Electrical Fault Classification

Chad Hucey | Deep Suchak | Isha Jain | Tanish Kandivlikar

Table of contents

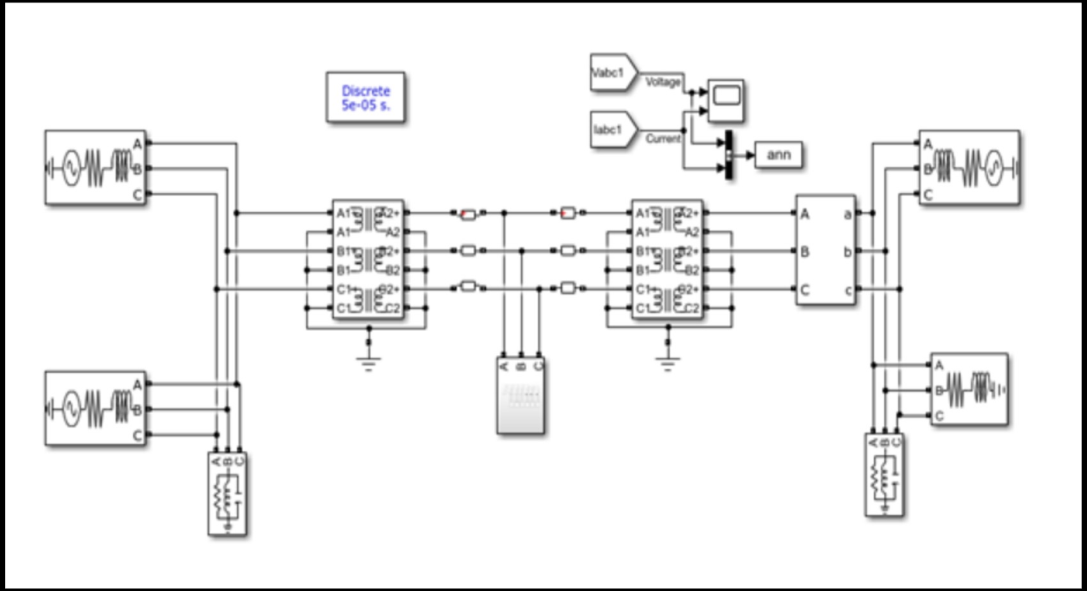
- Problem statement
- Dataset
- Exploratory Data Analysis
- Feature Engineering
- Machine Learning
- Results
- Challenges
- Conclusion

Problem Statement

- Distribution Systems are dynamic and complex
- Faults and/or disruptions of service can happen frequently
- Operators need to be able to accurately detect different faults and resolve them quickly
- Accurately predict 6 types of faults

Dataset

- Obtained from Kaggle
- 6 features
  - Current and voltages in the three lines



Prakash, E Sathya. “Electrical Fault Detection and Classification.” *Kaggle*, 22 May 2021, www.kaggle.com/datasets/esathyaprakash/electrical-fault-detection-and-classification/?select=classData.csv.

- 4 output columns
  - {G C B A}
  - Combined to create the Target Variable

Output Column	Meaning	Output Entry
G	Ground	0 = not fault, 1 = fault
C	Line C	0 = not fault, 1 = fault
B	Line B	0 = not fault, 1 = fault
A	Line A	0 = not fault, 1 = fault

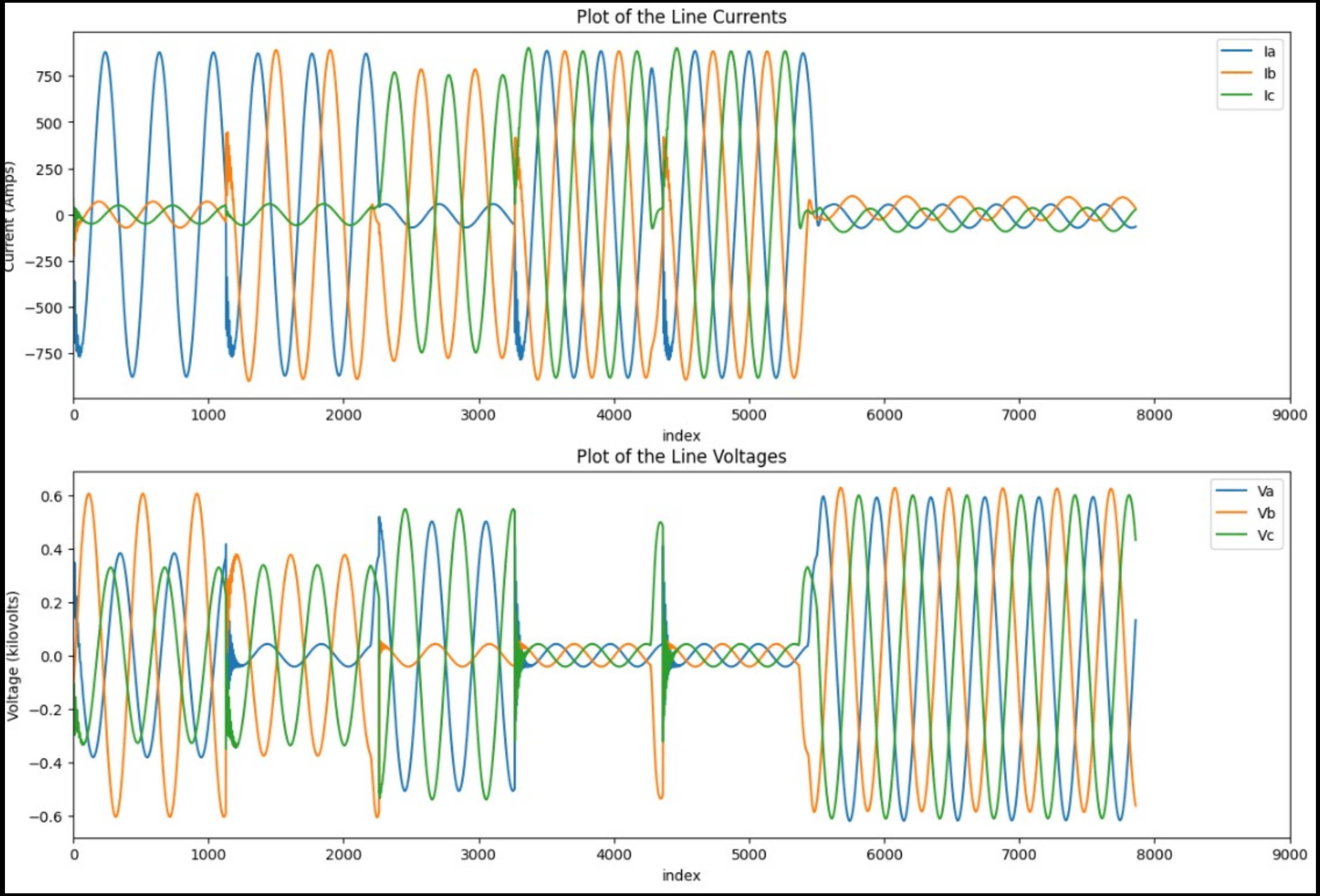
Classes

- Format [G C B A}:

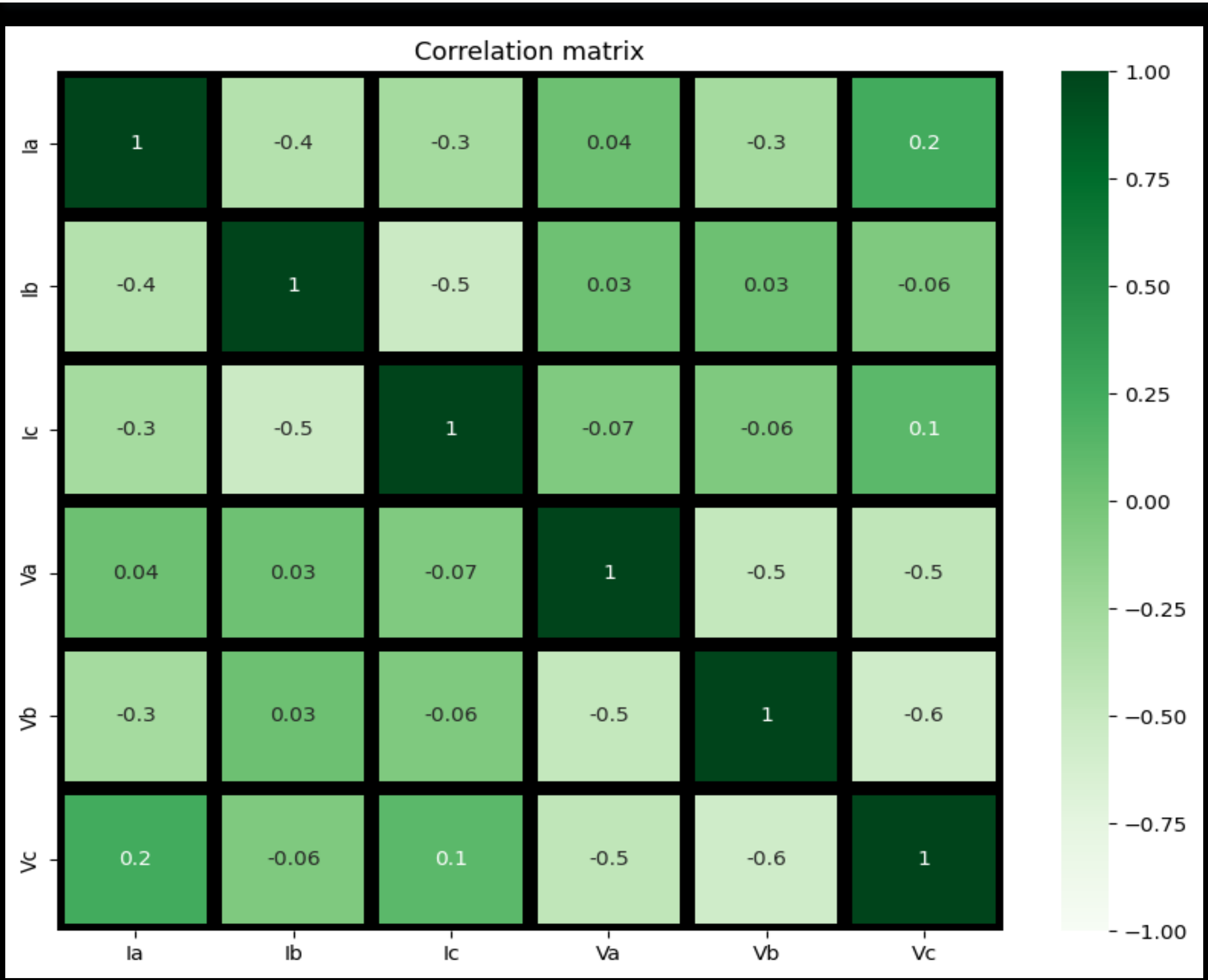
  [0 0 0 0] - No Fault
  [1 0 0 1] - LG fault (Between Phase A and ground)
  [0 1 1 0] - LL fault (Between Phase B and Phase C)
  [1 0 1 1] - LLG Fault (Between Phases A,B and ground)
  [0 1 1 1] - LLL Fault (Between all three phases)
  [1 1 1 1] - LLLG fault (Three phase symmetrical fault)

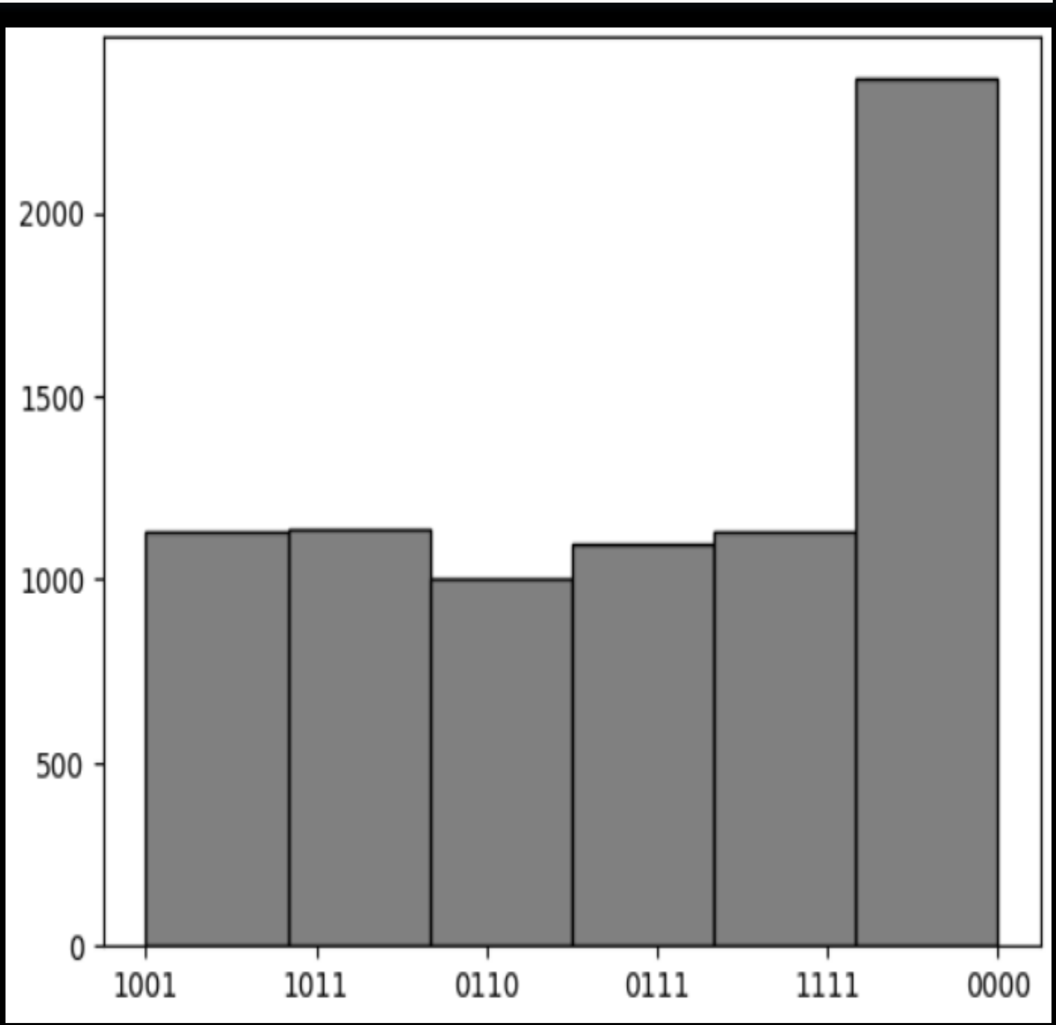
Exploratory Data Analysis

- 7861 rows and 10 columns
- No missing values
- Periodic data



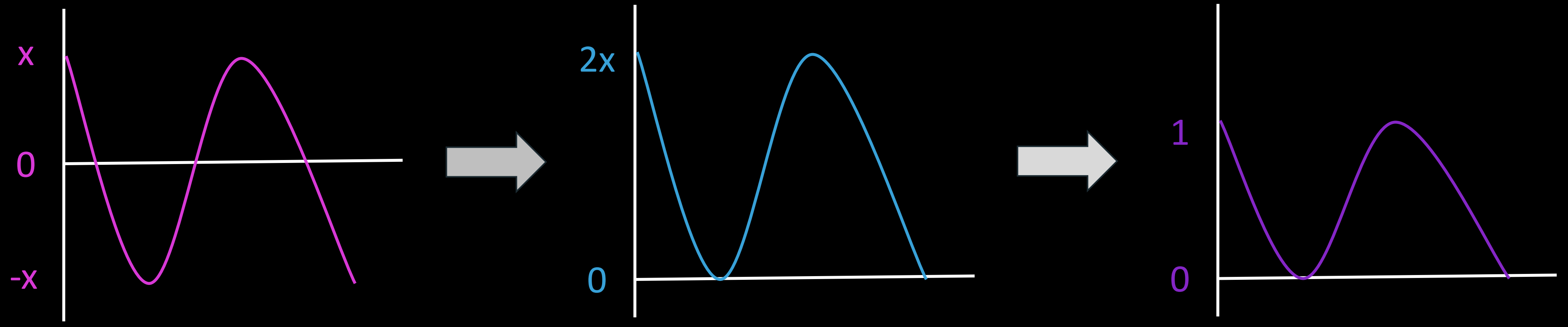
EDA Continued





Feature Engineering

- Used the 6 given features as predictors, and the combined output column as the target



Original Data

Data points >= 0

Scaled from 0 to 1

Machine Learning

Logistic Regression:

Tested if the faults can be separated by linear boundaries in feature space

Served as a baseline to measure the performance of more complex models

K-Nearest Neighbors (KNN):

KNN could effectively classify faults by looking at 'nearby' instances in the feature space

Decision Tree:

Performs feature selection implicitly, which can be beneficial if some features are more indicative of faults than others

A decision tree uses thresholds to make decisions, making it a good fit for data with clear rule-based patterns.

Machine Learning

Random Forest:

Electrical fault detection can be complex and might not be well-modeled by a single decision tree.

Random forests can capture a wider range of fault signatures by combining the learning power of multiple trees.

Support Vector Machine:

SVMs can model complex, non-linear relationships using the kernel trick, and they focus on maximizing the margin, which can lead to better generalization.

The ability of SVMs to use different kernels allows them to model the complex boundaries that could occur in the feature space of electrical signals.

Results

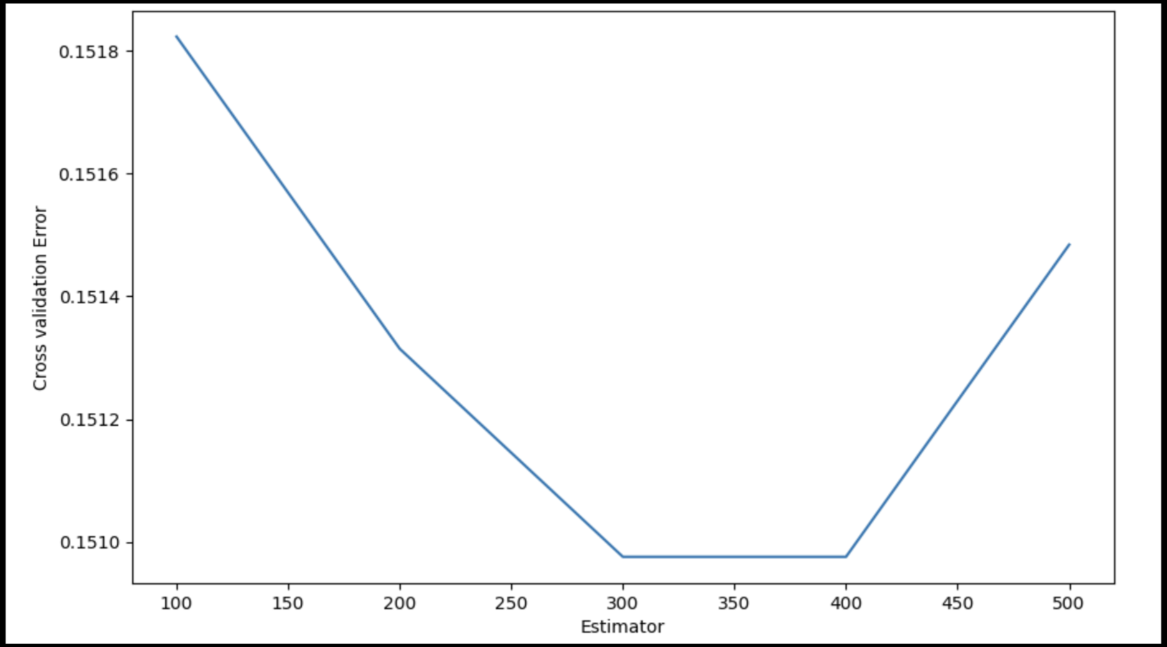
Baseline Models

Logistic Regression ”multiclass” = “multinominal	65%
KNN “n_neighbors” = 5	79.3%
Decision Tree “max_depth” = 10	83.3%
Random Forest “n_estimators” = 100	87.2%
Support Vector Machine “kernel” = “rbf”	76.1%

Tuned Models

KNN “n_neighbors” = 8	82.25%
Decision Tree “max_depth” = 17	85.69%
Random Forest “n_estimators” = 400	87.33%
Support Vector Machine c = 100, gamma = 1, kernel = rbf	83.26%

Cross Validation and its Importance



cross-validation error v/s number of estimators in Random Forest Classifier

Definition:

Cross-validation is a key method in machine learning, assessing how well a model predicts new data. Instead of a single data split, it divides the dataset into subsets, training on one and testing on another in an iterative process. This thorough evaluation ensures the model's performance and reliability across diverse data scenarios.

Importance:

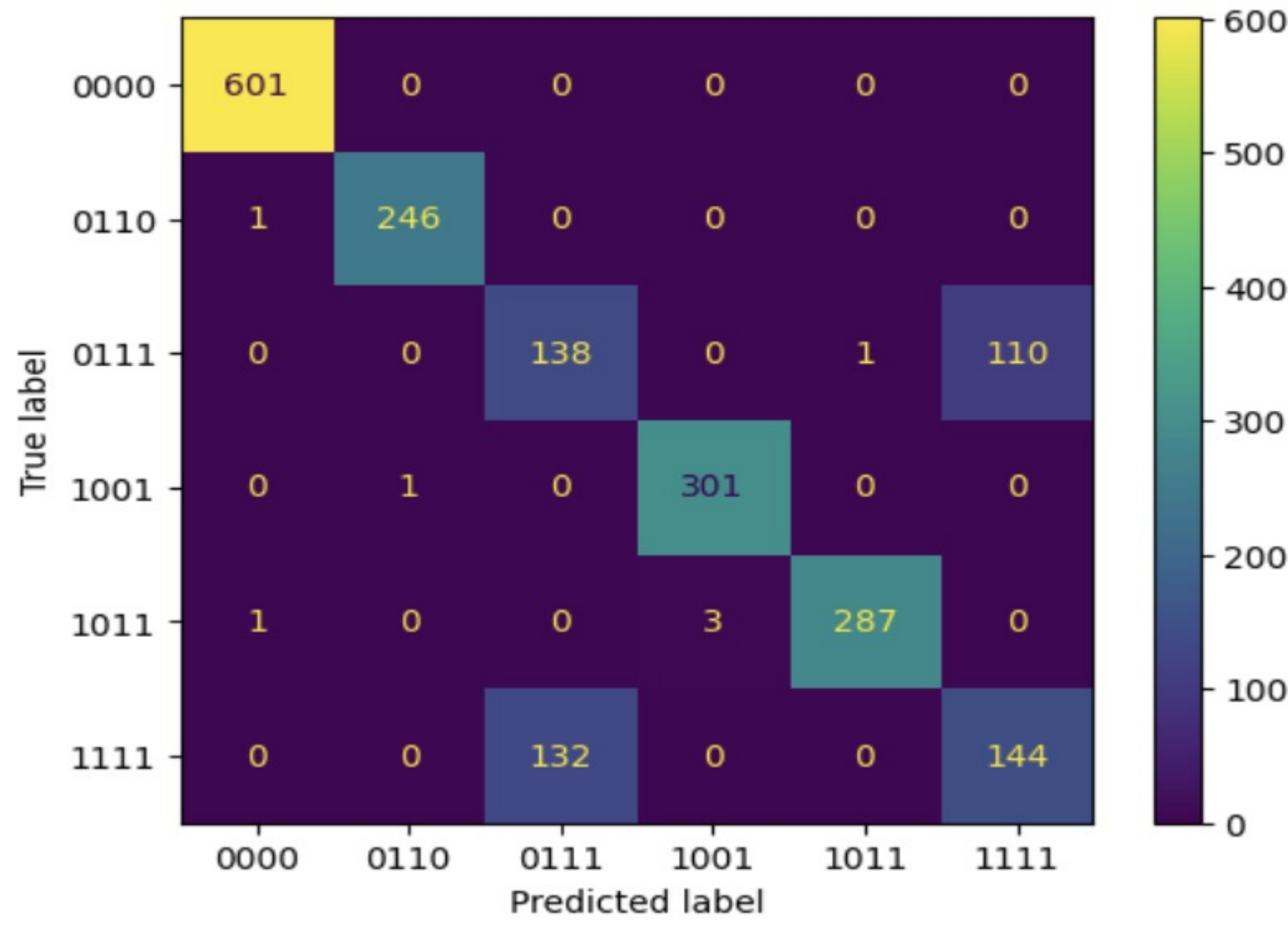
Cross-validation is crucial for three main reasons.
- it ensures a model can generalize, making accurate predictions on unseen data.
- it provides a robust assessment of real-world performance by testing the model on various data subsets.
- it cross-validation helps prevent overfitting, where a model may memorize the training data but struggle with new data.

Challenges

- Poor performance of logistic regression mode
- Due to high bias
- Too simple for this application
- Would likely need more sophisticated feature engineering

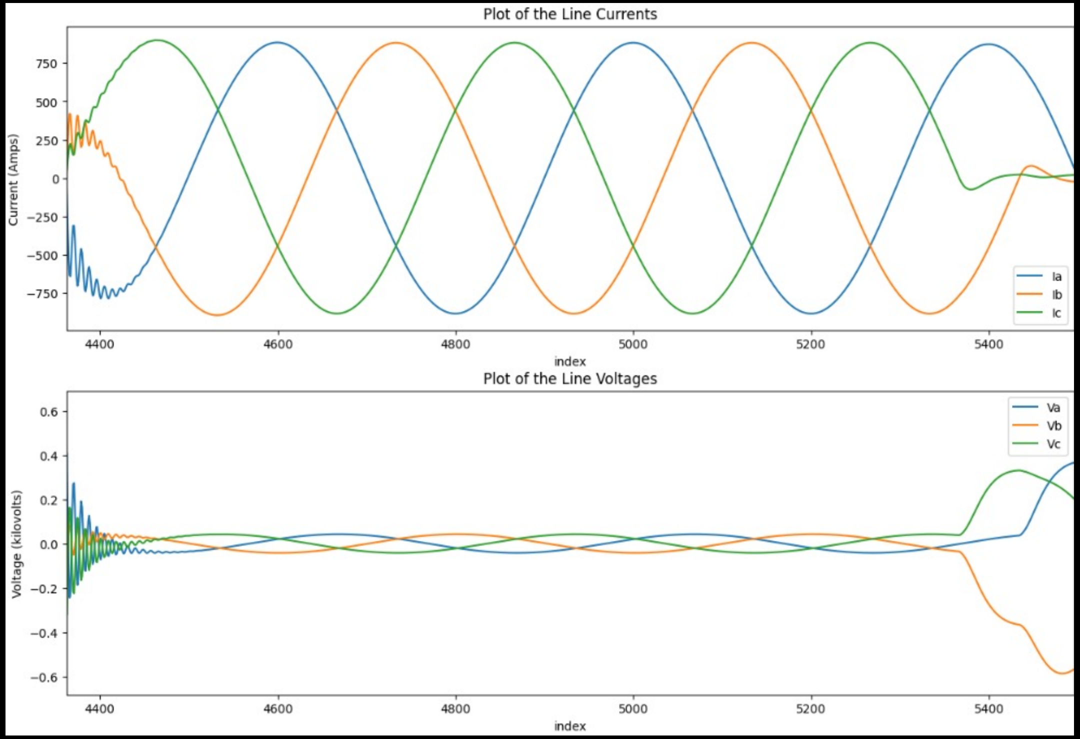
Challenges Continued

- Difficulty differentiating between class 0111 and 1111
- Predicted class “0111” 270 times
    - Wrong 132 times
- Predicted class “1111” 254 times
    - Wrong 110 times

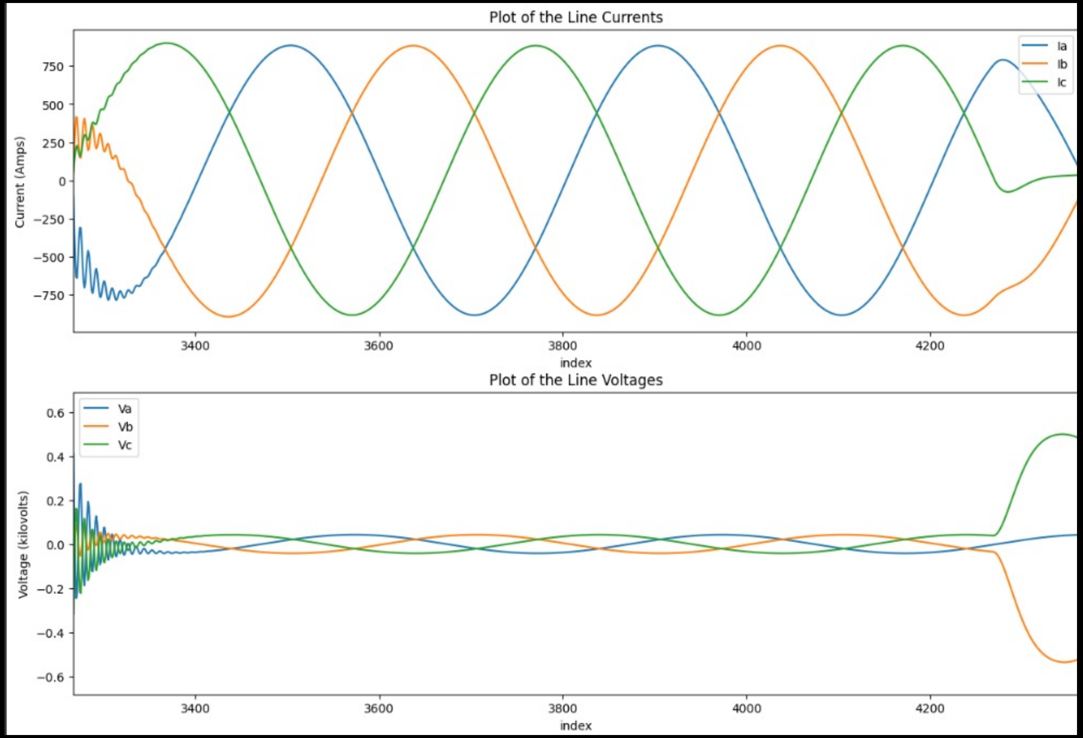


True label \ Predicted label	0000	0110	0111	1001	1011	1111
0000	601	0	0	0	0	0
0110	1	246	0	0	0	0
0111	0	0	138	0	1	110
1001	0	1	0	301	0	0
1011	1	0	0	3	287	0
1111	0	0	132	0	0	144

[0 1 1 1] - LLL Fault (Between all phases, A, B and C)
[1 1 1 1] - LLLG fault (Between phase A, B, C and Ground)



Class 0111



Class 1111

Class 0111 vs 1111

- Very similar trends
- ML models struggled to accurately predict these two classes

Conclusion

Through rigorous experimentation, we observed distinct performances across different models

Logistic Regression, while simple and effective for linear relationships, may fall short in capturing the non-linear intricacies of electrical fault data

K-Nearest Neighbours and Support Vector Machines performed well but were outperformed by Random Forests

Decision Trees, though capable of capturing complex patterns, exhibited a tendency to overfit to our dataset

Notably, Random Forest emerged as the most robust performer, effectively balancing complexity and generalization

Future works could involve gathering more data and/or more sophisticated techniques

THANK YOU!

ANY QUESTIONS?