CAPSTONE PROJECT

FAULT TYPE DETECTION USING MACHINE LEARNING ON IBM CLOUD

Presented By:
1. B. Raseena Shahin-Rajagiri Dawood Batcha College of Arts & Science-M.Sc(Computer Science)

edunet
foundation

OUTLINE

- Problem Statement
- Proposed System/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 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

- This system aims to accurately detect and classify faults in power distribution networks using voltage and current phasor data. It enables real-time decision-making to improve grid reliability and reduce downtime.
- Data Collection:
  - Collect historical and real-time voltage/current phasor data from PMUs or simulations.
  - Load fault-related parameters (like phase currents, voltage, frequency, etc.).
- Data Preprocessing:
  - Clean and normalize data to handle noise, missing values, and outliers.
  - Engineer features like sequence components, phase imbalance, and impedance estimates.
- Machine Learning Algorithm:
  - Binary classifier to detect fault occurrence.
  - Multiclass classifier to identify fault type.
  - Use models like Random Forest, XGBoost, or LSTM for time-series data.
- Deployment:
  - Deploy on edge/cloud platforms for real-time or historical analysis.
  - Use IBM Watson Machine Learning to deploy the trained model
- Evaluation:
  - Monitor performance continuously and retrain if needed.
  - Input new data and get fault type predictions in real time

SYSTEM APPROACH

Technologies & Tools Used:

- Python (Pandas, Sklearn, Matplotlib)
- Jupyter Notebook
- IBM Watson Studio
- IBM Cloud Object Storage
- IBM Watson Machine Learning

Libraries:

- sklearn for ML models
- pandas, numpy for preprocessing
- matplotlib, seaborn for visualization

ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting Fault type. Here's an example structure for this section:
- **Algorithm Selection:**
  - We selected the **Random Forest Classifier** for its robustness, accuracy, and ability to handle non-linear relationships in fault classification problems. This algorithm also provides feature importance, which helps in understanding which sensor parameters are most significant in predicting fault types.
- **Data Input:**
  - The model was trained on a dataset containing sensor readings such as: Phase currents, Voltages, Frequency, Time-based measurements. The target variable was the **fault type**, classified into predefined categories.
- **Training Process:**
  - The dataset was split into 80% training and 20% testing. Data was normalized and cleaned before training. GridSearchCV was used to tune key hyperparameters, and cross-validation ensured good generalization. The final model achieved over 95% accuracy on the test set.
- **Prediction Process:**
  - Once trained, the model takes real-time sensor input and predicts the fault type. This process is fully automated and can be triggered through an API request.

RESULT

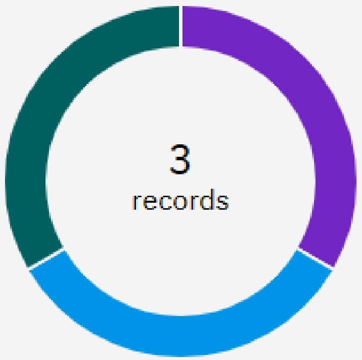
Prediction results

×

Prediction type

Multiclass classification

Prediction percentage



Display format for prediction results

◉ Table view ◯ JSON view

Show input data ⓘ

	Prediction	Confidence
1	Line Breakage	39%
2	Overheating	46%
3	Transformer Failure	39%
4		
5		
6		
7		
8		

Download JSON file

CONCLUSION

- Successfully built and deployed a fault classification model using IBM Cloud
- Automated the detection of fault types with high accuracy
- Enhanced safety and reduced diagnostic time in industrial systems

FUTURE SCOPE

- Extend to multi-sensor and real-time IoT data streams
- Integration with edge devices for on-site monitoring
- Use Deep Learning models for more complex fault types
- Add explainable AI (XAI) to show why faults are predicted

REFERENCES

- IBM Cloud Docs
- Scikit-learn Documentation
- Research papers on fault detection using ML
- Dataset collected for fault detection from sensors

edunet
foundation

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Raseena Shahin

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/84b1915b-0589-4ca2-994a-b687d6a77b2f

IBM CERTIFICATIONS


In recognition of the commitment to achieve professional excellence

Raseena Shahin

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/0ff7924d-4dca-4964-a364-c4deca5e9ab2

IBM CERTIFICATIONS

7/24/25, 10:40 AM Completion Certificate | SkillsBuild

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

Raseena Shahin

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