MACHINERY FAULT PREDICTION FROM SENSOR DATA

SINAN BAYATI
KAREEMA KILANI

INTRODUCTION

INTRODUCTION

- Motors are the backbone of industrial systems, powering machines across the globe

- Unexpected motor failures can lead to:
  - Costly downtime
  - Safety hazards
  - Expensive repairs

BACKGROUND

BACKGROUND

- Traditional motor fault detection relies on:
  - Regular manual inspections
  - Periodic maintenance schedules
  - Simple vibration threshold-based systems

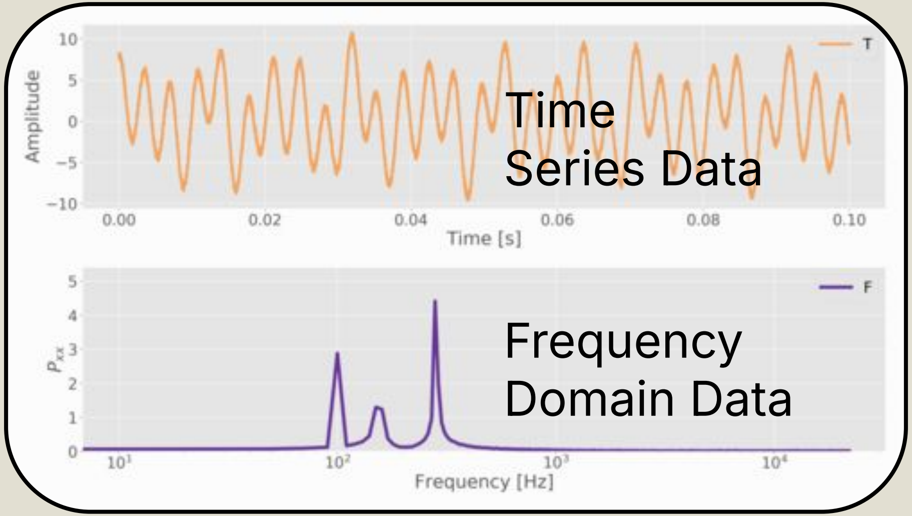
- Challenges with traditional methods:
  - Labor-intensive
  - Not real-time
  - Inability to predict specific fault types or patterns

PROBLEM

PROBLEM

- Motors experience various types of faults:
  - Bearing faults
  - Rotor imbalances

- These faults manifest differently:
  - Different vibration patterns
  - Varying amplitudes and frequencies



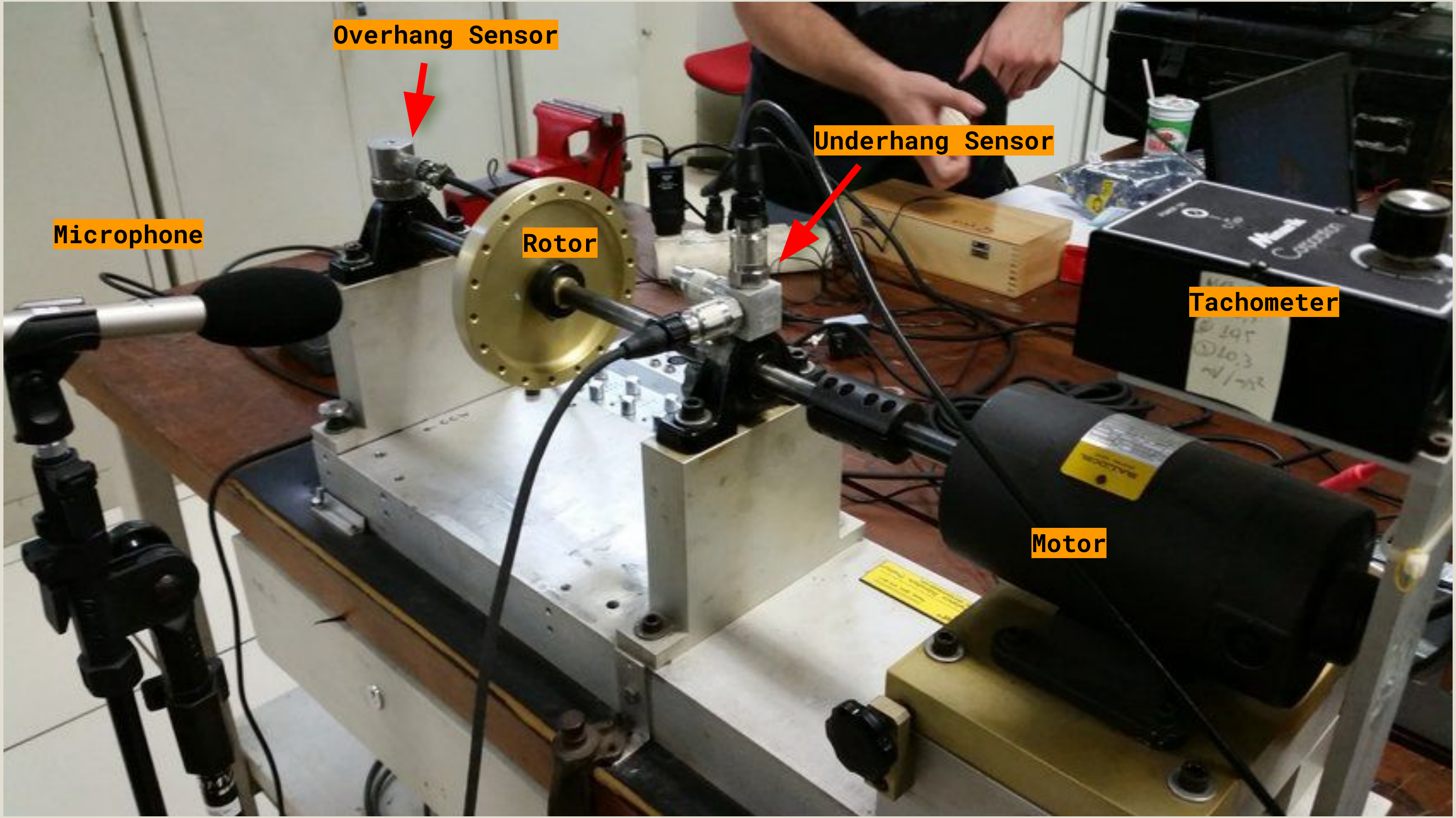
OBJECTIVE

OBJECTIVE

- The goal of this project is to:
  - Develop a machine learning model to classify motor faults using vibration data.
  - Use vibration data from a lab test bench setup as input.
  - Achieve high accuracy in identifying fault types in real time.

- Outcome:
  - Reduce downtime
  - Enhance predictive maintenance strategies
  - Improve system reliability and safety

SETUP


Overhang Sensor
Microphone
Rotor
Underhang Sensor
Tachometer
Motor

DATA

SAMPLE DATA

50 kHz data for all channels

Tacho	Underhand Axial	Underhand Radial	Underhand Tangential	overhang axial	overhang Radial	overhang Tangential	Speaker
4.6038	-0.051295	-0.19405	-0.060071	-0.41809	0.036547	-0.11043	0.11831
4.5703	-0.96908	0.038033	-0.028329	-0.43081	0.041924	-0.14331	-0.071527
4.587	0.89127	0.072973	0.0074526	-0.40017	0.04109	-0.11984	0.043445
4.5887	-1.716	-0.32929	-0.033063	-0.50281	0.040474	-0.2527	0.023901
4.5675	1.2403	0.35401	0.04046	-0.36806	0.044062	-0.14258	-0.05488
4.6052	-1.5955	-0.47204	-0.071376	-0.49493	0.045082	-0.27611	0.12137
4.5556	0.89214	0.42547	0.0094502	-0.3614	0.047495	-0.16086	-0.10988
4.6097	-0.79182	-0.40115	-0.09155	-0.45266	0.048458	-0.24753	0.11269
4.5583	-0.051937	0.23298	-0.027103	-0.38217	0.049433	-0.20108	-0.10407
4.5966	0.19039	-0.14388	-0.05264	-0.3818	0.046969	-0.18243	0.086498
4.5727	-1.2957	-0.10847	-0.035895	-0.41444	0.046045	-0.21344	-0.056535
4.5783	1.0556	0.15694	0.020054	-0.31527	0.040423	-0.10154	0.017014
4.5916	-1.744	-0.41731	-0.038561	-0.43002	0.040134	-0.21352	0.027747
4.5612	1.1804	0.34835	0.053164	-0.28124	0.038183	-0.051593	-0.0569
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

FEATURE EXTRACTIONS

Mean And STD

Skewness (Detect Asymmetry In The Data)

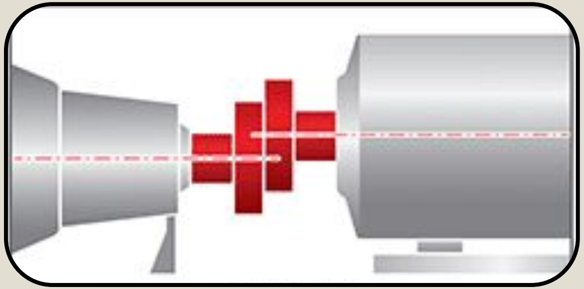
Kurtosis (Detect Outliers In The Data)

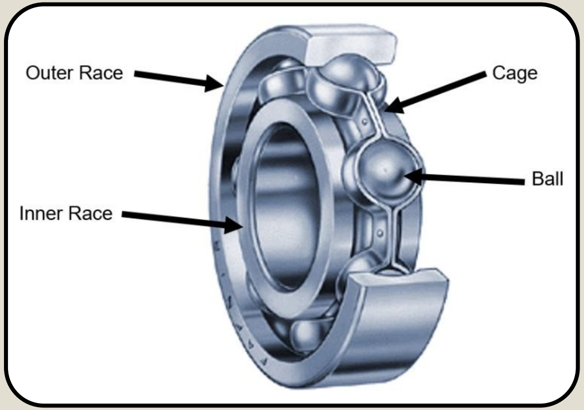
Top 10 Frequencies And Their Respective Power Levels

FAILURE MODES

Different failures were induced into the setup:

- **Normal**
- Horizontal/Vertical misalignment with varying lengths
- Imbalance (adding weights)
- Bearing
    - Cage fault
    - Outer race
    - Ball fault

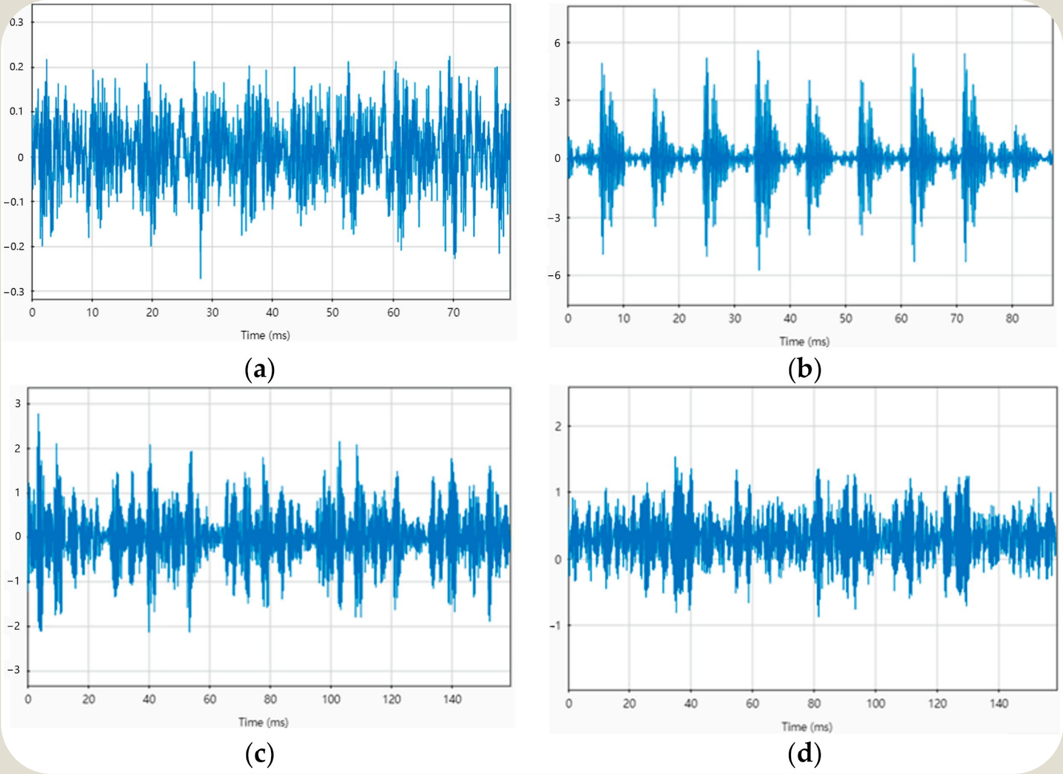




10 Total Fault Types

TIME SERIES DATA

Data in the time series domain

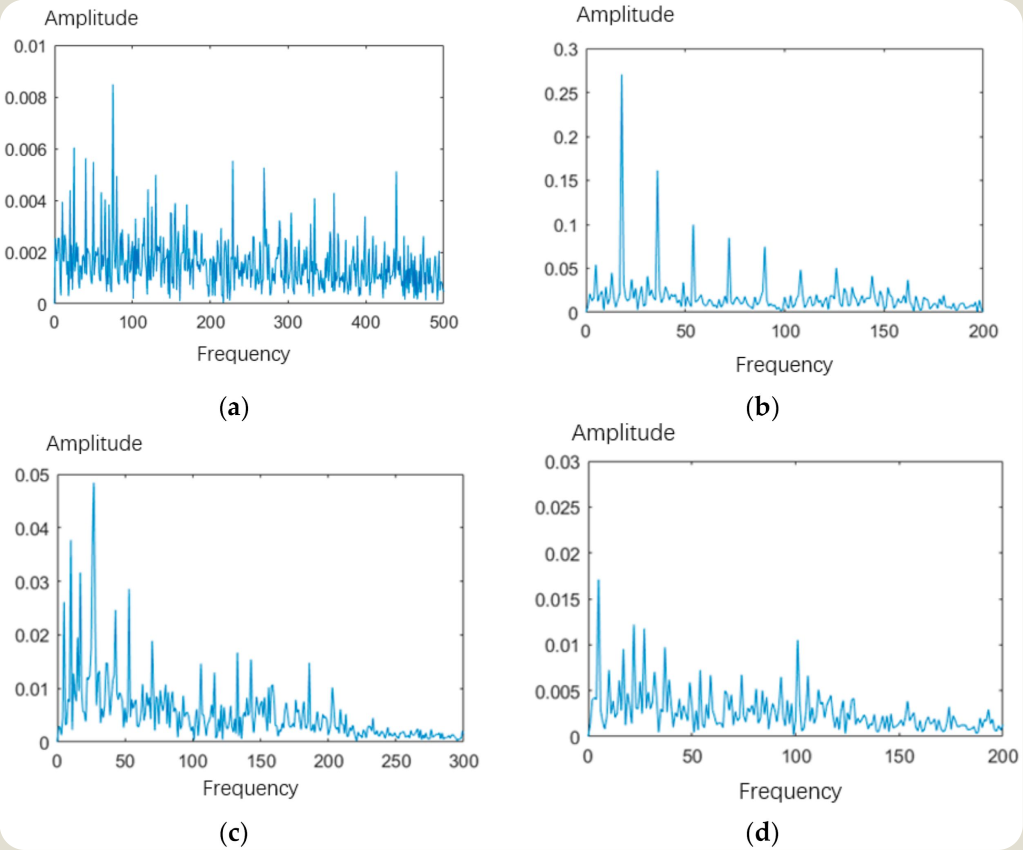


(**a**) Normal bearings
(**b**) Outer-race failure
(**c**) Cage failure
(**d**) Ball failure

FFT DATA

- Data In The Frequency Domain

(**a**) Normal bearings
(**b**) Outer-race failure
(**c**) Cage failure
(**d**) Ball failure



APPLIED SOLUTIONS

APPROACH ONE

Normal vs. Horizontal Misalignment (2.00 mm):

Binary Classification of All Speeds for Normal State vs. Fault State at Each Speed using Time Series Data with an LSTM approach

Dataset Preparation

Convert raw data to DataFrame, separate features (X) and labels (y), reshape for LSTM input.

Data Splitting

Split into training (80%) and test (20%) sets.

Model Definition

Build LSTM model with 1 LSTM layer, **dropout 0.2**, and dense output
Activation: Sigmoid

Model Compilation

Compile with Adam optimizer, **binary cross-entropy**, and accuracy.

Model Training

Train for **10 epochs** with batch size 32, using test set for validation.

APPROACH TWO

Multi-Class Classification of All Fault Types

Multi-Class Classification of Normal and All Faults at a Single Similar Speed ***Using Time Series Data***

Dataset Preparation

Convert raw data to DataFrame, separate features (X) and labels (y), reshape for LSTM input. **(samples, timesteps, features)**

Data Splitting

Split into training **(80%)** and test **(20%)** sets.

Model Definition

Build LSTM model with 1 LSTM layer, dropout, and dense output
Activation: **Softmax** **(10 classes)**

Model Compilation

Compile with Adam optimizer, **sparse categorical cross-entropy**, and accuracy.

Model Training

Train for **50 epochs** with batch size 32, using test set for validation.

LSTM MODEL

Model Type: Sequential → LSTM Layer → Dropout Layer → Dense Layer with Softmax or Sigmoid

Model Type: Sequential

Preprocessed time-series samples are fed into the model.

LSTM Layer

Processes sequential patterns in the time-series data to extract meaningful features.

Dropout Layer

Reduces overfitting by randomly disabling some neurons during training

Dense Layer with Softmax or Sigmoid

Converts the LSTM's output into probabilities for each classification category

APPROACH THREE

Multi-Class Classification of 10 Fault Types

Multi-Class Classification of Normal and 10 Faults at a Single Similar Speed ***Using Feature Extraction***

Feature Extractions → Train-Test Split → Train XGBoost Classifier → Model Compilation

Split raw_data into features (X) and target (y). **NOT THE RAW TIME SERIES DATA**

Split into training **(80%)** and test **(20%)** sets.
169 Features Total

Train the XGBoost model with specified hyperparameters **(n_estimators, learning_rate, max_depth)**.

METRICS

METRICS

APPROACH ONE: BINARY

Test Accuracy: 88.54%

EPOCHS: 10

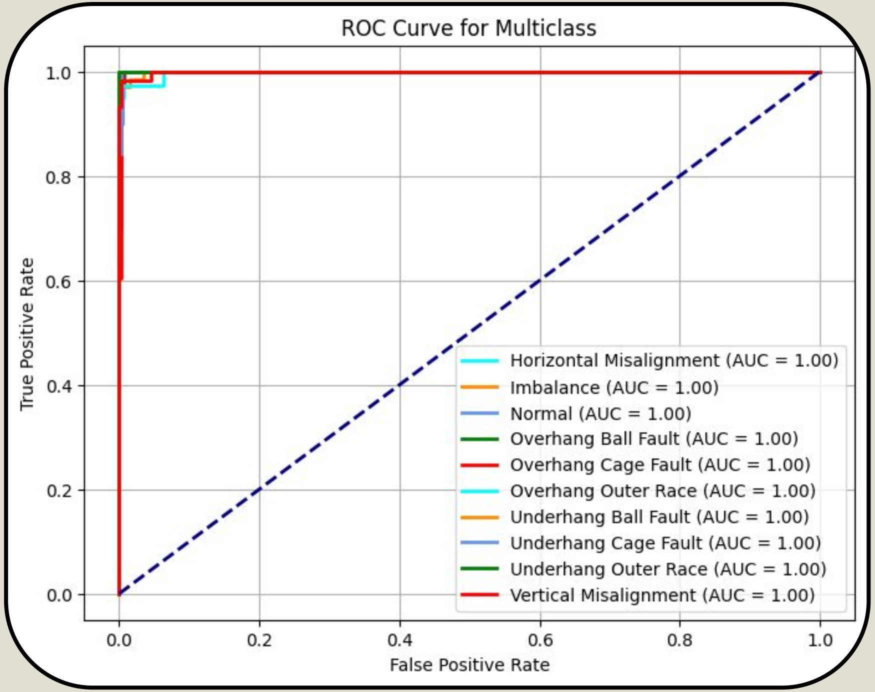
APPROACH TWO: MULTI-CLASS

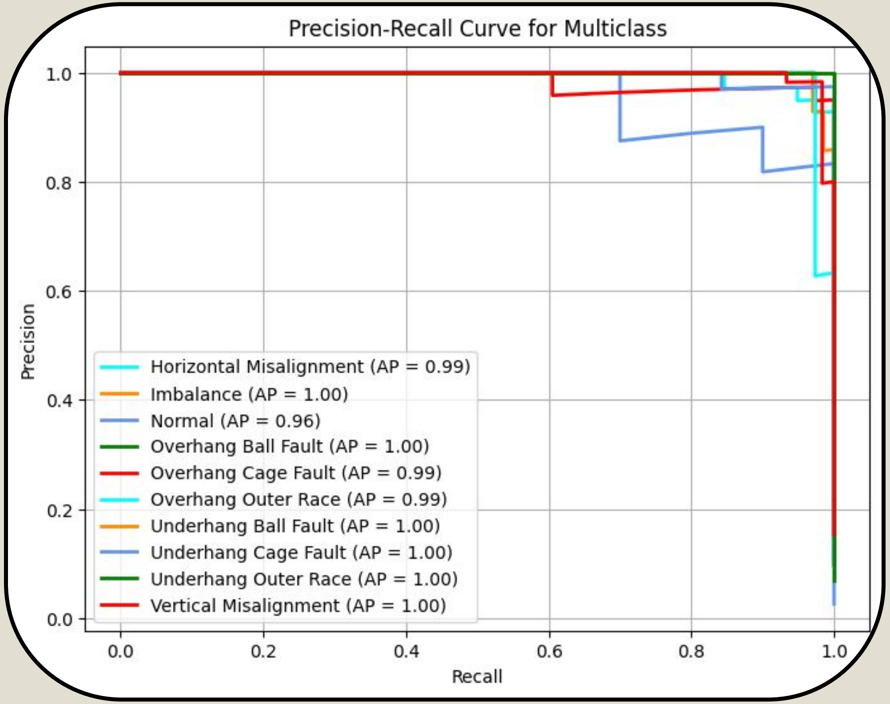
Test Accuracy: 96.02%

EPOCHS: 50

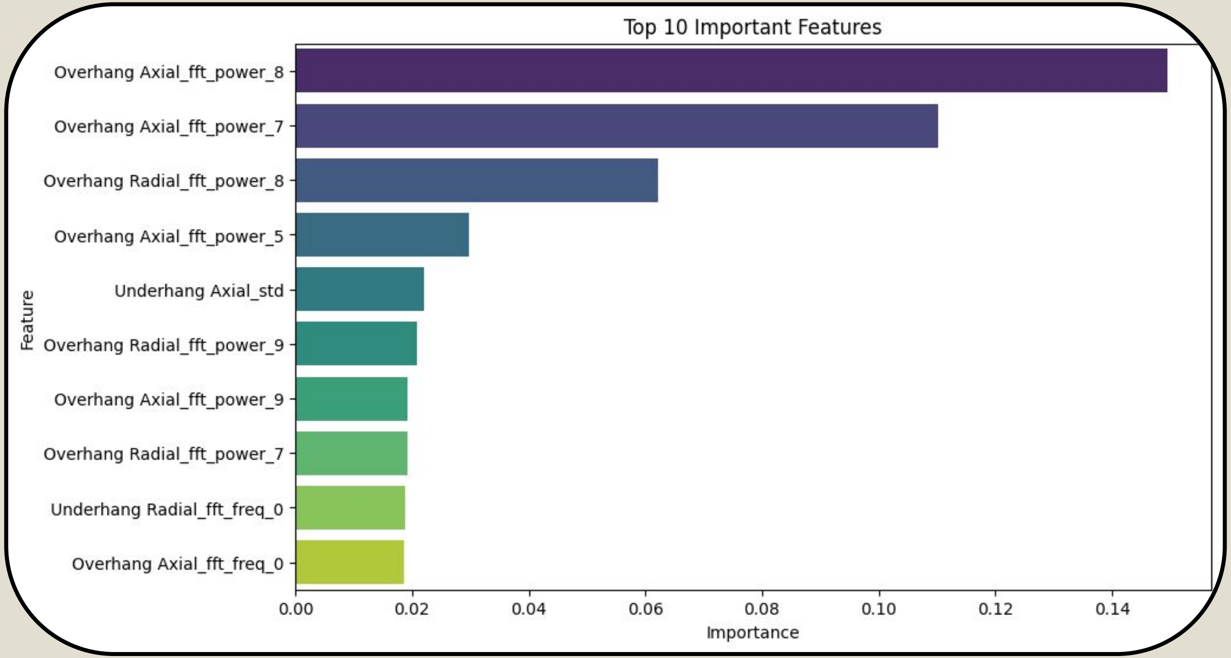
METRICS - APPROACH THREE

Test Accuracy: 98.21%



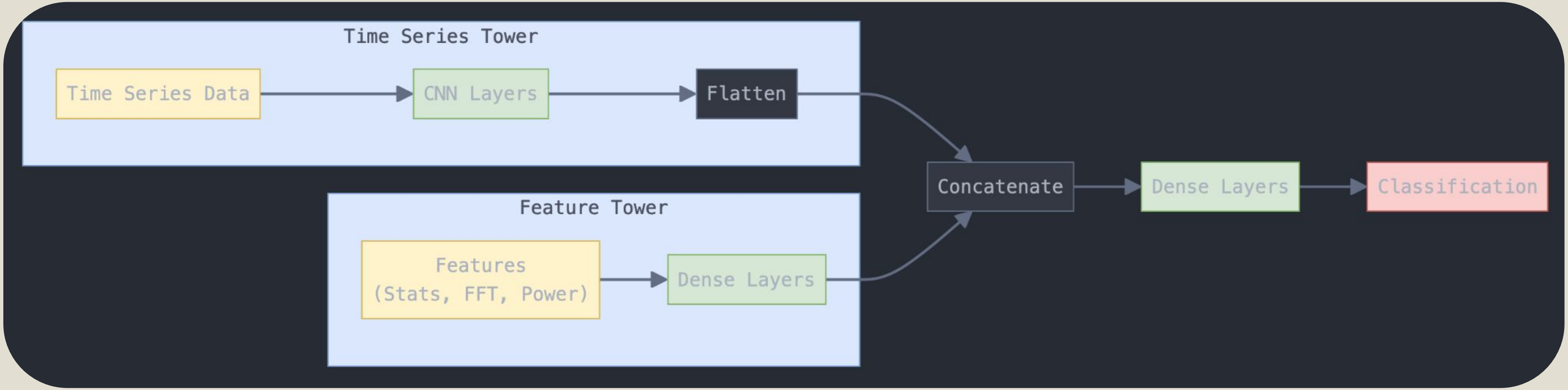


FEATURE IMPORTANCE - APPROACH THREE



FUTURE WORK

FUTURE WORK - TWO TOWER MODEL



THANK YOU