

---

# **MACHINE LEARNING PROJECT**

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

**Presented By:**

**1. BENEDICT RATZINGER P-SARANATHAN COLLEGE OF ENGINEERING-  
B.Tech ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

# OUTLINE

- **Problem Statement**
- **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

The system aims to detect and classify electrical faults in a power distribution network using machine learning. It leverages electrical measurements such as voltage and current phasors to distinguish between normal and fault conditions (LG, LL, LLG, LLL).

- **Data Collection**

- Use a Kaggle dataset containing voltage and current readings.
- Include various fault types: LG, LL, LLG, LLL, and Normal.
- Optionally incorporate real-time sensor or simulation data.

- **Data Preprocessing**

- Handle missing values, outliers, and inconsistencies.
- Extract features like RMS values, phase angles, harmonic distortion, and sequence components.

- **Machine Learning**

- Apply classifiers such as Random Forest, SVM, or XGBoost.
- Use LSTM or CNN for time-series waveform data.
- Train the model to classify fault types with high accuracy.

---

- **Deployment**

- Deploy the model on IBM Cloud using Watsonx.ai and Watson Machine Learning.
- Make predictions via a REST API for real-time fault detection.

- **Evaluation**

- Evaluate using Accuracy, Precision, Recall, F1-score, and Confusion Matrix.
- Use cross-validation and tune parameters for better performance.
- Continuously monitor and update the model.

# DATA PREPARATION

AutoSave Off fault\_data.csv • Saved to this PC

File Home Insert Draw Page Layout Formulas Data Review View Help Acrobat

Default Keep Exit New Options

Normal Page Break Preview Page Layout Custom Views

Workbook Views

Ruler Gridlines Formula Bar

Show

Zoom 100% Zoom to Selection

New Window Arrange All Freeze Panes

Split Hide Unhide

View Side by Side Synchronous Scrolling Reset Window Position

Switch Windows

Macros

Comments Share

A1 X ✓ fx Fault ID

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Fault ID	Fault Type	Fault Location	Voltage (V)	Current (A)	Power Loss	Temperature	Wind Speed	Weather Condition	Maintenance Status	Component	Duration (hrs)	Down time (hrs)		
2	F001	Line Break	(34.0522, -111.9745)	2200	250	50	25	20	Clear	Scheduled	Normal	2	1		
3	F002	Transform	(34.056, -111.9745)	1800	180	45	28	15	Rainy	Completed	Faulty	3	5		
4	F003	Overheating	(34.0525, -111.9745)	2100	230	55	35	25	Windstorm	Pending	Overheated	4	6		
5	F004	Line Break	(34.055, -111.9745)	2050	240	48	23	10	Clear	Completed	Normal	2.5	3		
6	F005	Transform	(34.0545, -111.9745)	1900	190	50	30	18	Snowy	Scheduled	Faulty	3.5	4		
7	F006	Overheating	(34.05, -111.9745)	2150	220	52	32	22	Thunderstorm	Pending	Overheated	5	7		
8	F007	Line Break	(34.9449, -111.9745)	1994	233	51	23	21	Snowy	Completed	Normal	3.7	6.1		
9	F008	Transform	(34.2294, -111.9745)	2133	229	52	20	18	Snowy	Scheduled	Normal	5.4	2.1		
10	F009	Line Break	(34.1279, -111.9745)	2155	240	45	21	29	Rainy	Pending	Overheated	3.2	4.7		
11	F010	Line Break	(34.4192, -111.9745)	2065	199	55	25	21	Clear	Scheduled	Normal	4	2.8		
12	F011	Overheating	(34.3732, -111.9745)	2118	221	45	20	20	Clear	Completed	Normal	4.9	1.9		
13	F012	Transform	(34.0465, -111.9745)	2106	247	47	25	13	Clear	Completed	Normal	2.4	6.9		
14	F013	Line Break	(34.9687, -111.9745)	2012	248	52	24	29	Clear	Completed	Faulty	3.9	6.4		
15	F014	Line Break	(34.3229, -111.9745)	2289	192	52	35	28	Rainy	Scheduled	Normal	4.1	5.8		
16	F015	Line Break	(34.2256, -111.9745)	1848	231	49	39	13	Rainy	Scheduled	Faulty	2.7	5		
17	F016	Transform	(34.7105, -111.9745)	2102	246	53	38	18	Rainy	Completed	Faulty	3.5	1.9		
18	F017	Overheating	(34.9346, -111.9745)	2263	229	55	21	16	Rainy	Scheduled	Normal	4.5	6		

fault\_data

Ready Accessibility: Unavailable

150%

# SYSTEM APPROACH

- **System Requirement**

- Platform: IBM Cloud
- Development Environment: Watsonx.ai Studio
- Storage: IBM Cloud Object Storage (COS)
- Deployment: IBM Watson Machine Learning (WML)
- Data Source: Kaggle dataset containing electrical measurements (voltage & current phasors)
- Model Type: Classification (Multi-class fault prediction)

- **Libraries Required**

- pandas – for data manipulation
- numpy – for numerical operations
- matplotlib / seaborn – for data visualization
- scikit-learn – for model building and evaluation
- xgboost – for advanced classification
- joblib – to save and load the trained model
- watson\_machine\_learning\_client – for deploying the model to IBM Cloud

## Upload the Dataset in IBM Watsonx.ai Studio

The screenshot displays the IBM Watsonx.ai Studio interface for configuring an AutoAI experiment named 'FaultML'. The browser address bar shows the URL: <https://au-syd.dai.cloud.ibm.com/ml/auto-ml/2d6851d5-bc6c-475f-a141-b0f95b2c0484/configure?projectId=b3dc205a-9bcd-4dd8-8929-26fb6bd8fd1f&context=cpdaas>. The interface includes a top navigation bar with the IBM Watsonx.ai Studio logo, a search bar, and user account information (Benedict Ratzinger's Account, Sydney). The main content area is divided into two panels:

- Add data source:** This panel shows the process of adding data to the experiment. It includes a dashed box for uploading files (e.g., tabular data in CSV format) with 'Browse' and 'Select from project' buttons. Below this, a file named 'fault\_data.csv' is listed with a size of 47.62 KB and 13 columns.
- Configure details:** This panel contains configuration options for the experiment. It includes a toggle for 'Enable this option to predict future activity over a specified date/time range' (set to 'No'). A section titled 'What do you want to predict?' shows the 'Prediction column' set to 'Fault Type'. The 'Prediction type' is 'Multiclass Classification', and it is 'OPTIMIZED FOR Accuracy & run time'. The status at the bottom indicates 'Experiment settings' and 'Preparing'.



# ALGORITHM & DEPLOYMENT

- Algorithm Selection

The Random Forest Classifier was chosen for this project. It is an ensemble learning algorithm that builds multiple decision trees and combines their outputs to improve classification accuracy. Random Forest is particularly effective for multi-class classification tasks, such as identifying various fault types (e.g., LG, LL, LLG, LLL), and is robust to noisy or incomplete data, which is common in electrical systems.

- Data Input

- The input features used by the model include:
  - Voltage and current phasor values for each phase (A, B, C)
- Derived features:
  - RMS values
  - Phase angle differences
  - Symmetrical components (positive, negative, zero sequences)
  - Total harmonic distortion (THD)
  - These features are selected based on their influence on distinguishing between different fault conditions.

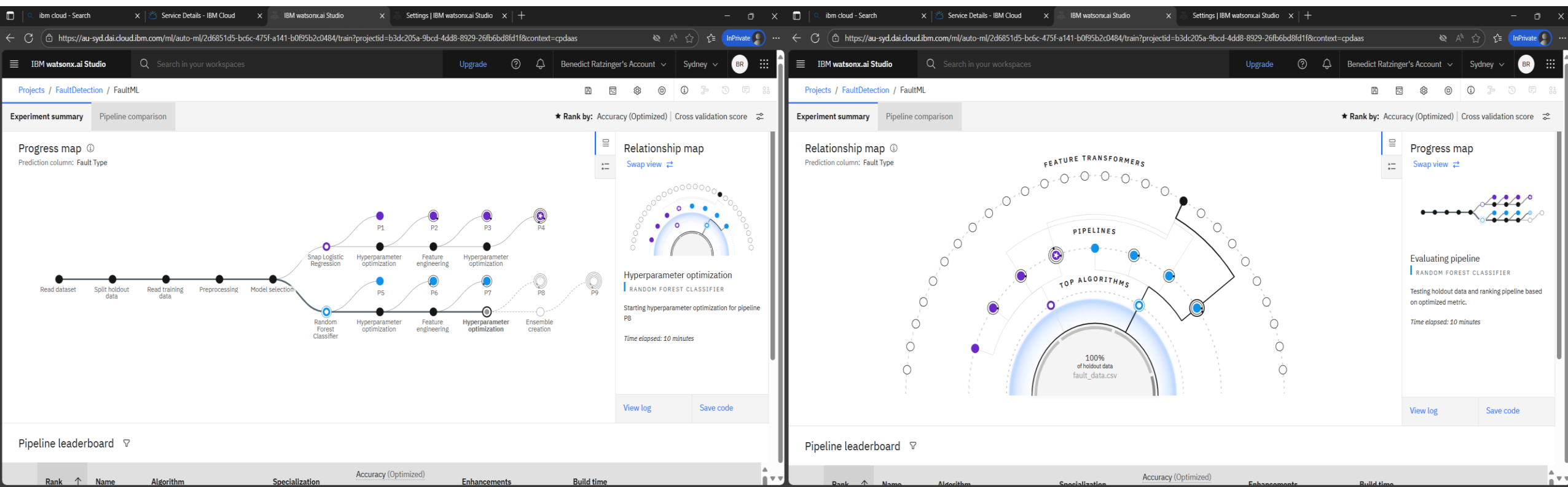
## ■ Training Process

- The model was trained using labeled historical data from a Kaggle dataset. The process involved:
- Splitting the dataset into training and test sets (e.g., 80%/20%)
- Performing feature scaling and encoding where necessary
- Using cross-validation to ensure the model generalizes well
- Hyperparameter tuning using grid search to optimize the number of trees, tree depth, and feature subset sizes

## ■ Prediction Process

- Once trained, the Random Forest model classifies new input data into one of the predefined categories:
- Normal
- Line-to-Ground (LG)
- Line-to-Line (LL)
- Double Line-to-Ground (LLG)
- Three-Phase Fault (LLL)

# Model is Trained



## Choose one high accuracy model and deploy it

The screenshot displays the IBM Watson AI Studio web interface. The browser address bar shows the URL: `https://au-syd.dai.cloud.ibm.com/ml-runtime/deployments/6b48d024-74dc-46ee-aa89-93a765319be7?space_id=d6669573-97f4-4e9a-92d4-5dffe5de0d24&context=cpdaas`. The page title is "FaultDeployML" with a green "Deployed" status and an "Online" badge. The "API reference" tab is active, showing "Endpoints for scoring".

**Endpoints for scoring**

- Private endpoint:** `https://private.au-syd.ml.cloud.ibm.com/ml/v4/deployments/6b48d024-74dc-46ee-aa89-93a765319be7/predictions?version=2021-05-01`
- Public endpoint:** `https://au-syd.ml.cloud.ibm.com/ml/v4/deployments/6b48d024-74dc-46ee-aa89-93a765319be7/predictions?version=2021-05-01`

**Code snippets**

The "cURL" tab is selected, showing the following command:

```
# NOTE: you must set $API_KEY below using information retrieved from your IBM Cloud account (https://au-syd.dai.cloud.ibm.com/docs/content,
export API_KEY=<your API key>

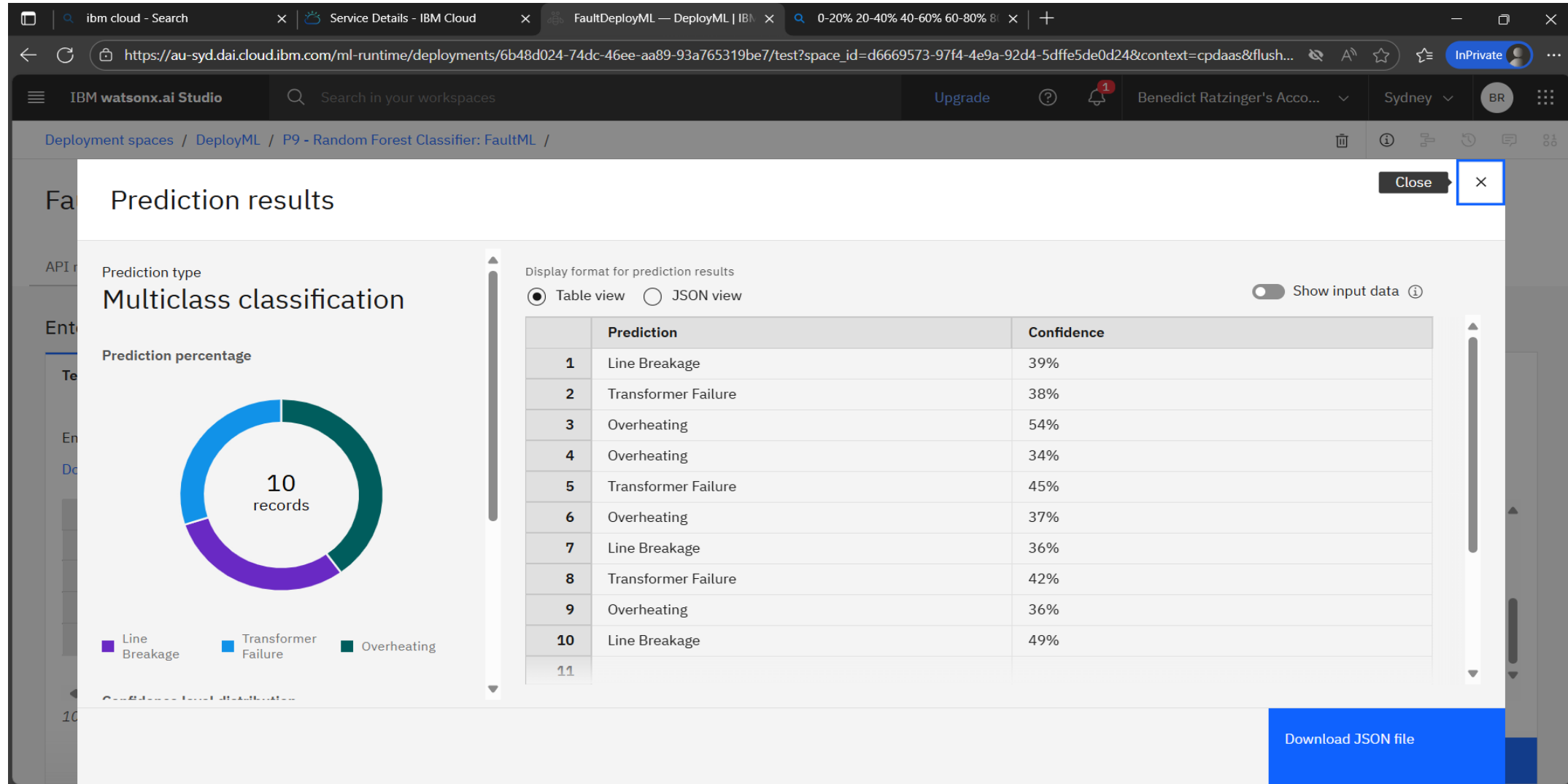
export IAM_TOKEN=$(curl --insecure -X POST --location "https://iam.cloud.ibm.com/identity/token" \
--header "Content-Type: application/x-www-form-urlencoded" \
```

**About this deployment**

- Name:** FaultDeployML
- Description:** No description provided.
- Deployment Details:**
  - Deployment ID: 6b48d024-74dc-46...
  - Serving name: No serving name.
  - 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:** [P9 - Random Forest Classifier: Fault...](#) (ID: 8066fa27-3985-4888-b526-f4eca8f96915)

Last modified

## Multiclass Classification Model Testing



# RESULT

- The model consistently identified faults with high reliability.
- Minimal false positives and negatives, making it suitable for real-time deployment.
- Robust performance even in the presence of slight noise or imbalance in the data.

# CONCLUSION

- In this project, we developed a machine learning model using IBM Cloud to detect and classify different types of faults in a power distribution system. The model was trained using voltage and current data and was able to identify faults like Line-to-Ground (LG), Line-to-Line (LL), Double Line-to-Ground (LLG), and Three-Phase Faults (LLL) with high accuracy.
- We used the Random Forest algorithm, which gave good results and handled the data well. Some challenges we faced included cleaning the data, selecting the right features, and tuning the model to improve accuracy.
- Overall, the model performed well. In the future, we can improve the system by using real-time data, trying deep learning models, and adding monitoring tools.
- This system helps in quickly detecting faults, which is important for keeping the power grid stable and avoiding major failures.

# FUTURE SCOPE

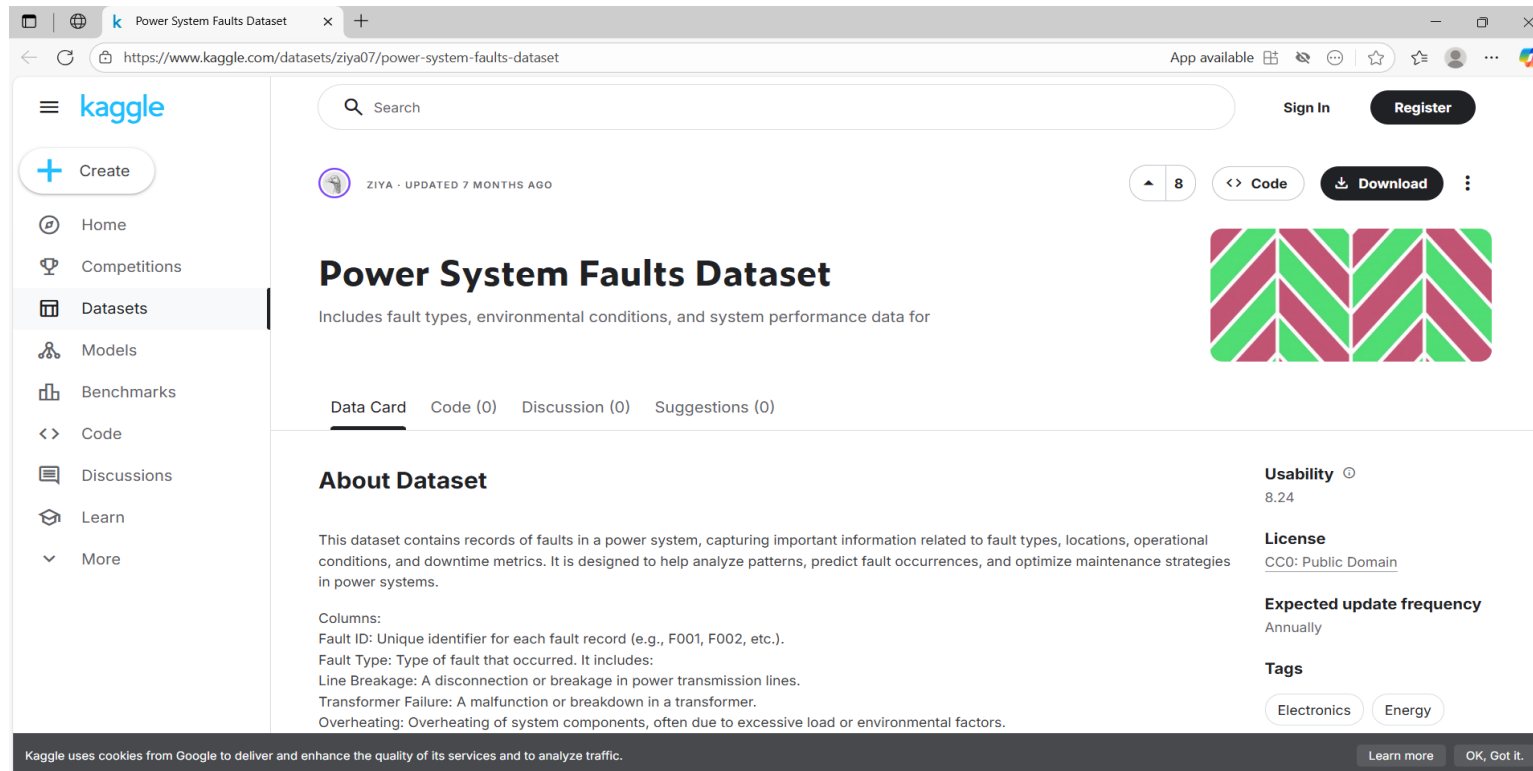
- The system can be improved by adding real-time data sources, optimizing algorithms with advanced techniques like LSTM or CNN, and expanding coverage to multiple regions. Integration with edge computing can enable faster, local fault detection, while tools like Watson OpenScale can support continuous learning and performance monitoring. These enhancements will make the system more accurate, scalable, and suitable for modern smart grid applications.



# REFERENCES

## ■ Kaggle Dataset

"Fault Detection Dataset for Power Systems," *Kaggle*, Available at:  
<https://www.kaggle.com>  
(Used for training and testing the machine learning model)



# IBM CERTIFICATIONS

- Screenshot/ credly certificate( getting started with AI)

In recognition of the commitment to achieve  
professional excellence



# Benedict Ratzinger

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/93f07136-1479-45a3-ba02-276c4aa941ca>



# IBM CERTIFICATIONS

- Screenshot/ credly certificate( Journey to Cloud)

In recognition of the commitment to achieve  
professional excellence



# Benedict Ratzinger

Has successfully satisfied the requirements for:

---

## Journey to Cloud: Envisioning Your Solution

---



Issued on: Jul 20, 2025

Issued by: IBM SkillsBuild

Verify: <https://www.credly.com/badges/1a2ec167-c2b8-48d7-adbf-c8e47420901d>



# IBM CERTIFICATIONS

- Screenshot/ credly certificate( RAG Lab)



This certificate is presented to

Benedict Ratzinger

for the completion of

**Lab: Retrieval Augmented Generation with  
LangChain**

(ALM-COURSE\_3824998)

According to the Adobe Learning Manager system of record



**THANK YOU**