Machine Learning Report on Energy Consumption of Fischertechnik Sorting Line

Aaditya Neupane

May 2025

1 Introduction

- This report shows Data collection, Preparation, Feature Extraction and Selection, Feature Reduction, and approaches used to train ml models on data collected from the Fischertechnik Sorting Line.

2 Data Collection

- Energy consumption data on following objects were collected from Sorting Line using an arduino. Each time an object is passed through the sorting line it recorded the energy data into a CSV file.

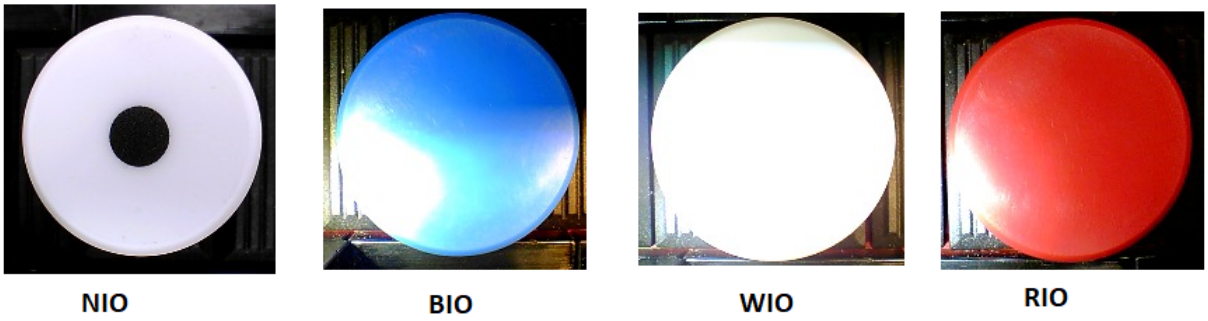

Figure 1: Objects

- Data was collected at different frequencies: 100hz, 400hz, 800hz
- Data for each objects were recorded 1000 times; creating 4000 total data for each frequency
- Initial features on the collected data:
  Datum, Uhrzeit, I_In, V_In, P_In(W), I_Out(A), V_Out(V), P_Out(W), Temp(°C), Energie_In(Wh), Energie_Out(Wh)

3 Data Preperation

- For each frequency, all the individual csv files for all objects were merged.
- While merging, new column was added to indicate rows their particular object-labels
  (target column : "color")

- Following Unnecessary and Derived columns were dropped: 'Datum', 'P_In(W)','P_Out(W)', 'Energie_In(Wh)','Energie_Out(Wh)'
- Then the dataframe was divided into train and test and saved as csv files on different train:test ratios:- 20:80, 50:50, 80:20

4 Feature Extraction and Selection

- Features were extracted using 2 methods:
  - Tsfresh
  - Wavelet Transformation
- For Wavelet Transformation, "energy", "entropy", "variance" features were extracted at Decomposition levels of 2,3, and 4 . Then it was used to train and test the models
- Tsfresh was used to extract and select features. Then, the selected features were used to train and test the models.

5 ML models and Result

- Following models were trained:
  - XGBoost
  - RandomForest
- Results using Tsfresh :

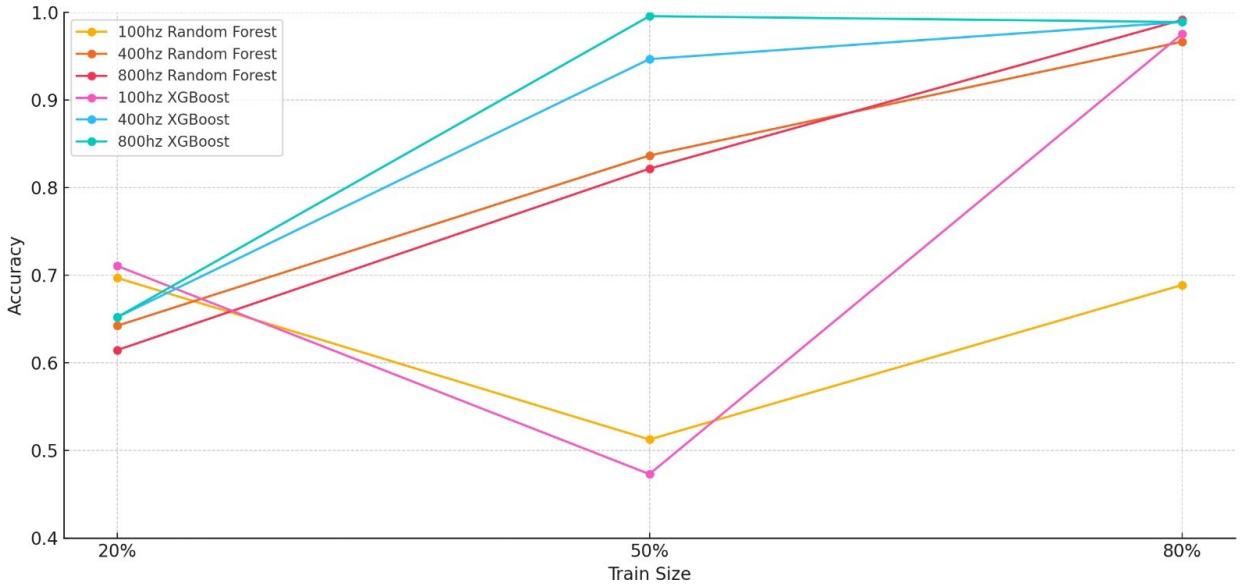

Figure 2: Tsfresh: Model accuracy vs Training size

- Results using Wavelet Transformation:
  - With 20% Train Data

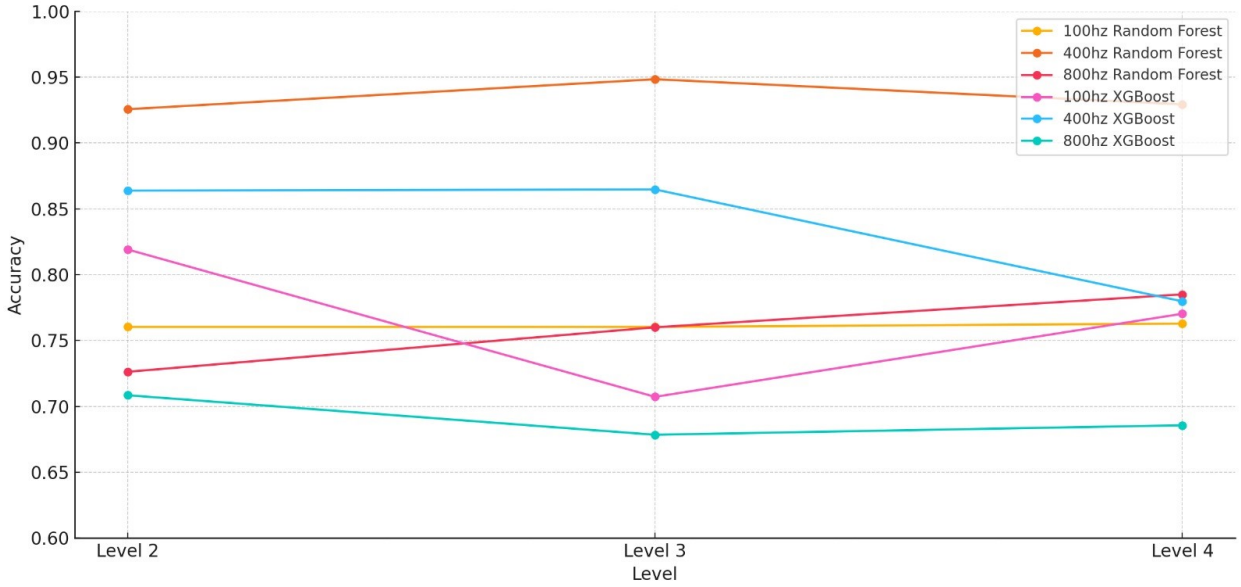

Figure 3: Wavelet Transformation: Model accuracy vs Decomposition Level

  - With 50% Train Data

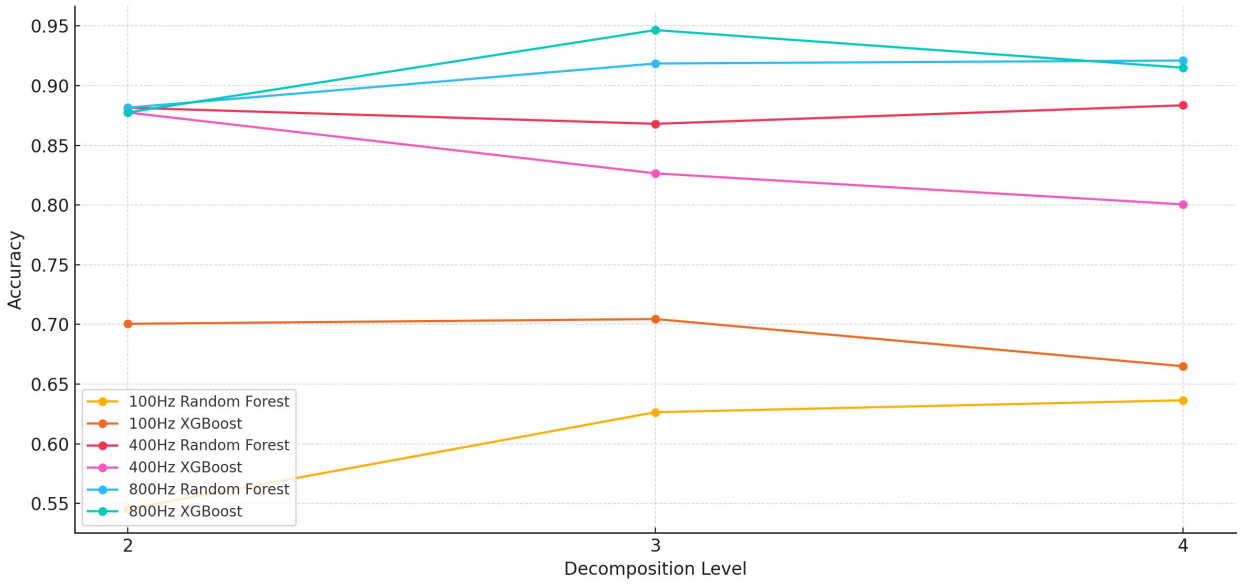

Figure 4: Wavelet Transformation: Model accuracy vs Decomposition Level

– With 80% Train Data

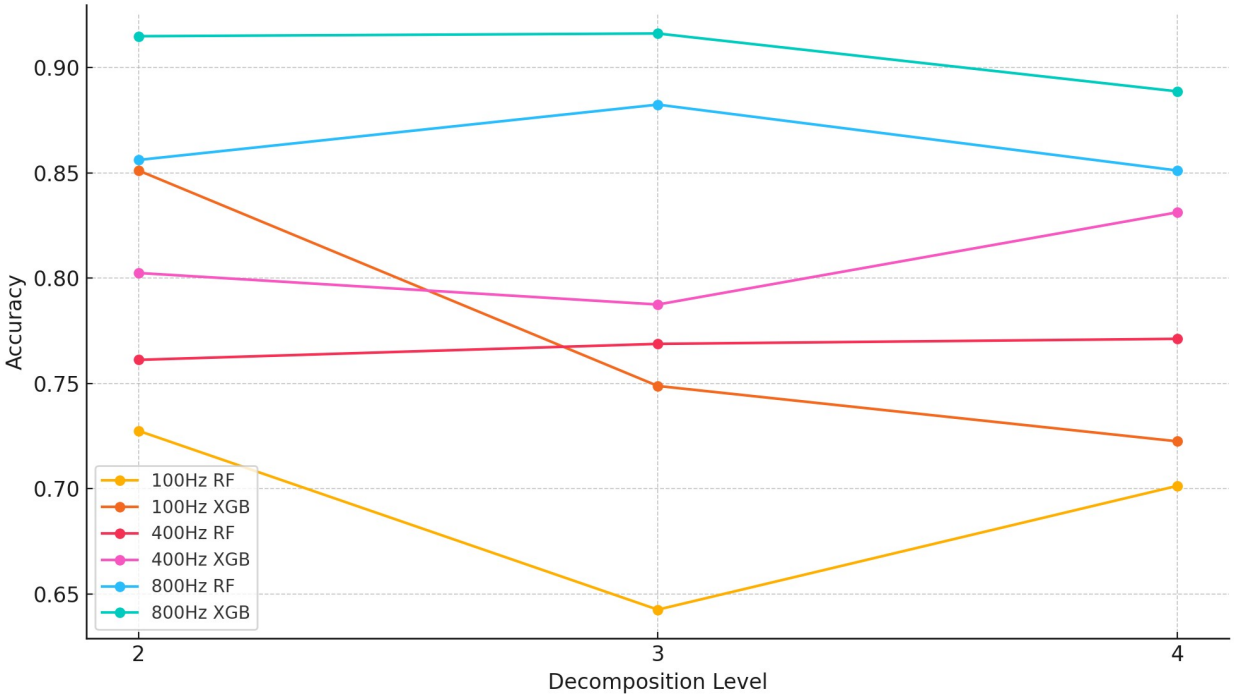

Figure 5: Wavelet Transformation: Model accuracy vs Decomposition Level

- Best models with Tsfresh:
  - 100hz : XGBoost with 80% train split
    Accuracy: 0.9750
    Precision: 0.9766

$$\begin{bmatrix} 198 & 0 & 0 & 2 \\ 0 & 188 & 0 & 12 \\ 0 & 1 & 195 & 4 \\ 0 & 1 & 0 & 199 \end{bmatrix}$$

  - 400hz : XGBoost with 80% train split
    Accuracy: 0.9888
    Precision: 0.9889

$$\begin{bmatrix} 199 & 0 & 1 & 0 \\ 0 & 197 & 3 & 0 \\ 1 & 1 & 198 & 0 \\ 0 & 1 & 2 & 197 \end{bmatrix}$$

  - 800hz : XGBoost with 50% train split
    Accuracy: 0.9955
    Precision: 0.9955

$$\begin{bmatrix} 499 & 0 & 1 & 0 \\ 0 & 495 & 0 & 5 \\ 2 & 0 & 497 & 1 \\ 0 & 0 & 0 & 500 \end{bmatrix}$$

- Best models with Wavelet Transformation:
  - 100hz : XGBoost with Decomposition level 2 on 80% train split
    Accuracy: 0.8512
    Precision: 0.8614

$$\begin{bmatrix} 188 & 0 & 11 & 1 \\ 0 & 166 & 9 & 25 \\ 0 & 67 & 133 & 0 \\ 0 & 6 & 0 & 194 \end{bmatrix}$$

  - 400hz : RandomForest with Decomposition level 3 on 20% train split
    Accuracy 0.9484
    Precision 0.9487

$$\begin{bmatrix} 776 & 0 & 24 & 0 \\ 0 & 743 & 40 & 17 \\ 14 & 28 & 750 & 8 \\ 0 & 34 & 0 & 766 \end{bmatrix}$$

  - 800hz : XGBoost with Decomposition level 3 on 50% train split
    Accuracy: 0.9465
    Precision: 0.9483

$$\begin{bmatrix} 499 & 0 & 0 & 1 \\ 0 & 470 & 11 & 19 \\ 19 & 20 & 428 & 33 \\ 0 & 4 & 0 & 496 \end{bmatrix}$$

6 Feature Reduction

- Since, 800hz XGBoost 50% train split with Tsfresh trained the best model, same extracted features was used for feature reduction.
- Features were selected based on the importance of features.
- Before feature reduction (2122 features)
  Accuracy: 0.9955
  Precision: 0.9955

$$\begin{bmatrix} 499 & 0 & 1 & 0 \\ 0 & 495 & 0 & 5 \\ 2 & 0 & 497 & 1 \\ 0 & 0 & 0 & 500 \end{bmatrix}$$

- Reduction to 187 features
  Accuracy: 0.9955
  Precision: 0.9955

$$\begin{bmatrix} 499 & 0 & 1 & 0 \\ 0 & 496 & 0 & 4 \\ 2 & 1 & 496 & 1 \\ 0 & 0 & 0 & 500 \end{bmatrix}$$

- Reduction to 50 Features

  Accuracy: 0.9930
  Precision: 0.9931

$$\begin{bmatrix} 499 & 0 & 1 & 0 \\ 1 & 490 & 2 & 7 \\ 2 & 0 & 497 & 1 \\ 0 & 0 & 0 & 500 \end{bmatrix}$$

- Reduction to a single Feature

  Accuracy: 0.9395
  Precision: 0.9403

$$\begin{bmatrix} 498 & 0 & 1 & 1 \\ 0 & 421 & 61 & 18 \\ 9 & 11 & 479 & 1 \\ 0 & 19 & 0 & 481 \end{bmatrix}$$

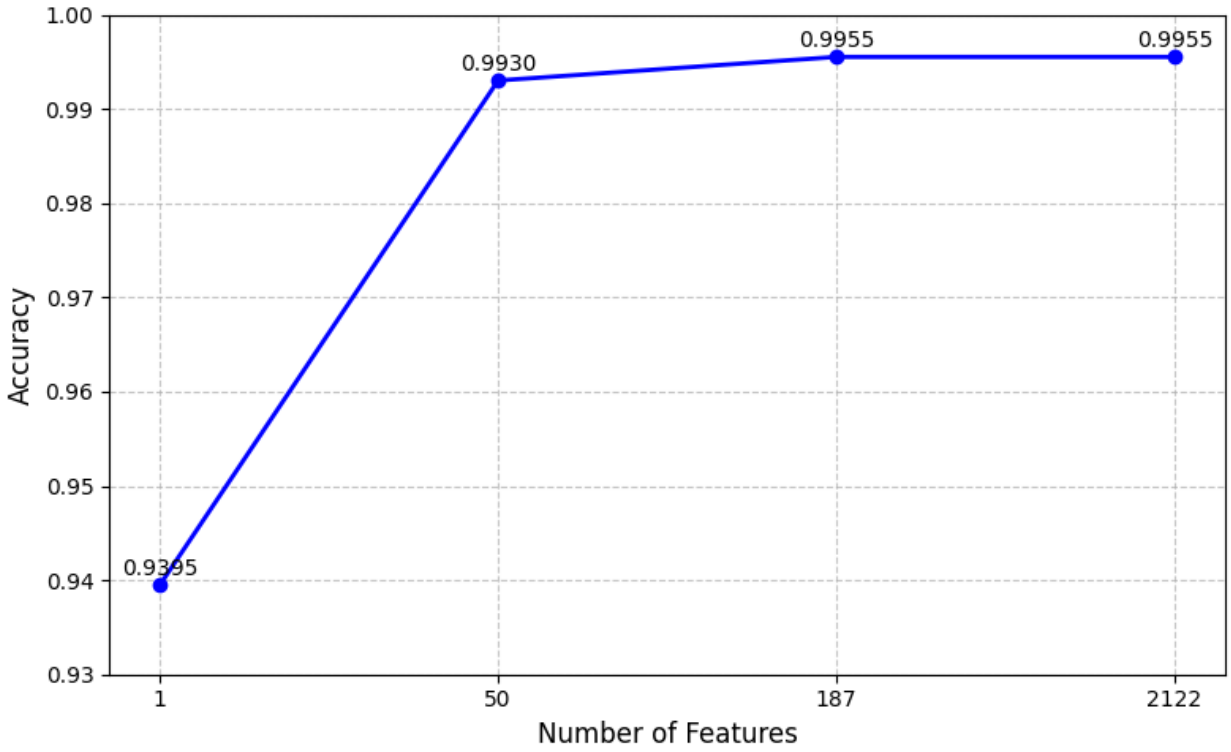

Figure 6: Impact of Feature Reduction