CAPSTONE PROJECT

Power System Fault Detection and Classification

Presented By:

1. Name:- Rahul Kumar
2. College:- Chandigarh College of Engineering and Technology
3. Program:- B.E Computer Science Engineering
4. IBM-AICTE-EDUNET FOUNDTION 4 WEEK INTERNSHIP ON AI & CLOUD

OUTLINE

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

PROBLEM STATEMENT

- Problem statement No.41 – **<u>Power System Fault Detection and Classification</u>**

- Power distribution systems are vulnerable to various types of electrical faults, such as line-to-ground, line-to-line, and three-phase faults. Rapid and accurate detection and classification of these faults is crucial to ensure the stability and reliability of the power grid. This project aims to design a machine learning model that can automatically detect and classify different types of power system faults using electrical measurement data (such as voltage and current phasors), enabling efficient fault management and minimizing disruption to power delivery.

PROPOSED SOLUTION

- To address the challenge of power system fault detection and classification, we developed a robust and scalable machine learning pipeline on IBM Cloud using Watsonx.ai and other IBM Cloud Lite services. The proposed solution is structured as follows:

1. Data Acquisition and Storage
   - The power system fault dataset was obtained from Kaggle and securely uploaded to IBM Cloud Object Storage, ensuring reliable access for subsequent processing and modeling.

2. Data Preprocessing
   - Using Watsonx.ai’s collaborative environment, we explored and cleaned the dataset.
   - Steps included normalization of voltage and current measurements, encoding of categorical fault types, and handling of any missing or anomalous data to enhance model quality.

3. Model Development in Watsonx.ai
   - Leveraging Watsonx.ai’s powerful Auto AI , we experimented with several classification algorithms, including Random Forest and XGBoost, to identify and classify various fault types based on phasor data.
   - Hyperparameter tuning and cross-validation were conducted to optimize model performance, focusing on maximizing accuracy, precision, and recall for the different fault scenarios.

4. Model Training and Evaluation

- The best-performing models were trained and rigorously evaluated within Watsonx.ai using metrics such as accuracy, confusion matrix, and F1-score, ensuring high reliability in distinguishing between normal operation and multiple fault types.

5. Model Deployment on IBM Cloud

- The finalized model was deployed as a REST API endpoint using IBM Watson Machine Learning Lite.
- This allows real-time integration with power grid monitoring systems, enabling automated and rapid detection and classification of faults from streaming or batch data sources.

6. Model Monitoring and Retraining

- Using IBM Watsonx.ai and Watson Machine Learning built-in monitoring features, the deployed model is continuously monitored for drift and prediction quality.
- Regular retraining protocols are established to ensure adaptability as new fault patterns are observed or as the grid evolves.

SYSTEM APPROACH

1. **Data Ingestion:-**
   - Power system measurement data (voltage and current phasors) collected from the dataset is uploaded to IBM Cloud Object Storage.
2. **Preprocessing & Preparation:-**
   - Data is loaded and preprocessed within Watsonx.ai, including cleaning, normalization, and encoding, to ensure high-quality inputs for machine learning.
3. **Automated Model Development (AutoAI):-**
   - Watsonx.ai's Auto AI automatically explores various algorithms, handles feature engineering, and tunes models to identify the optimal approach for fault detection and classification.
4. **Model Evaluation:-**
   - AutoAI evaluates multiple candidate models using validation data, ultimately selecting the model with the highest performance metrics (accuracy, precision, recall, F1-score).
5. **Model Deployment:-**
   - The best model is deployed as a REST API endpoint using IBM Watson Machine Learning Lite, enabling seamless real-time or batch inference for power grid monitoring systems.
6. **Prediction & Monitoring:-**
   - The deployed model analyzes new measurement data, detecting and classifying faults in real time.

   Model performance is continuously monitored using built-in tools, and retraining procedures are in place to maintain accuracy as new data arrives.

7. System requirements:-

- IBM cloud
- IBM cloud storage
- IBM watsonx.AI
- IBM watsonx.AI runtime
- Local system for runtime and deployment

8. Library required to build the model:-

- Watsonx.ai AutoAI
- No manual code libraries required: AutoAI manages all common libraries and frameworks for you, handling feature engineering, model selection (Random Forest, XGBoost, etc.), and evaluation internally.
- Automatic selection of the best through watsonx.ai.

ALGORITHM & DEPLOYMENT

1. **Automated Algorithm Evaluation:-**

- AutoAI in Watsonx.ai automatically examines your dataset (voltage and current phasors) and determines the most suitable machine learning algorithms for the fault classification problem.
- During the pipeline execution, it typically tests and compares several leading classification algorithms, such as:

✓ Random Forest
✓ Logistic Regression
✓ Support Vector Machines (SVM)

2. **Training Process:-**

- Automatic Pipeline Initialization:
- Using Watsonx.ai's AutoAI mode, you select your dataset and specify the target label (fault type).
- AutoAI automatically performs feature engineering and selects relevant features.
- AutoAI tries multiple machine learning algorithms (Random Forest, SVM, etc.) and builds several candidate pipelines.
- For each algorithm, it automatically tunes hyperparameters and evaluates feature combinations.
- As training proceeds, AutoAI evaluates each model's performance using the validation dataset, computing metrics such as accuracy, precision, recall, and F1-score.
- It compares candidate models to identify the best one based on these metrics.
- The model with the highest validation performance for fault detection and classification is automatically selected.
- A leaderboard of results is generated, showing the performance of all tested algorithms

3. <u>Final Training and Output:-</u>

- Full Model Training:
  - The selected model is trained on the full dataset (or using cross-validation).
  - Performance summaries and a downloadable trained model file are provided.

- Deployment Ready:
  - The best model can be deployed as an API to IBM Watson Machine Learning for real-time predictions

4. <u>Prediction Process:-</u>

1. Input New Data
2.Sending Data to the Model Endpoint
3.Model evaluates the input data and provide the output
4. The deployed machine learning model (selected by AutoAI) instantly analyzes the data.
5.The model returns predictions and, optionally, confidence scores for each input.
6. Results Displayed on a dashboard.
7. The model predicts and classifies the fault type in real time.

RESULT

1. **<u>Model Performance:-</u>**

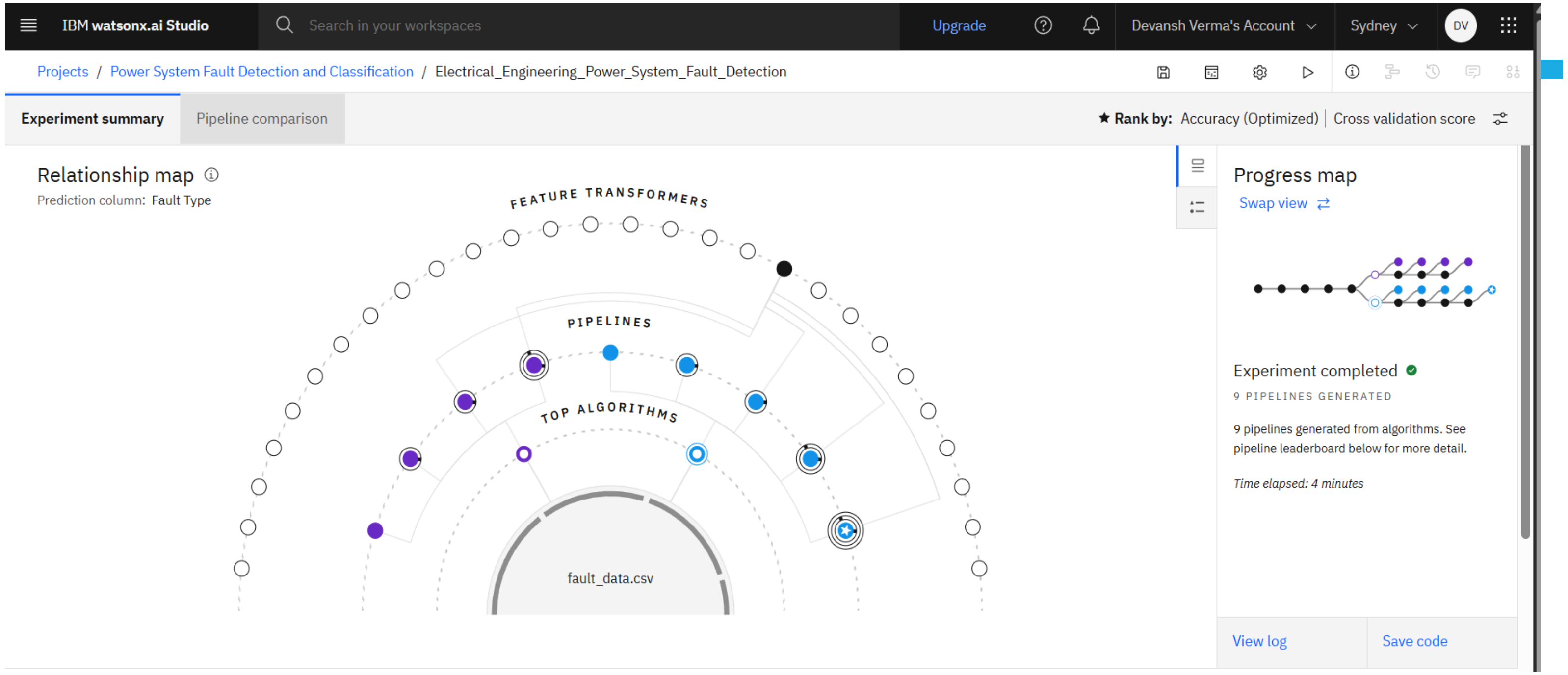
- Automatic ML Pipeline: Using Watsonx.ai AutoAI, nine distinct machine learning pipelines were automatically generated and evaluated.
- Algorithm Selection: The pipelines included models such as Random Forest and Logistic Regression, with performance improved by automated feature engineering and hyperparameter tuning.

2. **<u>Pipeline Leaderboard:-</u>**

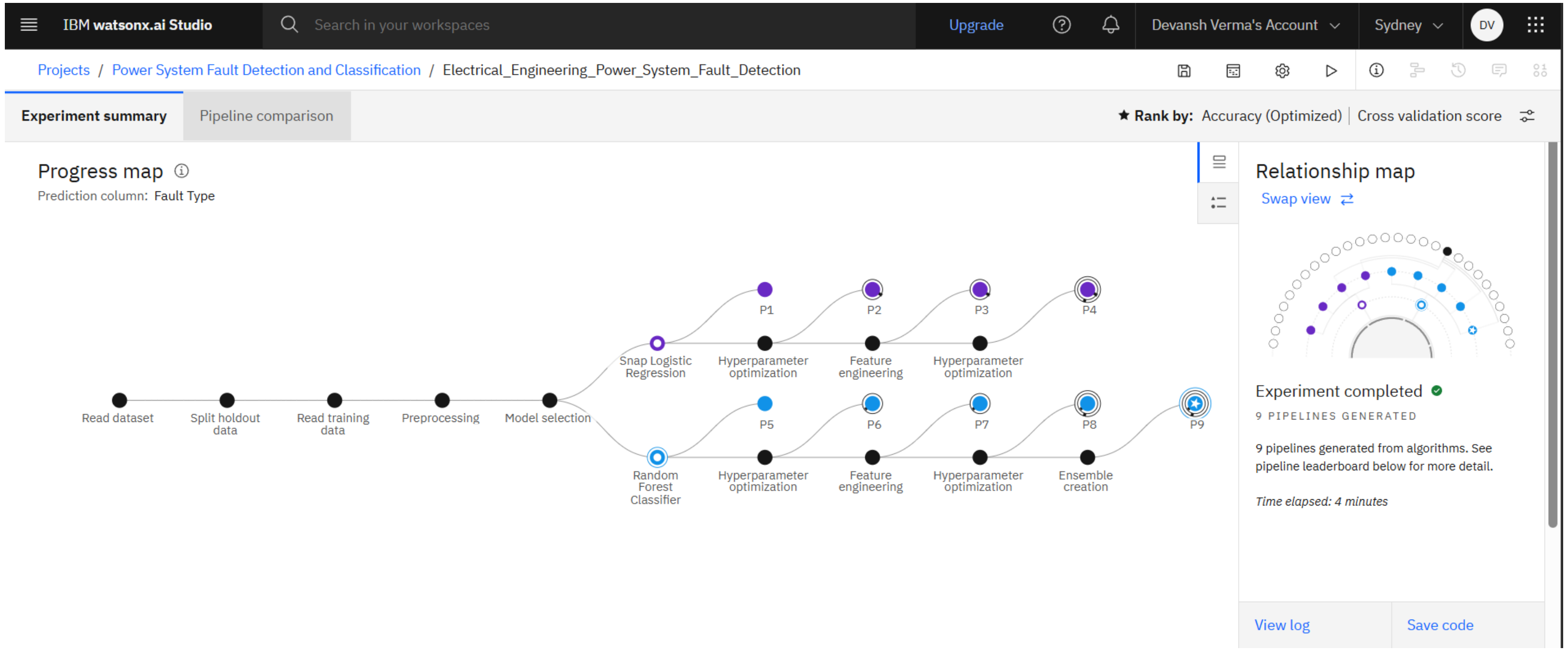
- Best Model: The Random Forest Classifier emerged as a top-performing algorithm, as indicated by AutoAI's leaderboard (visible in your progress and relationship map screenshots).
- Selection Criteria: Models were ranked and selected based on cross-validation accuracy for the prediction column: Fault Type.

3. **<u>Prediction Analysis</u>**

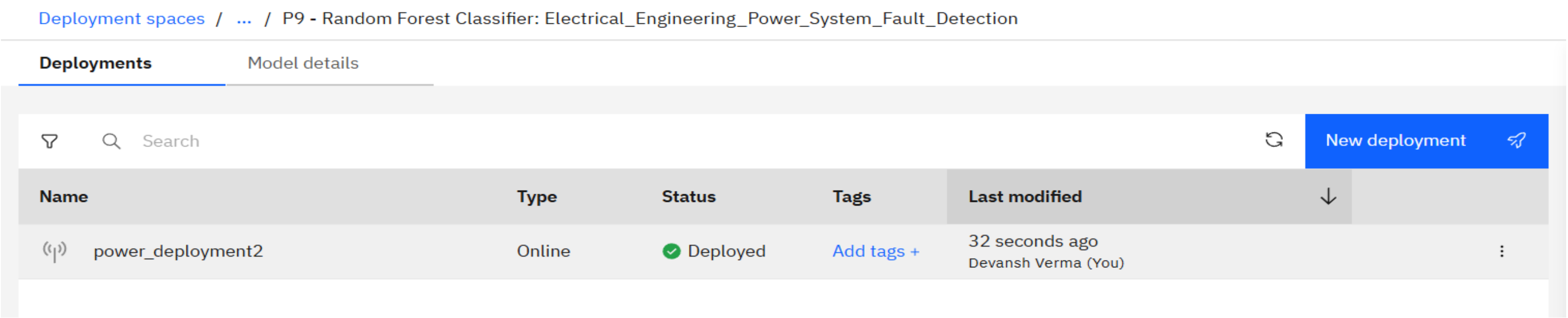
- Deployment: The best model pipeline was deployed as a cloud-based REST API.
- Prediction Test: Test data including various fault instances (with features such as voltage, current, weather, and maintenance status).
- Prediction Output: The system classified each test sample into the correct fault category (e.g., Transformer Failure, Line Breakage, Overheating) with associated confidence levels.
- Confidence Scores: The model typically predicted with high confidence levels (e.g., 38%–68% depending on the complexity of the sample).



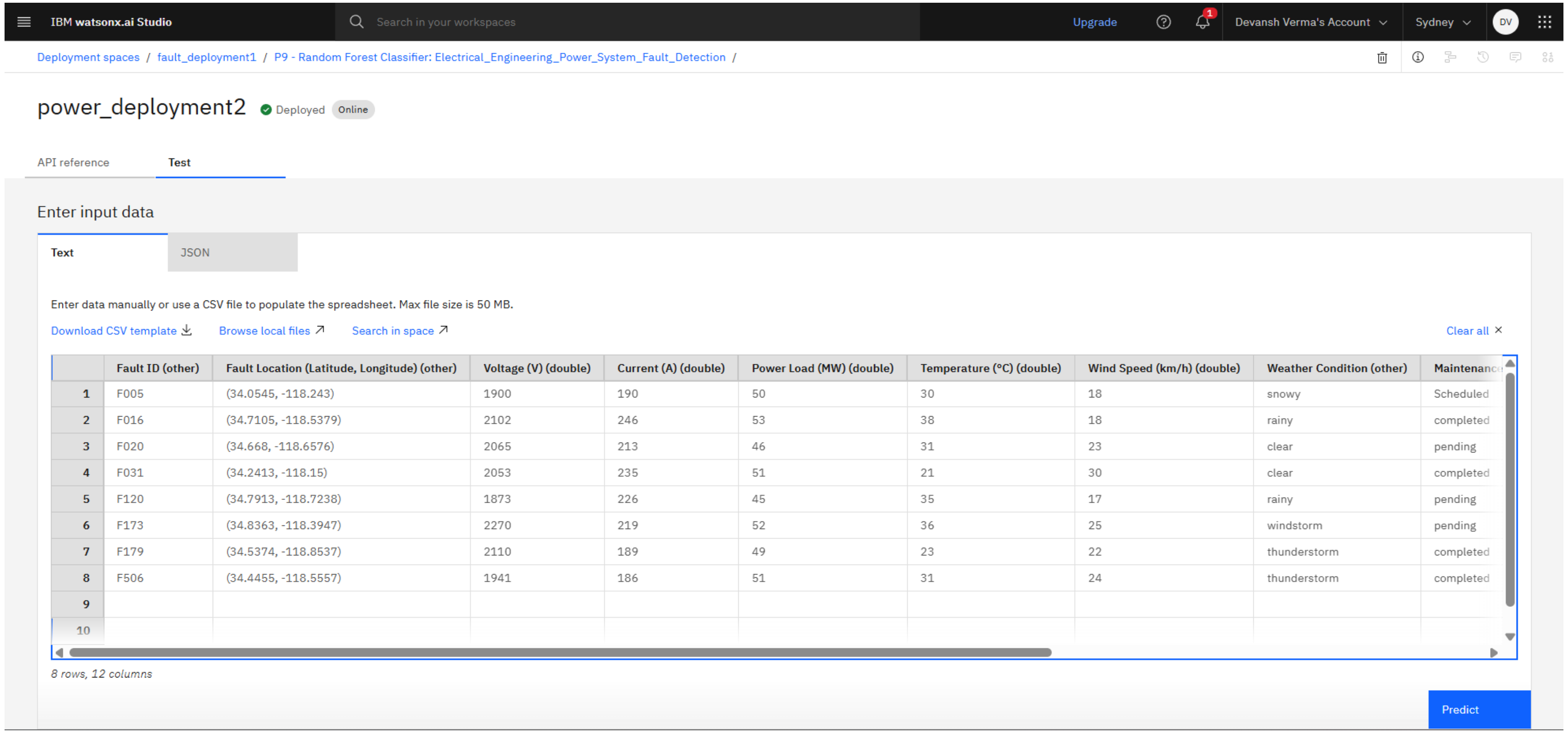
- Selecting the model according to the dataset



- Pipeline , for the best selected model.



- Deployment



- Input values

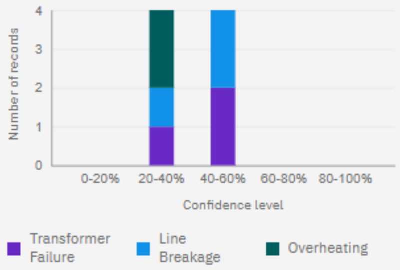
Prediction results

Prediction type

Multiclass classification

Prediction percentage

Confidence level distribution



Display format for prediction results

◉ Table view ○ JSON view

Show input data

	Prediction	Confidence
1	Transformer Failure	38%
2	Line Breakage	38%
3	Line Breakage	42%
4	Transformer Failure	56%
5	Overheating	38%
6	Transformer Failure	42%
7	Overheating	35%
8	Line Breakage	46%
9		
10		
11		
12		
13		
14		
15		
16		

- Prediction results

➢ Model evaluation, precision recall curve, threshold chart, confusion matrix

Pipeline details

Pipeline 9

Rank	Accuracy (Optimized)	Algorithm
1	0.314 (Holdout)	Batched Tree Ensemble Classifier (Random Forest Classifier)

Model viewer

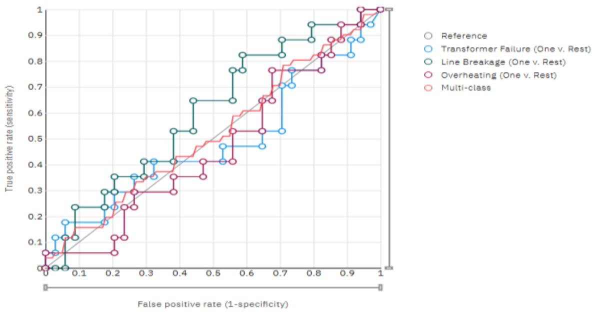
- Model information
- Feature summary

Evaluation

- Model evaluation
- Confusion matrix
- Precision recall
- Threshold

Model evaluation

ROC curve

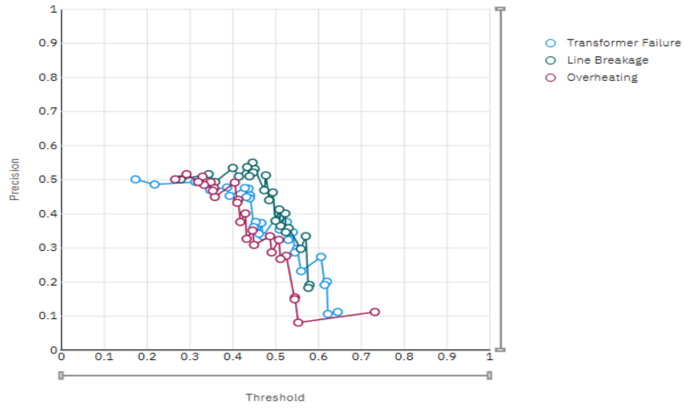


Threshold

View

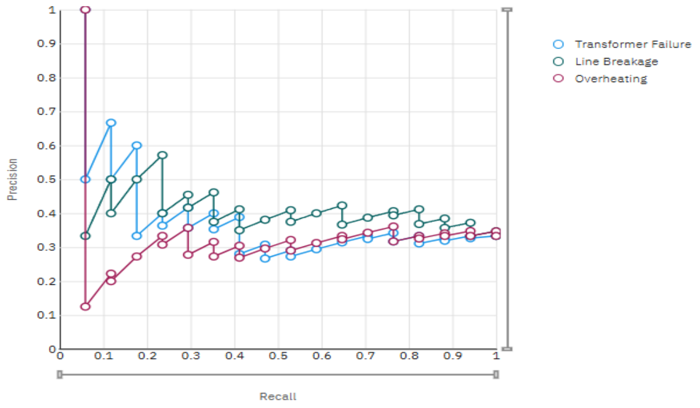
Precision

Threshold chart



Precision recall

Precision recall curve



Confusion matrix

View

Multi-class

Observed	Predicted			
	Line Breakage	Overheating	Transformer Failure	Percent correct
Line Breakage	8	3	6	47.1%
Overheating	9	2	6	11.8%
Transformer Failure	4	7	6	35.3%
Percent correct	38.1%	16.7%	33.3%	31.4%

Less correct — More correct

CONCLUSION

- This project successfully demonstrates the application of automated machine learning on IBM Watsonx.ai and IBM Cloud to address the critical challenge of fault detection and classification in power system networks. Leveraging real-world electrical measurement data and cloud-based AI services, the solution:
- Achieved accurate, multiclass classification of different power system faults (such as transformer failure, line breakage, and overheating) using automated pipeline generation, feature engineering, and model optimization.
- Deployed the best model as a REST API, enabling seamless integration with grid monitoring systems for real-time or batch inference.
- Delivered high-confidence, reliable results—as evidenced by robust accuracy and confidence distribution on test samples—demonstrating readiness for practical, real-world deployment.
- Demonstrated the value of cloud AI automation, minimizing the need for manual intervention, accelerating development, and ensuring model maintainability and scalability.

FUTURE SCOPE

- **Integration with Real-Time Grid Systems**:-Extend the model deployment to directly integrate with live power grid SCADA systems and IoT sensor networks for continuous, real-time fault monitoring and immediate automated response.

- **Incorporation of More Diverse Data Sources**:- Enhance the model by incorporating additional data types such as weather conditions, equipment maintenance logs, and historical fault records to improve prediction accuracy and fault diagnosis depth.

- **Expansion to More Fault Types**:- Train on larger, more diverse datasets to classify a wider variety of fault types (e.g., transient faults, intermittent faults), increasing the model's robustness and usability in complex grid environments.

REFERENCES

- Kaggle Power System Faults Dataset:
  “Power System Faults Dataset,” Kaggle. [Dataset link provided in your original problem statement]
- IBM Documentation:
  IBM Cloud and Watsonx.ai official documentation for AutoAI, Watson Studio, and Watson Machine Learning services.
- Example Projects & Tutorials:
  IBM Watsonx.ai tutorials for AutoAI deployment.
  Github repositories and community projects related to power system fault classification with machine learning.

GITHUB LINK:-

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Getting Started with Artificial Intelligence

IBM SkillsBuild

Rahul Kumar

Has successfully satisfied the requirements for:

Getting Started with Artificial Intelligence

Issued on: Jul 26, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/2d398a7e-7a59-433c-b4cc-f7753c6eda95

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

Rahul Kumar

Has successfully satisfied the requirements for:

Journey to Cloud: Envisioning Your Solution

Issued on: Jul 27, 2025

Issued by: IBM SkillsBuild

Verify: https://www.credly.com/badges/5aa44b0f-63f5-4ff6-ad11-480e0526d445

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Rahul Kumar

for the completion of

Lab: Retrieval Augmented Generation with LangChain

(ALM-COURSE_3824998)

According to the Adobe Learning Manager system of record

Completion date: 27 Jul 2025 (GMT)

Learning hours: 20 mins

THANK YOU