learning using labeled fault types
- **Prediction Process:**
- Model deployed on IBM Watson Studio with API endpoint for real-time predictions.

RESULT


IBM watsonx.ai Studio
Search in your workspaces
Upgrade
ROHAN AHIRE's Account
Sydney
RA
Projects / Final_Project / PowerSystem_Ml41
Experiment summary
Pipeline comparison
Rank by: Accuracy (Optimized) | Cross validation score
Progress map
Prediction column: Fault Type
Read dataset
Split holdout data
Read training data
Preprocessing
Model selection
Snap Logistic Regression
Random Forest Classifier
Hyperparameter optimization
Feature engineering
Ensemble creation
P1
P2
P3
P4
P5
P6
P7
P8
P9
Relationship map
Swap view
Experiment completed
8 PIPELINES GENERATED
8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.
Time elapsed: 7 minutes
View log
Save code
Pipeline leaderboard
Rank
Name
Algorithm
Specialization
Accuracy (Optimized)
Cross Validation
Enhancements
Build time
1
Pipeline 8
Random Forest Classifier
0.409
HPO-1 FE HPO-2
00:03:20
2
Pipeline 4
Snap Logistic Regression
0.393
HPO-1 FE HPO-2
00:01:18
3
Pipeline 3
Snap Logistic Regression
0.393
HPO-1 FE
00:01:13

RESULT

IBM watsonx.ai Studio | Search in your workspaces | Upgrade | ROHAN AHIRE's Account | Sydney | RA

Deployment spaces / Power_Fault41 / P8 - Random Forest Classifier: PowerSystem_Ml41 /

Power_fault41_Dep

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 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 of Fault (hrs) (double)	Down time (hrs) (double
1	(34.0522, -118.2437)	2200	250	50	25	20	clear	Scheduled	Normal	2	1
2	(34.056, -118.245)	1800	180	45	28	15	Rainy	Completed	Faulty	3	5
3	(34.0525, -118.244)	2100	230	55	35	25	Windstorm	Pending	Overheated	4	6
4	(34.055, -118.242)	2050	240	48	23	10	Clear	Completed	Normal	2.5	3
5	(34.0545, -118.243)	1900	190	50	30	18	Snowy	Scheduled	Faulty	3.5	4
6	(34.05, -118.24)	2150	220	52	32	22	Thunderstorm	Pending	OverHeated	5	7
7	(34.9449, -118.9839)	1994	233	51	23	21	snowy	Completed	Normal	3.7	6.1
8	(34.2294, -118.2988)	2133	229	52	10	18	Snowy	Scheduled	Normal	5.4	2.1
9	(34.1279, -118.8442)	2155	240	45	21	29	Rainy	Pending	Overheated	3.2	4.7
10	(34.4192, -118.8254)	2165	199	47	25	21	Clear	Scheduled	Normal	4	2.8
11											

10 rows, 12 columns

Predict

edunet foundation

RESULT

Prediction results

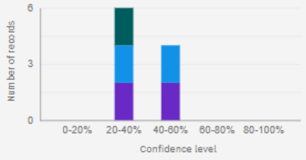
Prediction type

Multiclass classification

Prediction percentage

Line Breakage · Transformer Failure · Overheating

Confidence level distribution



Line Breakage · Transformer Failure · Overheating

Display format for prediction results

Table view · JSON view

Show input data

	Prediction	Confidence
1	Line Breakage	39%
2	Transformer Failure	35%
3	Overheating	37%
4	Line Breakage	54%
5	Transformer Failure	38%
6	Line Breakage	35%
7	Line Breakage	41%
8	Transformer Failure	46%
9	Transformer Failure	41%
10	Overheating	38%
11		
12		
13		
14		
15		
16		

Download JSON file

RESULT

IBM watsonx.ai Studio

Search in your workspaces

Upgrade

ROHAN AHIRE's Account

Sydney

Projects / Final_Project / PowerSystem_Ml41

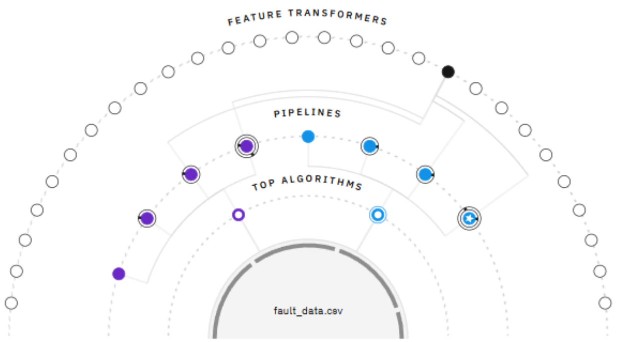
Experiment summary

Pipeline comparison

★ Rank by: Accuracy (Optimized) | Cross validation score

Relationship map

Prediction column: Fault Type



Progress map

Swap view

Experiment completed

8 PIPELINES GENERATED

8 pipelines generated from algorithms. See pipeline leaderboard below for more detail.

Time elapsed: 7 minutes

View log

Save code

Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 8	Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:03:20
	2	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:01:18
	3	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:01:13
	4	Pipeline 7	Random Forest Classifier		0.376	HPO-1 FE	00:00:30
	5	Pipeline 6	Random Forest Classifier		0.369	HPO-1	00:00:08
	6	Pipeline 2	Snap Logistic Regression		0.367	HPO-1	00:00:08
	7	Pipeline 5	Random Forest Classifier		0.360	*None*	00:00:02

edunet foundation

CONCLUSION

- This approach not only enables **rapid fault identification** but also supports **automated decision-making** in real-time, reducing system downtime and improving the resilience of the power grid. With the integration of advanced feature engineering, robust classification algorithms, and real-time deployment capabilities, such a system plays a crucial role in the modernization of smart grids and energy management systems.
- In essence, machine learning empowers utility companies to move from reactive to **predictive maintenance**, ensuring safer, smarter, and more sustainable power distribution networks.

FUTURE SCOPE

- 1. Integration with Smart Grids
  - Seamless integration with IoT-based smart grids for real-time monitoring and control.
  - Use of edge computing for local fault detection, reducing latency and dependency on centralized systems.
- 2. Advanced Deep Learning Models
  - Implementing Recurrent Neural Networks (RNNs), LSTMs, or Transformers to capture temporal dependencies in time-series data.
  - Using Graph Neural Networks (GNNs) to model the power grid topology and improve fault localization.
- 3. Transfer Learning & Domain Adaptation
  - Applying transfer learning to adapt models across different substations or networks with minimal retraining.
  - Facilitating model generalization across varied environmental and grid conditions.
- 4. Explainable AI (XAI)
  - Enhancing model interpretability and transparency for utility operators.
  - Providing visual explanations for fault classifications to support human-in-the-loop systems.
- 5. Self-Healing Networks
  - Integration with automated protection and control systems to isolate and correct faults without human intervention.
  - Moving toward self-healing grid architectures that can reconfigure themselves dynamically.
- 6. Cybersecurity-Aware Fault Detection
  - Combining fault detection with anomaly detection systems to identify and differentiate between physical faults and cyberattacks on grid infrastructure.
- 7. Big Data Analytics and Cloud Integration
  - Using cloud platforms for large-scale data storage and model training.
  - Employing real-time analytics to process high-frequency phasor measurement unit (PMU) data.

REFERENCES

- **1. IBM Watson Studio**
- **Description:** An integrated environment to build, train, and deploy AI and ML models using popular frameworks like TensorFlow, PyTorch, and Scikit-learn.
- **Relevance:** You can upload your power system measurement data, preprocess it, and experiment with various ML algorithms for fault classification.
- **Try Hands-on:** Use Watson Studio's AutoAI to automate feature engineering and model selection on your dataset.
- **2. IBM Cloud Pak for Data**
- **Description:** A data and AI platform that simplifies the collection, organization, and analysis of data with built-in AI tools.
- **Relevance:** Great for managing large power system datasets and creating scalable ML pipelines.
- **Use case:** You can build predictive maintenance workflows or fault detection pipelines with real-time data streaming.
- **3. IBM AI Engineering Professional Certificate (Coursera)**
- **What it offers:** End-to-end AI engineering, including data handling, modeling, and deployment on IBM Cloud.
- **Relevance:** The course contains practical labs on classification problems, which can be directly related to classifying fault types from electrical data.
- **4. IBM Tutorials on Time Series Analysis and Anomaly Detection**
- **Details:** Since fault detection often involves time-series sensor data, IBM tutorials on time series forecasting and anomaly detection are invaluable.
- **Use cases:** Detect deviations (faults) from normal operating conditions using ML models.

IBM CERTIFICATIONS

- Credly certificate(getting started with AI)

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

Rohan Ahire

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/f6b17fda-dcdc-44d5-8953-e8424c128fa4

IBM

IBM CERTIFICATIONS

- Credly certificate(Journey to Cloud)

edunet foundation

IBM CERTIFICATIONS

- Credly certificate(RAG Lab)

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Rohan Ahire

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