CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
Rishu-Hemvati Nandan Bahuguna Garhwal University-Computer Science & Engineering(AI/ML)

edunet foundation

OUTLINE

- Problem Statement
- Proposed 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 must distinguish between:

- Normal operating conditions
- Fault types:
    - Line-to-ground
    - Line-to-line
    - Three-phase faults

Goal: Enable rapid and accurate fault identification to support grid stability and reliability.

PROPOSED SOLUTION

The proposed model will:

- Ingest electrical measurement data.
- Preprocess and clean the dataset.
- Use supervised machine learning to classify fault types.
- Provide real-time or near-real-time fault classification.
- Be deployed using IBM Cloud Lite services.

SYSTEM APPROACH

Technology Used:

- Python (Data processing, model training)
- IBM Cloud Lite (Deployment platform)
- Jupyter Notebook (Exploratory analysis)

Libraries:

- NumPy, Pandas, Scikit-learn
- Matplotlib, Seaborn (visualization)

Dataset:

Kaggle Dataset:
(https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset)

ALGORITHM & DEPLOYMENT

Algorithm Selection:

- Random Forest / Decision Tree / SVM

Input Features:

- Voltage measurements
- Current phasors & Fault duration and type

Training:

- Data split into train/test sets
- Cross-validation applied

Deployment:

- Model exported as .pkl
- Web UI (optional) hosted on IBM Cloud Lite

RESULT

Best Performing Model: Random Forest Classifier
Accuracy (Cross Validation): 0.409
Other Models Tested:

- Snap Logistic Regression (Accuracy: 0.393)
- Another Random Forest variant (Accuracy: 0.376)

Enhancements Used:

- Hyperparameter Optimization
- Feature Engineering

RESULT

FAULT DETECTION

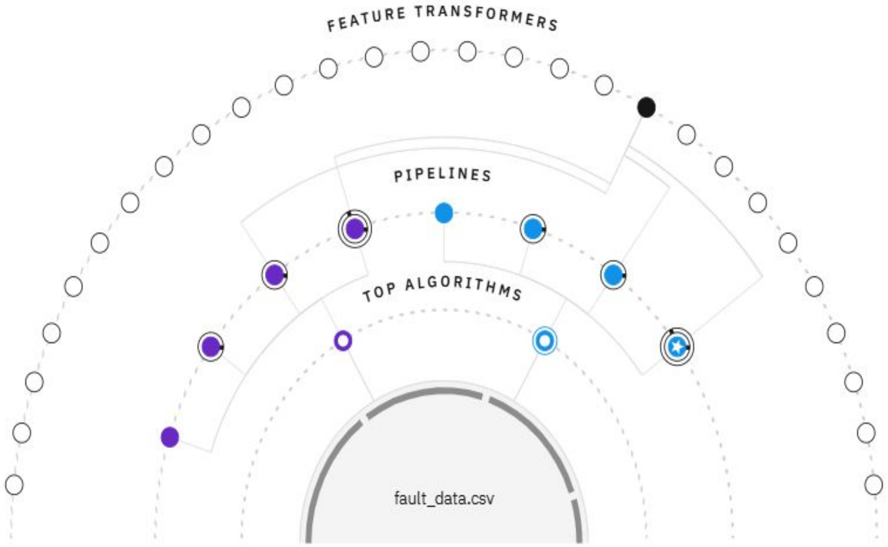
Projects / IBM_Final project / Power System Fault Detection and Classification_ML

Experiment summary | Pipeline comparison

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

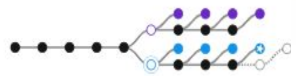
Relationship map

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.

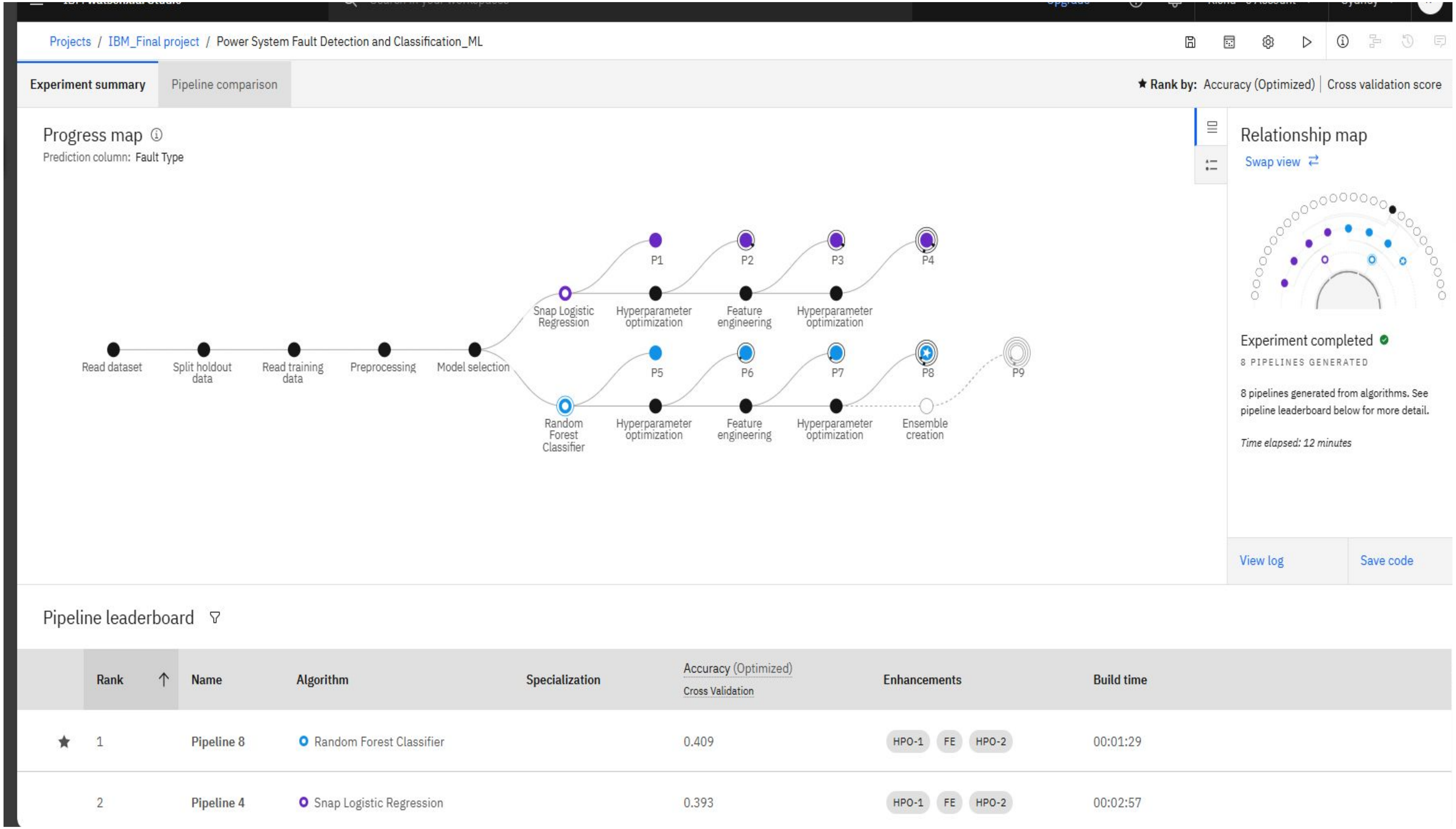
Time elapsed: 12 minutes

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	00:01:29
	2	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:02:57

RESULT



Projects / IBM_Final project / Power System Fault Detection and Classification_ML
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
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
P1
P2
P3
P4
Random Forest Classifier
Hyperparameter optimization
Feature engineering
Hyperparameter optimization
Ensemble creation
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: 12 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 00:01:29
2 Pipeline 4 Snap Logistic Regression 0.393 HPO-1 FE HPO-2 00:02:57

RESULT

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	00:01:29
	2	**Pipeline 4**	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:02:57
	3	**Pipeline 3**	Snap Logistic Regression		0.393	HPO-1 FE	00:02:03
	4	**Pipeline 7**	Random Forest Classifier		0.376	HPO-1 FE	00:01:18

RESULT

Deployment spaces / Power_Deployment / P8 - Random Forest Classifier: Power System Fault Detection and Classification_ML /

Power_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 ×

	l (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		25	20	clear	Scheduled	Normal	2	1
2		28	15	Rainy	Completed	Faulty	3	5
3		35	25	Windstorm	Pending	Overheated	4	6
4		30	18	Snowy	Scheduled	Faulty	3.5	4
5		32	22	Thunderstorm	Pending	Overheated	5	7
6		24	29	Clear	Completed	Faulty	3.9	6.4
7								
8								
9								

6 rows, 12 columns

au-syd.dai.cloud.ibm.com/.../81be56c3-d185-4feb-ba05-c640049696a8?co...

Predict

RESULT

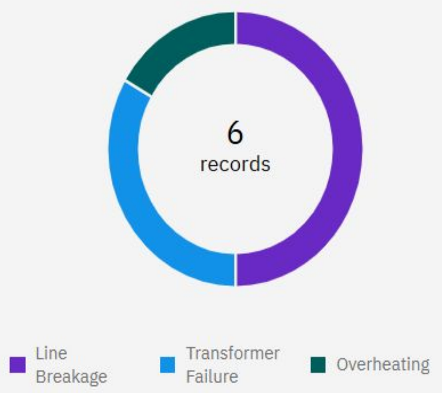
Deployment spaces / Power_Deployment / P8 - Random Forest Classifier: Power System Fault Detection and Classification_ML /

Prediction results

Prediction type

Multiclass classification

Prediction percentage



Confidence level distribution

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	Transformer Failure	38%
5	Line Breakage	35%
6	Line Breakage	46%
7		
8		
9		
10		
11		
12		

Download JSON file

edunet foundation

CONCLUSION

- Fault classification using ML models is feasible and efficient.
- Accurate fault type detection improves grid reliability.
- ML techniques can complement traditional fault detection systems.
- IBM Cloud enables smooth deployment and accessibility.
- Random Forest Classifier was most effective with an accuracy of 40.9%.
- AutoAI automated feature selection, optimization, and model testing.
- IBM Cloud Lite successfully supported the deployment.
- Real-time classification of power system faults is feasible but can be further improved with additional data or ensemble methods.

FUTURE SCOPE

- ❑ Integration with real-time SCADA systems.
- ❑ Use of deep learning (e.g., LSTM) for time-series data.
- ❑ Fault severity prediction.
- ❑ Deployment in industrial control environments.

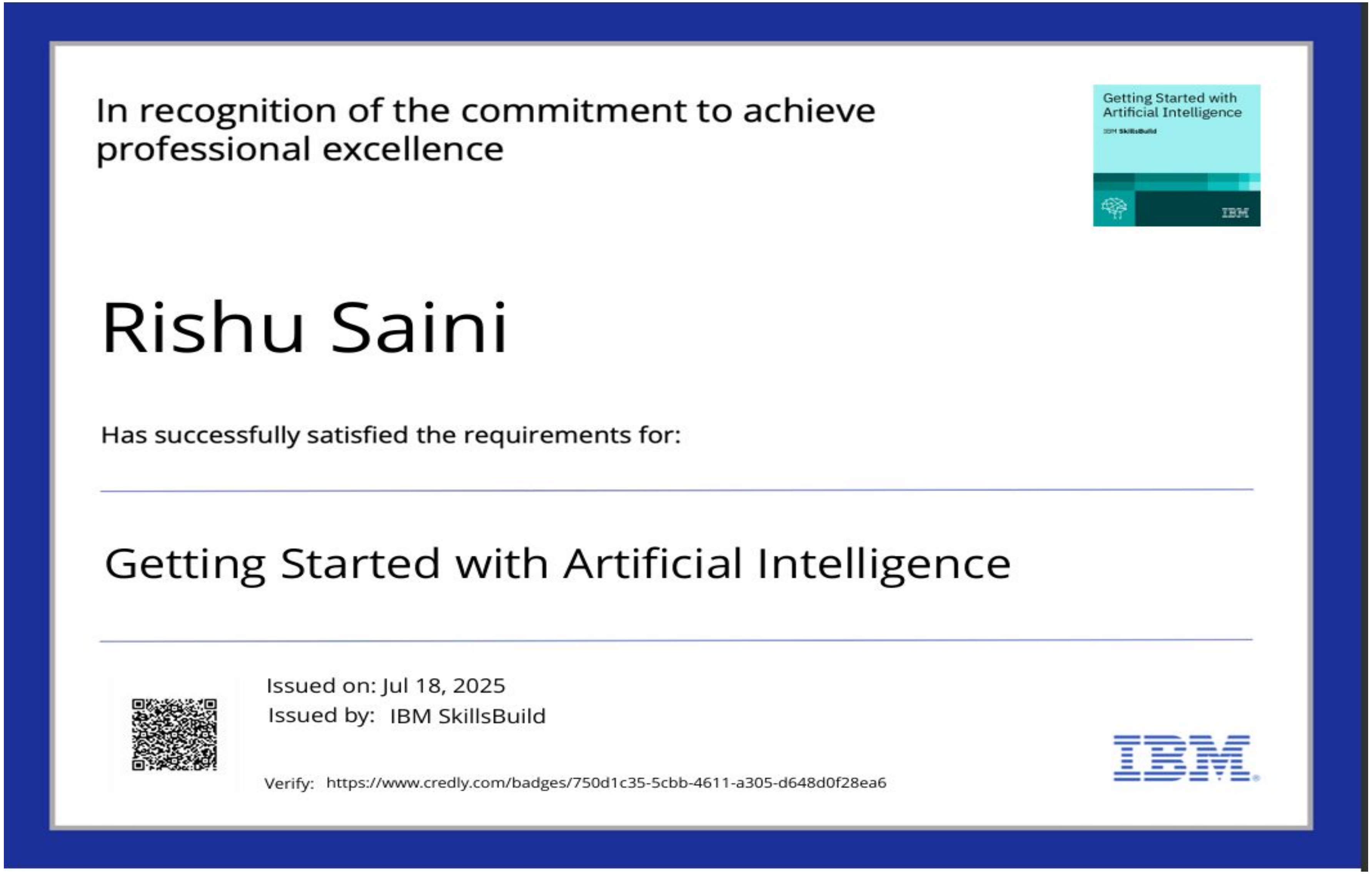
REFERENCES

- Kaggle Dataset: https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset
- Research papers on power fault classification
- Scikit-learn documentation
- IBM Cloud Documentation
- **IBM Cloud Lite**

  Deployment of Machine Learning Models using IBM Cloud Lite

  https://cloud.ibm.com

IBM CERTIFICATIONS



In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM

Rishu Saini

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 18, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/750d1c35-5cbb-4611-a305-d648d0f28ea6

IBM

FAULT DETECTION

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Journey to Cloud: Envisioning Your Solution

IBM SkillsBuild

IBM

Rishu Saini

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/34743822-3bc9-45a4-b821-ae948e1b072b

IBM

FAULT DETECTION

IBM CERTIFICATIONS

IBM SkillsBuild Completion Certificate

This certificate is presented to

Rishu Saini

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 !