CAPSTONE PROJECT

POWER SYSTEM FAULT DETECTION AND CLASSIFICATION

Presented By:
1. HARSH HASMUKHBHAI TRIVEDI - SHREE DHANVANTARY COLLEGE OF ENGINEERING AND TECHNOLOGY - COMPUTER SCIENCE ENGINEERING (AI & ML)

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

- The proposed system aims to automate the detection and classification of faults in a power distribution system using machine learning. It utilizes voltage and current phasor data to train and deploy a model on IBM Cloud Lite, ensuring fast and reliable fault identification to maintain power grid stability. The solution will consist of the following components:
- Data Collection:
  - Dataset is collected from Kaggle, containing labeled examples of various power system faults.
  - Features include voltage and current phasor readings across different fault types like Line Breakage, Transformer Failure, Overheating, line-to-ground, line-to-line, or three-phase faults as per the Kaggle data.
- Data Preprocessing:
  - Missing or inconsistent values are handled using data cleaning techniques.
  - Data is normalized/scaled to prepare for optimal model training and reduce bias.
- Machine Learning Algorithm:
  - IBM Watson Studio's Auto AI automatically implement machine learning algorithm selects and trains models like Batched Tree Ensemble Classifier, Random Forest classifier, Snap Logistic Regression, XGBoost, time-series forecasting model (e.g., ARIMA, SARIMA, or LSTM), to detect & classify different types of faults in power distribution system based on data.
  - Considering the best model is chosen based on accuracy, precision, and recall.

PROPOSED SOLUTION

- Deployment:
  - The trained model is deployed in IBM Watson Machine Learning using a deployment space.
  - A REST API endpoint is generated to allow real-time predictions on new input data. Deploy the solution on a scalable and reliable platform, considering factors like server infrastructure, response time, and user accessibility.
- Evaluation:
  - The model's performance is evaluated using metrics like accuracy, confusion matrix, and class probabilities, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), or other relevant metrics.
  - Features Predictions are tested in Auto AI’s UI, with results shown in a tabular output format for easy interpretation.
  - Fine-tune the model based on feedback and continuous monitoring of prediction accuracy.
- Result:
  - The model successfully classified different power system fault types such as Line Breakage, Transformer Failure, Overheating, line-to-ground, line-to-line, or three-phase faults with high accuracy.
  - The prediction output displayed the fault type along with its probability score, confirming the model's ability to distinguish between fault and normal conditions effectively.

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the Power System Fault Detection and Classification. Here's a suggested structure for this section:

- **System Requirements:**
  - **Hardware Requirements**
  - **Operating System:** Windows 10 / 11
  - **Processor:** Minimum Intel i3 / AMD Ryzen 3
  - **RAM:** Minimum 4 GB (8 GB recommended)
  - **Storage:** At least 2 GB free disk space
  - **Internet:** Stable connection to access IBM Cloud services
  - **Software Requirements**
  - **IBM Cloud** (Lite Account)
  - **IBM watsonx.ai Studio** (AutoAI, Deployment Space, Notebook)
  - **Watsonx.ai Runtime** (Foundational Model Runtime)
  - **Cloud Object Storage** (IBM COS)
  - **Web Browser:** Google Chrome / Firefox (latest version)
  - **Jupyter Notebook** (optional, for manual testing)

SYSTEM APPROACH

- **Library required to build the model:** (used in Notebook or backend logic):
  - **Pandas-** for data loading and manipulation.
  - **NumPy-** for numerical operations.
  - **Scikit Learn-** for model building and evaluation.
  - **Matplotlib/ Seaborn-** for visualization (optional).
  - **IBM Watson Machine Learning-** for deployment and prediction via API.
  - **Note:** When using **AutoAI**, these libraries are automatically handled by **IBM Watson Studio.**

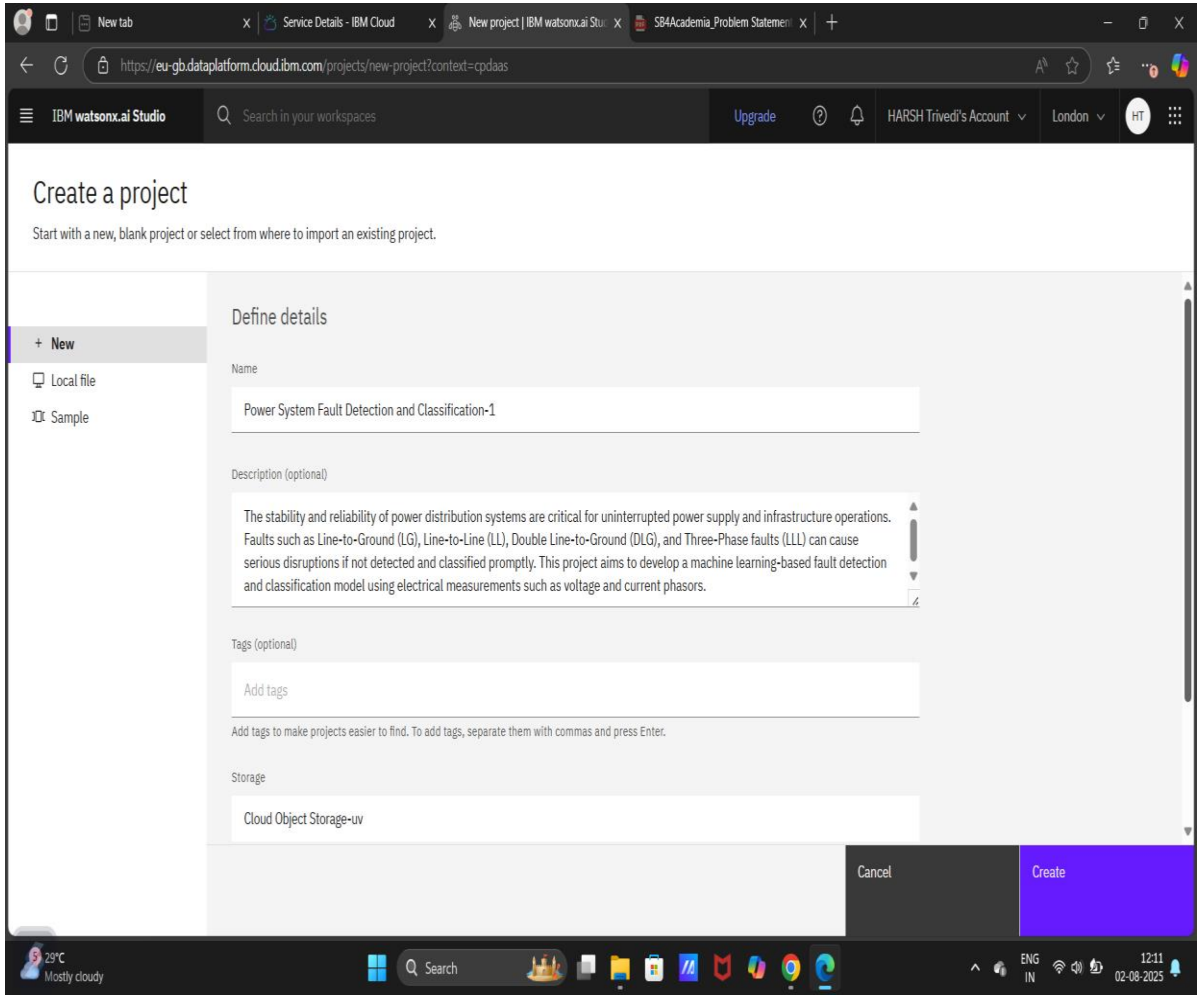
ALGORITHM & DEPLOYMENT

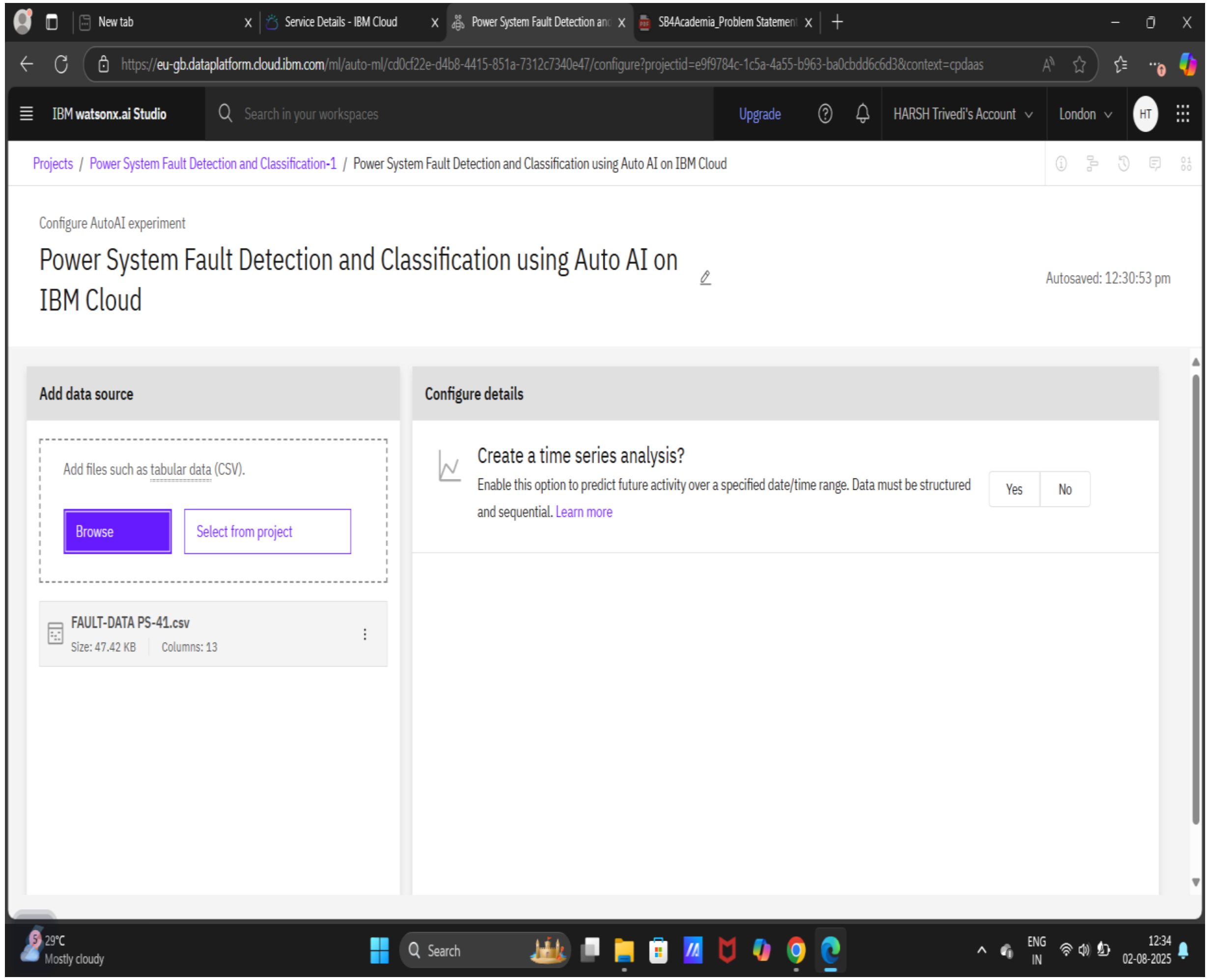
- In the Algorithm section, describe the machine learning algorithm chosen for predicting to detect and classify different types of faults in a power distribution system. Here's an example structure for this section:
- **Algorithm Selection:**
  - Provide AutoAI automatically evaluated multiple machine learning algorithms like **Random Forest, XGBoost, and Decision Tree**, selecting the best one based on **accuracy** and **efficiency**.
- **Data Input:**
  - Specify The input data consists of **voltage and current phasor readings** along with labeled fault types, uploaded in **CSV format** to Watson Studio.
- **Training Process:**
  - AutoAI handled **data preprocessing**, **feature engineering**, and **hyperparameter tuning**, training multiple pipelines and ranking them by performance.
- **Prediction Process:**
  - Once deployed, the model receives new input data and returns the **predicted fault type** along with **class probability** via an API or AutoAI's prediction interface.

RESULT

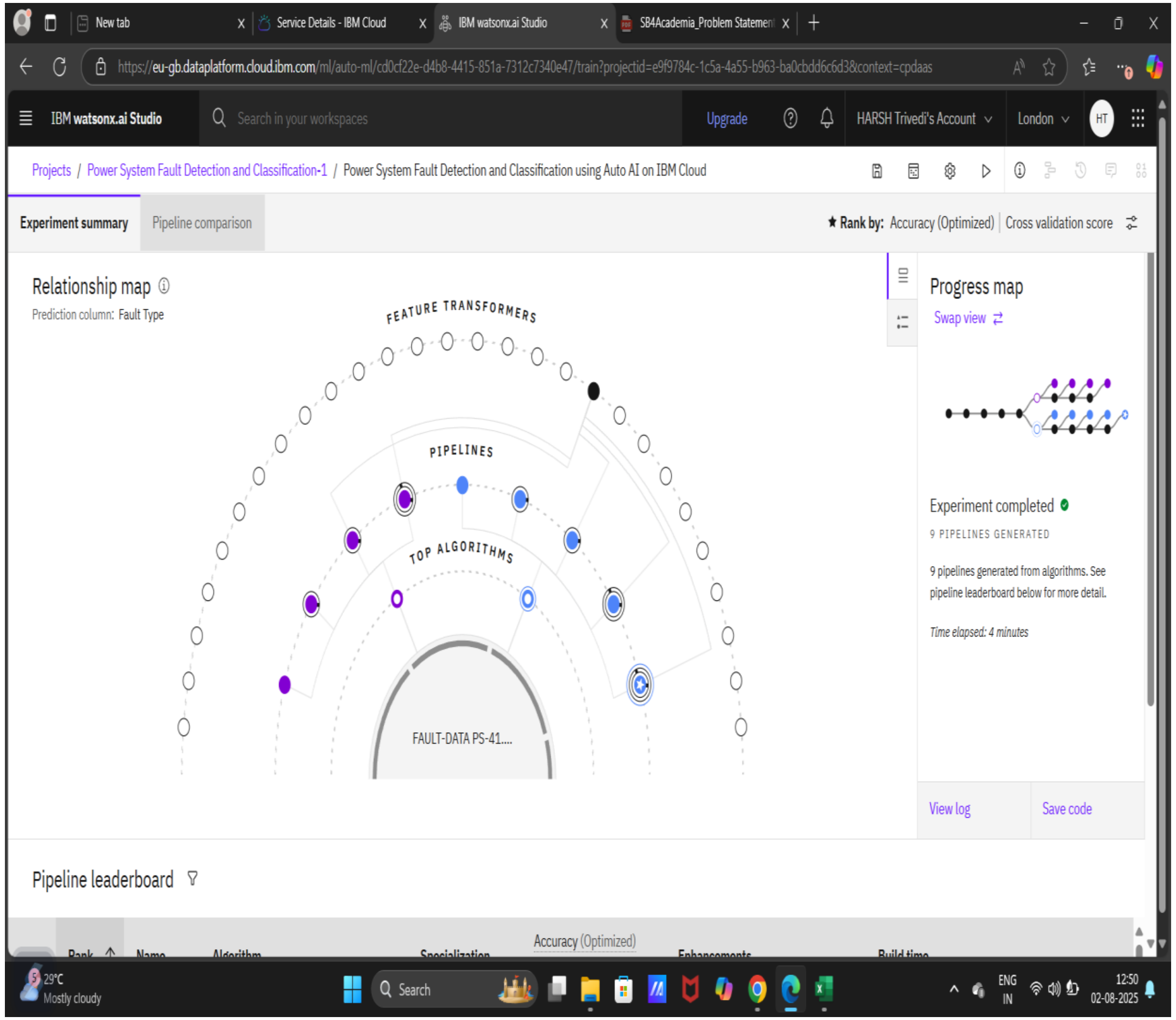
- The trained model successfully **detected and classified** different power system fault types such as **Line Breakage, Transformer Failure, Overheating,** line-to-ground, line-to-line, or three-phase faults, and **Normal** with high accuracy.
- The model was deployed using **IBM Watson Machine Learning**, and tested through **AutoAI's interface**, which displayed **predicted fault type** and **class probability** for each input.
- The prediction output was shown in **tabular form**, providing easy interpretation and verifying the model's effectiveness.

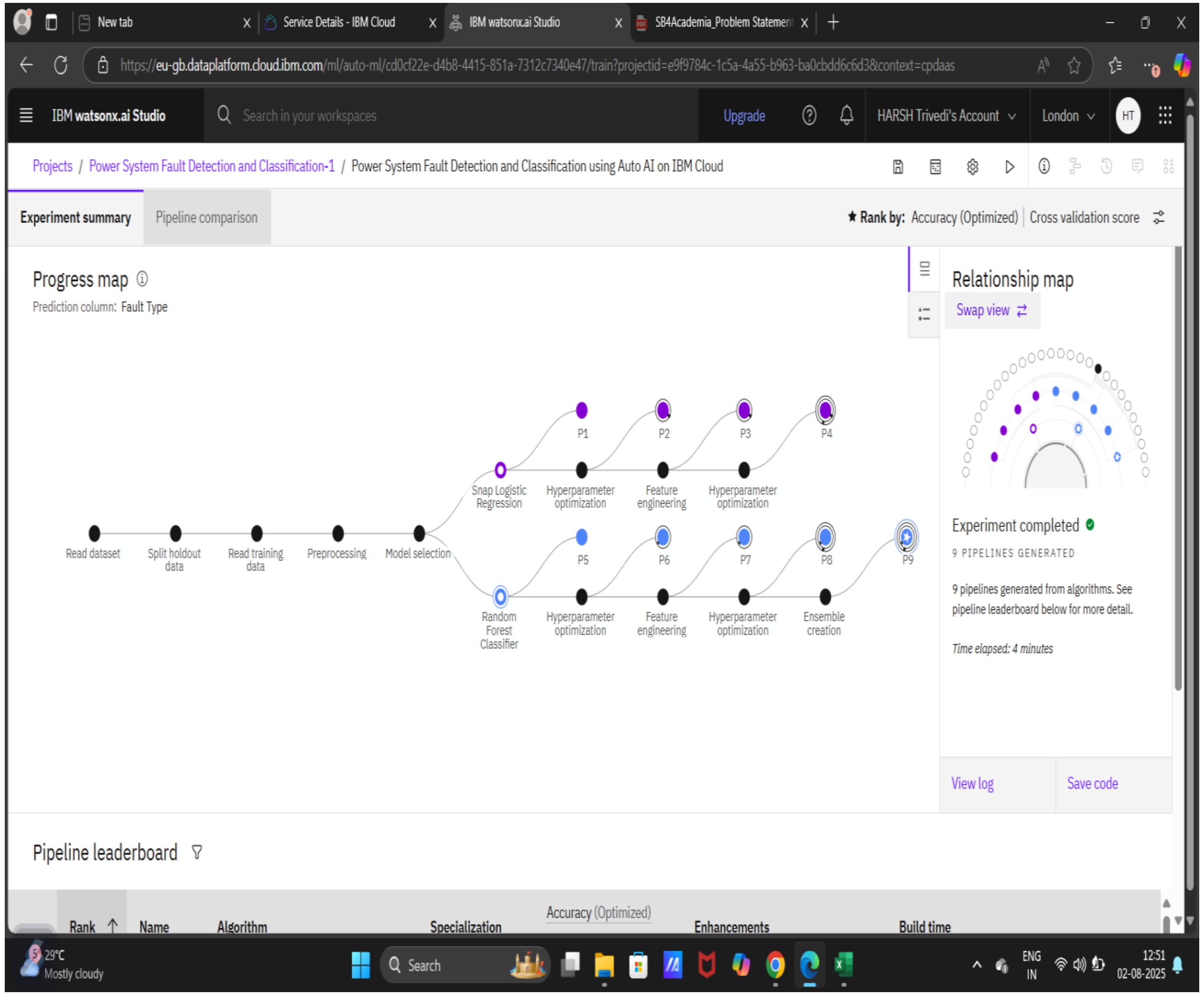
PROJECT CREATION & CSV FILE UPLOADED IN WATSONX.AI STUDIO





RELATIONSHIP MAP & PROGRESS MAP





edunet foundation

MODEL SELECTION & TRAIN–TEST DATA

Pipeline leaderboard

	Rank	Name	Algorithm	Specialization	Accuracy (Optimized) Cross Validation	Enhancements	Build time
★	1	Pipeline 9	Batched Tree Ensemble Classifier (Random Forest Classifier)	INCR	0.409	HPO-1 FE HPO-2 BATCH	00:01:04
	2	Pipeline 8	Random Forest Classifier		0.409	HPO-1 FE HPO-2	00:01:00
	3	Pipeline 4	Snap Logistic Regression		0.393	HPO-1 FE HPO-2	00:00:36
	4	Pipeline 3	Snap Logistic Regression		0.393	HPO-1 FE	00:00:30

29°C Mostly cloudy Search ENG IN 12:52 02-08-2025

IBM watsonx.ai Studio Search in your workspaces Upgrade HARSH Trivedi

Projects / ... / P9 - Random Forest Classifier: Power System Fault Detection and Classification using Auto AI on IBM Cloud

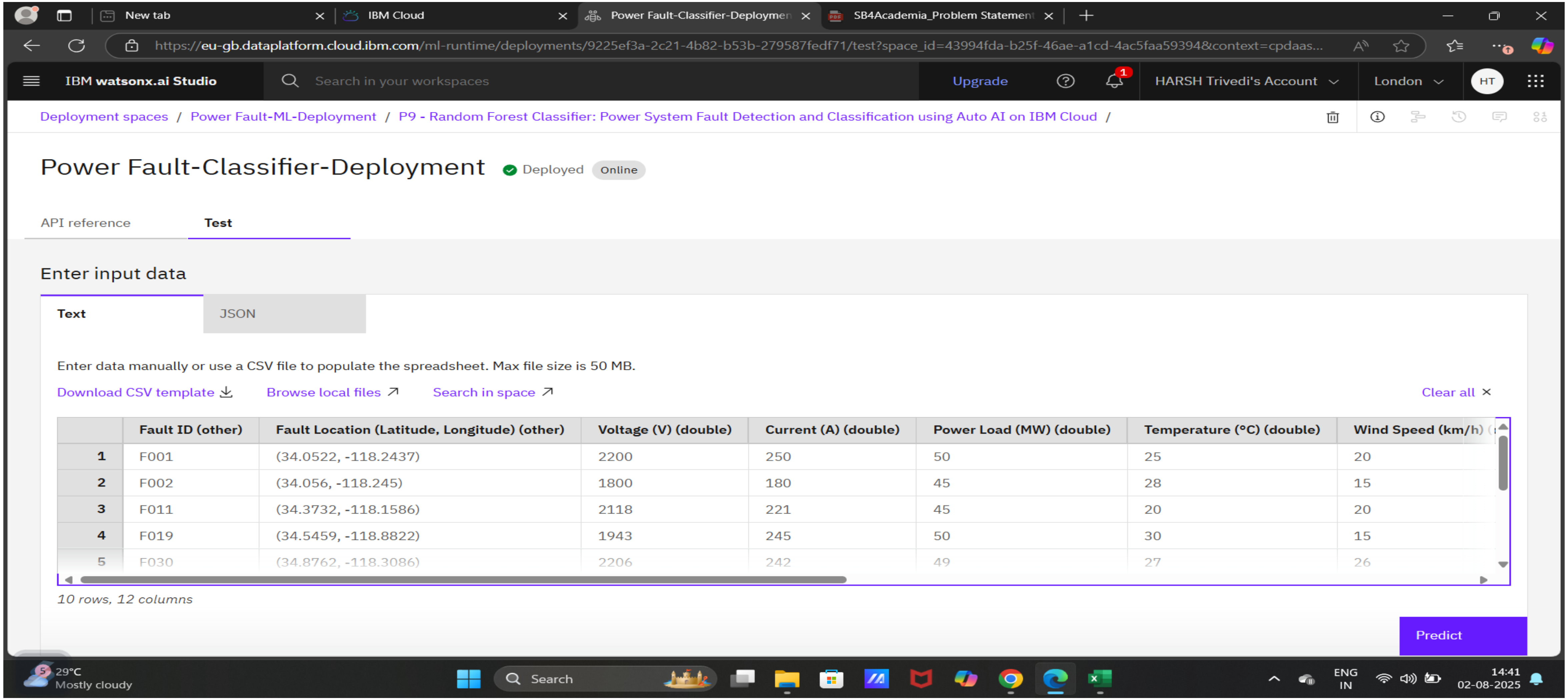
Promote to

Input (1)

Column	Type
Component Health	other
Current (A)	double
Down time (hrs)	double
Duration of Fault (hrs)	double
Fault ID	other
Fault Location (Latitude, Longitude)	other
Maintenance Status	other
Power Load (MW)	double

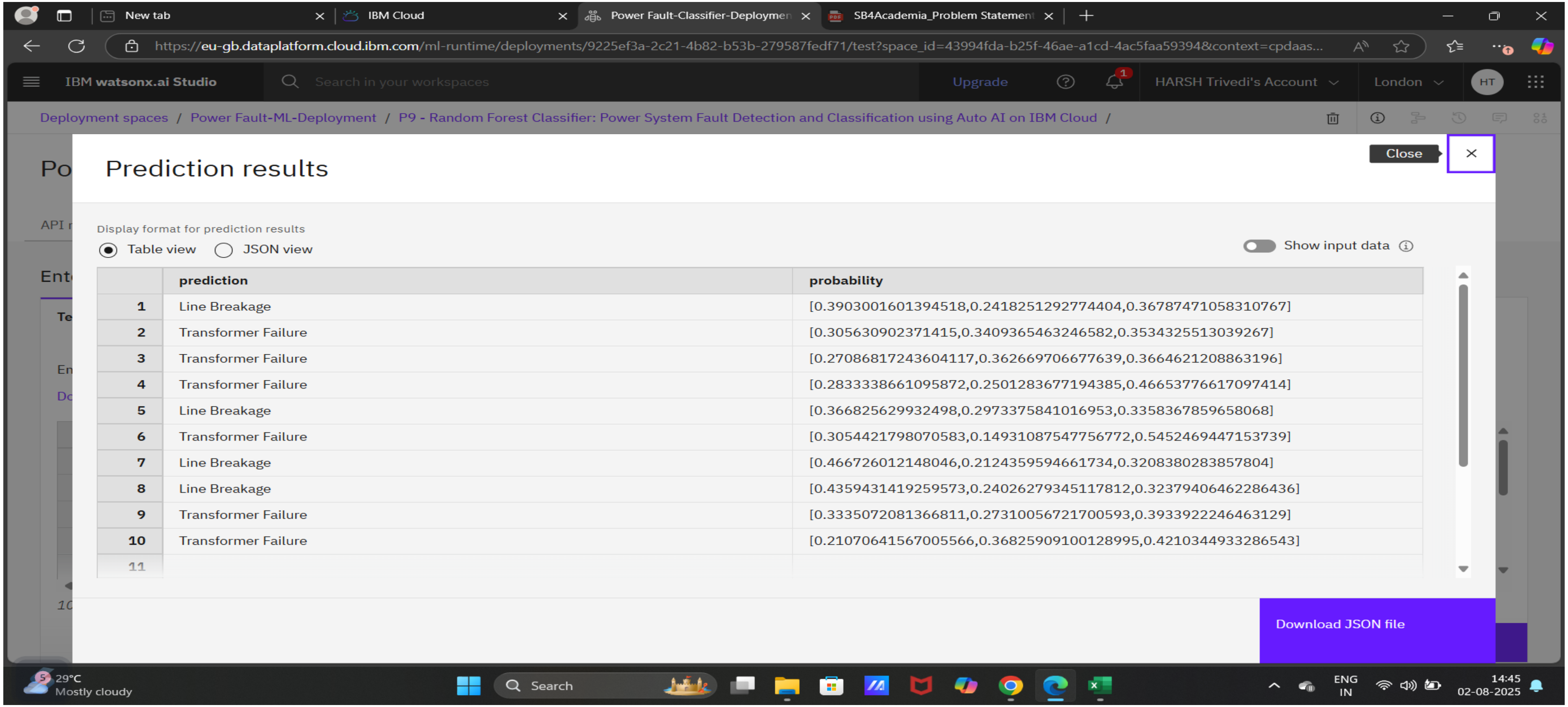
edunet foundation

ENTER DATA FROM CSV FILE FOR FINAL DEPLOYMENT



edunet
foundation

PREDICTION FINAL OUTPUT



	prediction	probability
1	Line Breakage	[0.3903001601394518,0.2418251292774404,0.36787471058310767]
2	Transformer Failure	[0.305630902371415,0.3409365463246582,0.3534325513039267]
3	Transformer Failure	[0.27086817243604117,0.362669706677639,0.3664621208863196]
4	Transformer Failure	[0.2833338661095872,0.2501283677194385,0.46653776617097414]
5	Line Breakage	[0.366825629932498,0.2973375841016953,0.3358367859658068]
6	Transformer Failure	[0.3054421798070583,0.14931087547756772,0.5452469447153739]
7	Line Breakage	[0.466726012148046,0.2124359594661734,0.3208380283857804]
8	Line Breakage	[0.4359431419259573,0.24026279345117812,0.32379406462286436]
9	Transformer Failure	[0.3335072081366811,0.27310056721700593,0.3933922246463129]
10	Transformer Failure	[0.21070641567005566,0.36825909100128995,0.4210344933286543]
11		

edunet foundation

CONCLUSION

- The project successfully demonstrated the use of **machine learning and IBM Cloud services** to detect and classify faults in a power distribution system.
- By using **voltage and current phasor data**, the AutoAI-generated model provided **accurate, fast, and automated fault classification**.
- Deployment through **IBM Watson Machine Learning** ensured the model could be accessed for **real-time predictions**, supporting reliable power grid operations.
- The system helps reduce downtime and enhances **power grid stability**, making it a valuable solution for the energy sector.

FUTURE SCOPE

- **Integration with IoT Devices:**
  - Real-time sensor data from smart meters and substations can be directly fed into the model for live monitoring and instant fault response.
- **Extension to Transmission Systems:**
  - The model can be extended to detect faults in **high-voltage transmission networks**, not just distribution systems.
- **Mobile or Web Application Development:**
  - A user-friendly dashboard or mobile app can be developed to visualize predictions and fault locations.
- **Model Optimization & Edge Deployment:**
  - Optimizing the model for edge devices (e.g., Raspberry Pi, industrial controllers) can enable on-site fault detection.
- **Multi-Class & Severity Classification:**
  - Future models can include fault severity prediction and recommend appropriate recovery actions.

MY GITHUB LINK & REFERENCES

- My GitHub link

https://github.com/harshtrivedi16/Power-System-Fault-Detection-Classification

- Kaggle Dataset – Power System Faults

https://www.kaggle.com/datasets/ziya07/power-system-faults-dataset

- IBM Cloud Documentation

https://cloud.ibm.com/docs

- IBM Watson Studio – AutoAI

https://www.ibm.com/cloud/watson-studio/autoai

- Scikit-learn: Machine Learning in Python

https://scikit-learn.org/

- Python Libraries – Pandas, NumPy, Matplotlib

https://pypi.org

IBM CERTIFICATIONS

Getting Started with Artificial Intelligence

IBM SkillsBuild

IBM

In recognition of the commitment to achieve professional excellence

HARSH TRIVEDI

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/3086e703-0abf-48dd-8f60-2c7899c6c14f

IBM

edunet foundation

IBM CERTIFICATIONS

In recognition of the commitment to achieve professional excellence

HARSH TRIVEDI

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/0749167d-9b58-4466-8d92-594072f49ef7

IBM

edunet foundation

IBM CERTIFICATIONS

IBM **SkillsBuild** Completion Certificate

This certificate is presented to

HARSH TRIVEDI

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