PROJECT REPORT

Table of Contents

Flat Tyer Detection System

Abstract:

Wheel flats are amongst the most common local surface defect in railway wheels, which can result in repetitive high wheel–rail contact forces and thus lead to rapid deterioration and possible failure of wheels and rails if not detected at an early stage. The timely and accurate detection of wheel flats is of great significance to ensure the safety of train operation and reduce maintenance costs. In recent years, with the increase of train speed and load capacity, wheel flat detection is facing greater challenges. This paper focuses on the review of wheel flat detection techniques and flat signal processing methods based on wayside deployment.

Introduction

Railways is considered one of the major means of transportation at present. In India railways is primary mode of freight and passenger transportation. For more than 150 years, Indian Railways has served as a major integrating force, in addition to being a vital mode of transportation. Despite these factors, Indian railway system is expanding at steady peace. The national rail network comprised total route length of 68,584 km (42,616 mi), with more than 132,310 km (82,210 mi) of track and 8,000+ stations and is the fourth-largest in the world. It is one of the busiest networks in the world, transporting more than 11 billion passengers and 1.416 billion tonnes of freight annually. As of August 2024, more than 64,080 km (39,820 mi) of all the routes have been electrified with 25 KV AC electric traction. The rolling stock consisted of 318,196 freight wagons, 84,863 passenger coaches, 14,781 locomotives and other multiple units owned by Indian Railways apart from rail-sets operated by metro rail corporations. With over 1.3 million employees, Indian Railways is India's single largest employer and the world's eighth-largest. In FY22, gross revenue was Rs 85,588.96 crore (US$ 11.44 billion) (until September 2021).

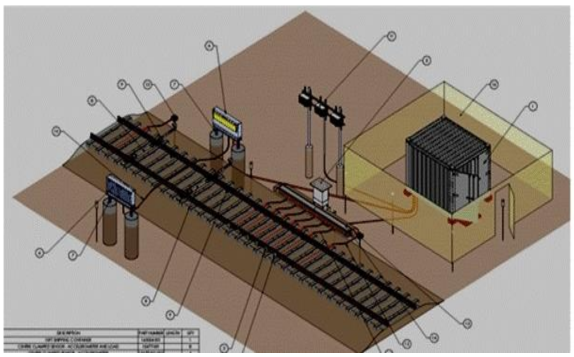
The flat tyer detection system is a pre-existing project in the Indian Railways.it was a test project introduced in the year of 2017, but the project was facing defects till the year 2019.this whole system was named as **ACOUSTIC DETECTION SYSTEM** which had two major projects which are OMRS and WILD relating to the Flat tyer issue, which we are working now presently.

Let's discuss in detail about the pre-existing projects:

OMRS (Online Monitoring of Rolling Stock System):

A-way-side automated inspection system that can detect the faults in the bearings and wheels of trains and catching them before it fails, thus resulting in efficient utilization of the coaches, wagons & locomotives.

OMRS System Overview – Site Details



Challenges of manual inspection The current practice of inspection of Rolling stock over Indian Railways is largely based on manual inspection, which is either trackside Rolling-in-Examination or pit examination of Rolling Stock in stationary or slow-moving condition. The visual inspections are done by trained manpower either in a pit or trackside location but this relies on the individual judgment.

To overcome this challenges of manual inspection new technologies were introduced which were:

System components: Acoustic Bearing Detector (**ABD**)/ Bearing Acoustic Monitor (**RailBAM**) gives an early warning on possible defects in the bearing box, before reaching the stage of hot box. Wheel Impact Load Detector **(WILD**)/Wheel Condition Monitor (**WCM**) system measures the wheel impacts on tracks to identify the flat surface on wheels in Rolling Stock. This system is based on Accelerometer device to measure the wheel impacts. **PhotoTAG** system is used for vehicle identification using Visual (photographic) identification technique.

OMRS online (as per the available article from the year of 2020): Installation of 25 OMRS systems at 20 locations is in progress over entire Indian Railways' network in phase-I. 1st OMRS system has been installed at Panipat in Ambala-Delhi section of Northern Railway in November 2017 and a Central Control Room termed as "National Command Centre (NCC)" for monitoring of all OMRS sites has been set-up at Delhi Kishanganj in March 2018. 6 OMRS systems have been installed and 10 systems are expected to be installed in the current financial year 2019-20,As per the report of Indian transport and logistics news

reference link:- https://www.itln.in/indian-railways-adopting-automated-train-defect-detection?infinitescroll=1

Defects detected by OMRS up to June 2019

- Faults in Bearings by **RailBAM** : Wagons- 33, Coaches – 6, Locomotives -1
- Faults in Wheels by WCM: Coaches – 7

"Encouraged by the results of the deployment of OMRS, including some critical detection which could have potentially been the cause of an accident, not otherwise detectable by normal maintenance procedure, Indian Railways is now going ahead with greater adoption of track-side based maintenance systems with an aim towards predictive maintenance," said the release.

As per the recent records available, OMRS is still in progressive state only. It's still in development stage.

The main component in the OMRS is an Accelerometer. The main task of this accelerometer is to measure the wheel impacts of the train rolling stocks. Its is placed on the tracks to measure the impact forces and vibrations exerted by the wheel of the train. This placement is done for the project WILD and for the project RailBAM accelerometer are used to detect acoustic emissions of from the bearings.

How an accelerometer works?

Accelerometer is a device that measures the vibration or acceleration of motion of an object or a structure. It works on the detection principle of detecting the vibrations and moments to monitor the condition of wheels and bearings.

Here even though an accelerometer is used it still needs man power to maintain and operate them.to overcome this problem we should install a trained model which has most of the problem cases and has an intelligence to solve these problems.

To train the model according to the sounds of train we need to have:

1. **Train sounds in different climates (**to avoid disturbance due to rain, train honking sound**)**
2. **Train sounds at different speeds**

3. **Difference between goods and a passenger train sound**
4. **Flat tyer sound**

Many researchers used several machine learning algorithms to analyze the data, implement a learning scheme, and apply intelligent decisions regarding the presence of wheel flats, including artificial neural networks (ANN), principal component analysis (PCA), and support vector machines (SVM). Although a number of publications have been published about railway defect detection, there is limited literature on automatic early wheel flat detection to the best of the authors' knowledge. According to most wheel flat detection schemes, there is no indicator that can automatically differentiate a defective from a healthy wheel. Additionally, in all the studies, multiple sensors are required to distinguish a defective wheel from a healthy one. Consequently, one of the goals of this research is to develop an unsupervised machine learning algorithm that can automatically detect a wheel flat, using only one sensor installed on the rail. The goal of this paper is to develop an acoustic flat wheel detection model which is capable of detecting healthy wheels from defective ones.

As part of the contract with COFMOW, **Wabtec's** teams in Australia and **India** will lead the design, development, supply, installation, and commissioning of 97 **OMRS** equipment sets in various zonal railway locations across the country.

- **Characteristic Force Pattern:** A sharp increase in vertical force followed by a rapid decrease as the flat spot makes contact with the rail and then leaves it.
- **Other Indicators:** Increased vibration, noise, and potential damage to the track.

Irregular Patterns:

- **Characteristic Force Pattern:** Fluctuations in the vertical force, often with a repetitive pattern.
- **Other Indicators:** Uneven wear on the rail, increased rolling resistance, and potential for derailment.

Other Defects:

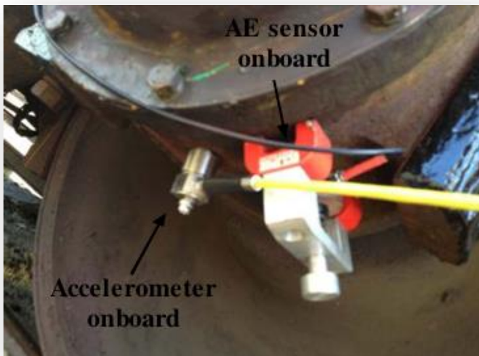
- **Cracks:** Can cause sudden, sharp increases in vertical force as the crack propagates.
- **Corrosion:** Can lead to gradual increases in vertical force as the tire material weakens.

Literature Review

1. **Advanced wayside condition monitoring of rolling stock wheelsets**

A wayside monitoring system is typically installed in or next to the track to detect and identify deterioration of wheel and axle bearings before failure can occur by measuring one or more parameters. Wayside monitoring technologies depending on their nature can be classified as reactive or predictive.

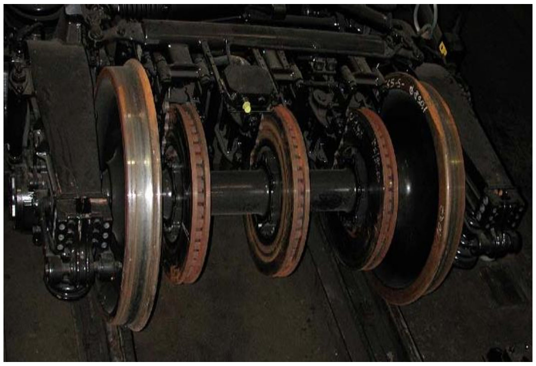
Reactive systems detect actual faults on the vehicles. In most cases the information from these systems is not suitable for trending, but is of importance to protect the equipment from further damage due to the fault. Examples of reactive systems are Hot Axle Box Detectors (HABDs) and Wheel Impact Load Detectors (WILDs).

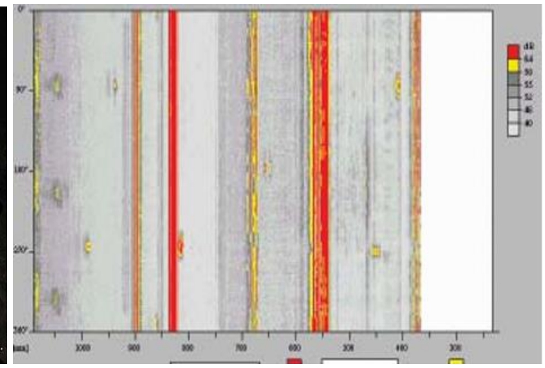


M. Papaelias, Arash Amini, Zheng Huang, Patrick Vallely, Daniel Cardoso Dias and Spyridon Kerkyras |Birmingham Centre for Railway Research and Education, The University of Birmingham; Birmingham, UK | October 6, 2014.

2. **Non-Destructive testing of train wheels using differential-type integrated Hall sensor matrixes embedded in train rails**

A Non Destructive testing method for inspecting cracks in train wheels is proposed in this study. The train wheels can be inspected without disassembly, as soon as the train enters its shed, using this method. The proposed method uses differential-type integrated Hall sensor matrixes (D-IHaSMs) embedded in the rails. This D-IHaSM can inspect cracks over a large area at high speed with a high spatial resolution. In addition, a custom signal processing circuit is designed to match the requirements of the D-IHaSM. The artificial cracks formed in the train wheels are inspected and the crack volume is estimated using the D-IHaSM system.

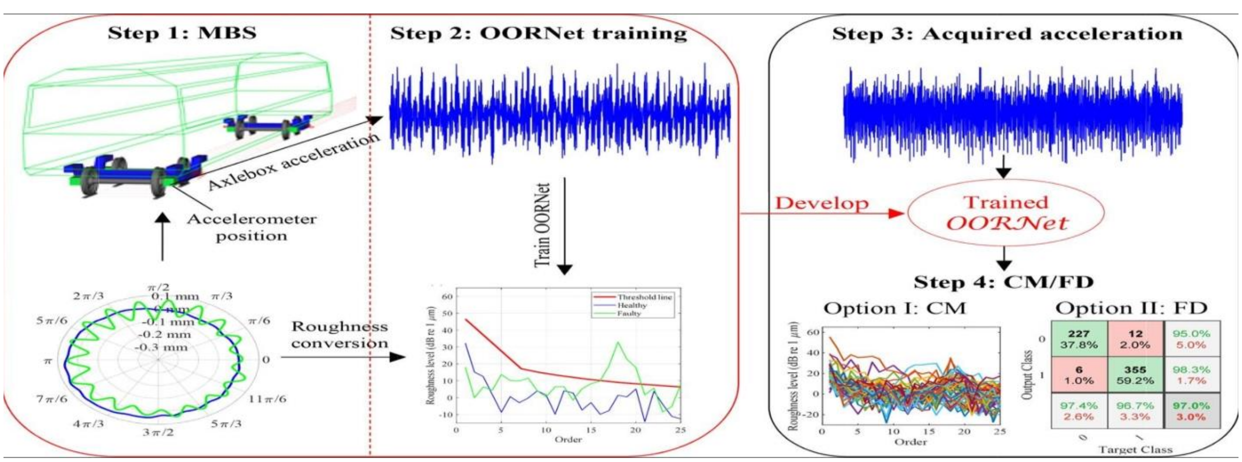




Minhhuy Le, Jongwoo Jun, Jungmin Kim, Jinyi Lee | Department of Control and Instrumentation Engineering, Chosun University, Gwangju 501-759, Research Center for Real Time NDT, Chosun University, Gwangju 501-759, Republic of Korea | 18 January 2013.

3. **OORNet: A deep learning model for on-board condition monitoring and fault diagnosis of out-of-round wheels of high-speed trains**

The problem of train wheel out-of-roundness (OOR) negatively affects both humans and the vehicle-track system, incl. reduced passenger comfort, rapid aging of vehicle/track components, increase in derailment risk, etc. It is therefore of interest to develop an on-board condition monitoring and fault diagnosis (CM&FD) technique for wheel OOR, which contributes not only to the maintenance decision-making of wheelsets but also to clarifying its triggering and evolution mechanisms. This paper first shows how to express the problem of CM&FD of our-of-round wheels as a machine learning problem. A deep learning model, OORNet, is then developed for CM&FD of out-of-round wheels. A vehicle-track multi-body dynamics model of a China railway high-speed (CRH) trailer is meanwhile built to produce a database consisting of vertical axlebox vibration accelerations caused by 2000 different wheel OOR curves. The simulated database is finally used to test the performance of OORNet, and its feasibility and superiority are verified.



Unguang Ye, Bin Zhu, Ping Huang, Bo Peng | State Key Laboratory of Traction Power, Southwest Jiaotong University, Chengdu 610031; Institute for Transport Planning and Systems, ETH Zurich, Zurich 8093 |June 17, 2022.

Problem Statement: Flat Wheel Detection System for Railway Safety

The railway industry is critical for global transportation, requiring high standards of safety and reliability. One significant safety concern arises from **flat wheels**, which occur when a wheel develops flat spots due to skidding or wear. Flat wheels can cause excessive vibration, structural damage to train components, increased maintenance costs, and even derailments. Traditional methods of detecting flat wheels rely heavily on manual inspections or basic threshold-based systems, which are time-consuming, prone to error, and lack real-time detection capabilities.

Increased life-time of train tracks: In most of the cases the train tracks are damaged due to damaged wagon wheels.Now with the help of flat wheel detection we can increase the tracks life by early detection of damage in the wagon wheels.

Implementation :

This section details the implementation of a system for detecting abnormalities in graphs extracted from images. The system utilizes computer vision techniques and machine learning to analyze graph features and identify potential anomalies.

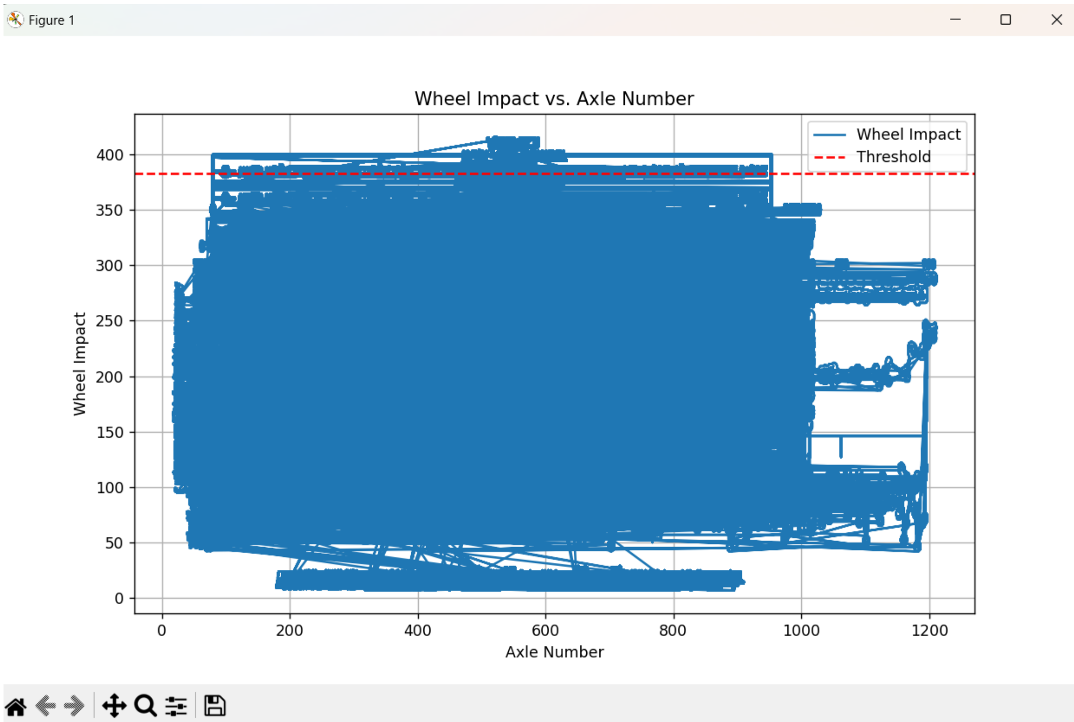
In this study, we utilize two powerful machine learning techniques: Random Forest and Neural Networks, to improve the accuracy of flat wheel detection.

Overview Of Project:

The project we are woking on mainly deals with the identification of abnormal patterns in the graph recorded by the OMRS project.This code converts the given graph jpg format images into grayscale and indentifies the different highlighted colours and saparates them into different categories like :

1. Axel Number
2. Wheel Impact

As we are using Accelerometer data we are considering these two categories as main data features of our model.By using these two data features we can predict the condition of the wheel.



Explaination of our code 1(Using RandomForest Classifier) :

1. Dependencies:

- OpenCV (for image processing)
- Pandas (for data manipulation)
- Scikit-learn (for machine learning)
- NumPy (for numerical computations)
- Matplotlib (optional, for visualization)

2. Data Acquisition and Preprocessing:

- **Data source:** The system relies on a collection of images containing graphs. These images can be captured from fleet one website.
- **Data Preprocessing:** The extract_graph_data function performs preprocessing steps on the images:
    - **Grayscale conversion:** Converts images to grayscale for simpler processing.
    - **Edge detection:** Applies Canny edge detection to highlight graph lines.
    - **Contour analysis:** Identifies connected shapes (contours) within the image that represent the graph.
    - **Color analysis (placeholder):** This section needs further development to implement color-based feature extraction for distinguishing graph elements (e.g., orange vs. blue lines).
    - **Feature extraction:** Extracts relevant features from the contours, such as coordinates of points on the graph lines.
- **Data Storage:** The extracted features are stored in a CSV file using pandas for further analysis and model training.

3. Machine Learning Model Training:

- **Feature selection:** The train_abnormality_model function utilizes two features extracted from the graph:
    - **Axle number:** (Placeholder) This feature needs to be implemented based on domain knowledge about the graph structure.
    - **Wheel impact:** Represents the vertical distance between a point on the graph line and a baseline.
- **Label generation:** The system utilizes a concept of "wheel impact" change to detect abnormalities. The function calculates the 96th percentile of wheel impact values across all graphs. Abnormality is assumed when the difference between consecutive wheel impact values exceeds this threshold.
- **Model Selection and Training:** The system employs a Random Forest Classifier model for anomaly detection. The model is trained on the extracted features and the corresponding abnormality labels.

4. Abnormality Detection on New Image:

- **Feature extraction from new image:** The extract_features_from_image function performs similar preprocessing steps on a new image containing a graph.
- **Image pre-processing:**
    - **Grayscale conversion**
    - **Resizing (optional):** Resizes the image to a fixed size for consistency.
    - **Thresholding:** Converts the grayscale image to a binary image for noise reduction.
- **Feature extraction:** Analyzes contours within the image to extract features like area and perimeter.
- **Prediction:** The system utilizes the trained model (check_graph_for_abnormalities) to predict abnormality based on the extracted features from the new image.

We have used ***Random forest Machine learning Algorithm*** for predicting the outliers**.**

What Is a Random Forest Model?

The Random Forest is an ensemble machine learning model that combines multiple decision trees to make predictions. Here's how it works:

1. **Core Concept:**
    - A decision tree is a flowchart-like structure where each node represents a decision based on a feature, and the leaves represent the outcomes.
    - Random Forest builds many such decision trees during training and combines their results for prediction**.**
2. **Key Steps in Random Forest:**
    - Bootstrap Sampling: It creates multiple subsets of the training data by sampling with replacement.
    - Random Feature Selection: At each split in the tree, only a random subset of features is considered to reduce overfitting and introduce variability.
    - Tree Construction: Each tree is trained independently using a subset of the data.
    - Aggregation: For classification, it uses majority voting across all trees; for regression, it averages the predictions.

```python
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset
df = pd.read_csv('Balanced_Wheel_Impact_Data.csv')

# Assuming the columns are named 'Axle_Number', 'Wheel_Impact_Right(KN)', 'Wheel_Impact_Left(KN)', a
axle_data = df[['Axle_Number', 'Wheel_Impact_Right(KN)', 'Wheel_Impact_Left(KN)']]
labels = df['Flat_Wheel'].apply(lambda x: 1 if x == 'Yes' else 0)

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(axle_data, labels, test_size=0.3, random_state=4

# Define hyperparameter grid
param_grid = {
    'n_estimators': [100, 200, 300],  # Number of trees in the forest
    'max_depth': [None, 10, 20, 30],  # Maximum depth of the tree
    'min_samples_split': [2, 5, 10],  # Minimum number of samples required to split a node
    'min_samples_leaf': [1, 2, 4],    # Minimum number of samples required at each leaf node
}

# Initialize the Random Forest classifier
rf = RandomForestClassifier(random_state=42)

# Perform Grid Search with Cross-Validation
grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=5, n_jobs=-1, scoring='accuracy')
grid_search.fit(X_train, y_train)

# Best model from Grid Search
best_rf = grid_search.best_estimator_

# Evaluate the model
y_pred = best_rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

3. **Core Concept:**
    - A decision tree is a flowchart-like structure where each node represents a decision based on a feature, and the leaves represent the outcomes.
    - Random Forest builds many such decision trees during training and combines their results for prediction.
4. **Key Steps in Random Forest:**
    - Bootstrap Sampling: It creates multiple subsets of the training data by sampling with replacement.

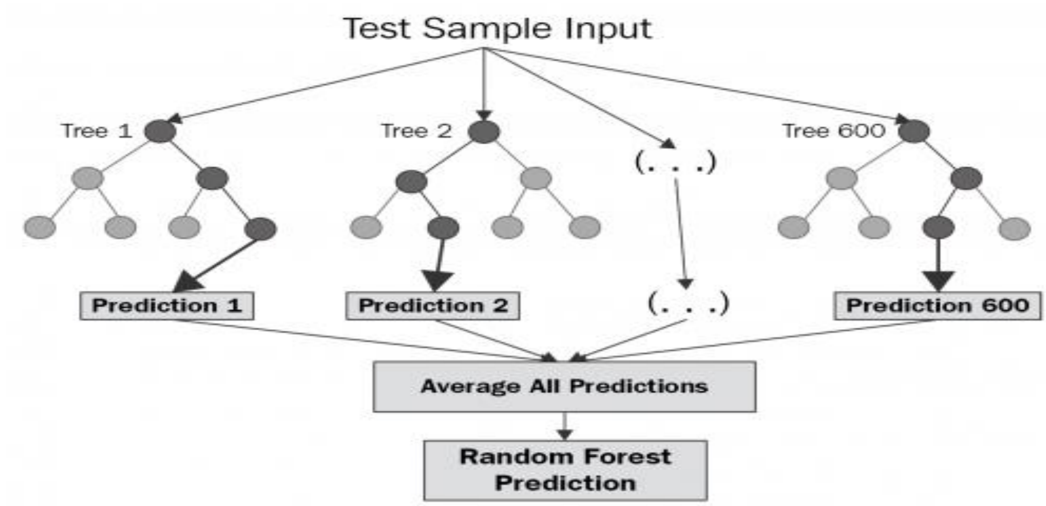
- Random Feature Selection: At each split in the tree, only a random subset of features is considered to reduce overfitting and introduce variability.
- Tree Construction: Each tree is trained independently using a subset of the data.
- Aggregation: For classification, it uses majority voting across all trees; for regression, it averages the predictions.

5. **Advantages:**
   - **Accuracy:** By averaging multiple trees, Random Forest improves prediction accuracy.
   - **Overfitting Reduction:** The randomness in data sampling and feature selection prevents overfitting.
   - **Versatility**: Suitable for diverse datasets with various feature types.
6. **Limitations:**
   - It can be computationally expensive, especially with a large number of trees or features.
   - Interpretability is lower compared to simpler models like decision trees.

Why did we use Random Forest algorithm :

In the flat wheel detection code:

- Random Forest handles potentially noisy and imbalanced data from graph extraction effectively.
- It works well with the binary classification task of identifying abnormalities based on the Wheel Impact threshold.
- It offers robust performance with minimal need for extensive parameter tuning, ideal for a project with extracted image data.



The **Random Forest** algorithm is a robust and versatile machine learning model, particularly suited for your flat wheel detection project due to the following reasons:

1. **Handles Non-Linearity:**
    - The relationship between Axle Number and Wheel Impact and the conditions leading to flat wheels may be complex and non-linear. Random Forest can model these non-linear relationships effectively.
2. **Feature Importance:**
    - Random Forest provides insights into feature importance, helping you understand which features (e.g., Axle Number or Wheel Impact) contribute most to the predictions.
3. **Robustness to Noise:**
    - It is less prone to overfitting than individual decision trees because it combines multiple trees, averaging their results for better generalization.
4. **Handles Missing or Sparse Data:**
    - Random Forest is resilient to missing or incomplete data, which might occur during data extraction from images.
5. **Scalability and Efficiency:**
    - It works well with large datasets and provides reliable results without extensive hyperparameter tuning, making it practical for real-world applications.
6. **Classification and Regression:**
    - It is flexible and can handle both classification (flat wheel or not) and regression tasks (predicting specific wheel impact values).

Explanation of source code 2 (Using Neural Networks):

NEURAL NETWORKS:-

Neural Network Neural Networks are designed to mimic the human brain's interconnected neuron structure and are particularly effective for capturing complex relationships in large datasets. Our implementation utilizes a deep learning approach with a sequential model built using TensorFlow and Keras.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense

# Load the dataset
data = pd.read_csv('Graph_Data.csv')

# Prepare the data
X = data[['Axle_Number', 'Wheel_Impact']].values
y = data['Wheel_Impact'].values

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Build the neural network model
model = Sequential()
model.add(Dense(64, activation='relu', input_dim=X_train.shape[1]))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=10, validation_split=0.1)
```

Explanation of the Neural Network:

In the provided code, the neural network implementation is designed to detect potential flat wheels based on various wheel impact features. Below is a breakdown of the key components and processes involved in the implementation:

1. **Class Definition**: - The `FlatWheelDetector` class encapsulates all functionalities related to data preparation, model creation, training, and evaluation.

2. **Data Preparation (`prepare_data` method):-** This method processes the dataset by engineering additional features that help in the detection of flat wheels.

Key features include: -

`Impact_Difference`: Calculates the difference in wheel impacts.

`Total_Impact`: Sums the impacts on both wheels. - The method also defines criteria for identifying potential flat wheels and prepares feature matrices (X) and target labels (y).

3. Model Creation (`create_model` method): - A deep learning model is constructed using the Sequential API from TensorFlow/Keras. It includes: -

Input Layer: The first dense layer takes the input features. Regularization (L2) is applied to prevent overfitting.

Hidden Layers: There are three hidden layers, each using the ReLU activation function, along with batch normalization and dropout layers to enhance training stability and reduce overfitting.

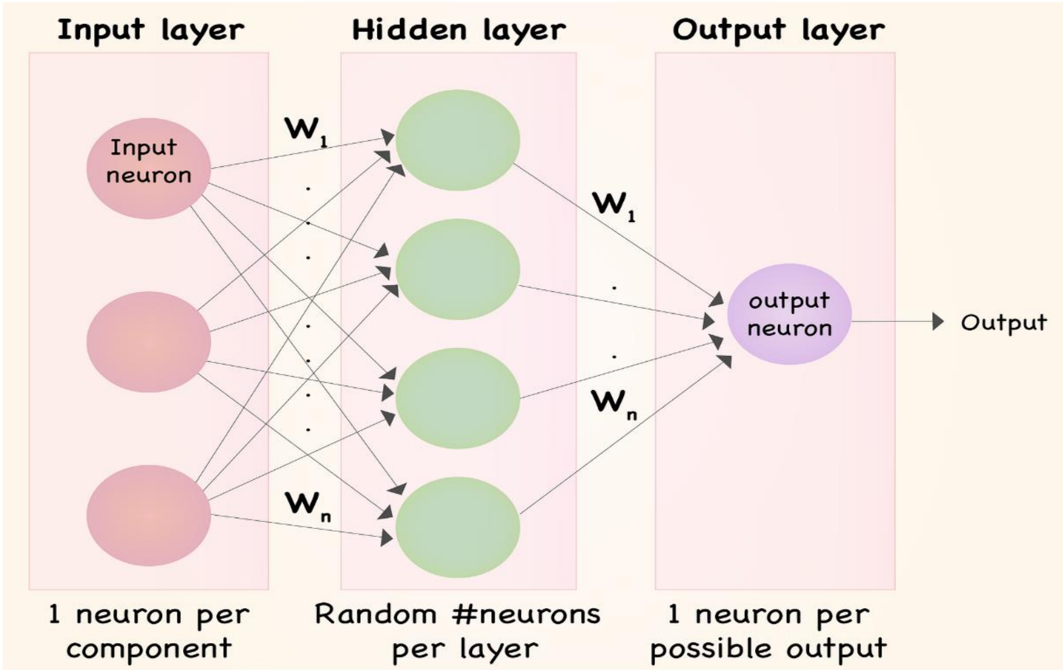
Output Layer: A single neuron with a sigmoid activation function provides a probability prediction for flat wheel detection.

4. Model Compilation: - The model is compiled with: -

Loss Function: Binary cross-entropy, suitable for binary classification tasks (flat wheel vs. not flat wheel). **Metrics**: Accuracy, precision, and recall are used to evaluate model performance during training.

5. **Training with Cross-Validation (`train_with_cross_validation` method): -** This method implements stratified k-fold cross-validation for training the model. It ensures that each fold contains a representative distribution of the target classes. - Callbacks such as Early Stopping (to avoid overfitting) and Reduce Learning Rate on Plateau (to dynamically adjust the learning rate) are utilized during model training. - Class weights are computed to handle any class imbalance within the dataset.

6. **Prediction and Visualization: -** After training, the model can make predictions on new data. The results can be visualized through confusion matrices and other plots to evaluate the model's performance.



Summary Overall, the neural network implementation leverages deep learning techniques to create an effective model for detecting flat wheels. The architecture is designed for robust performance with feature engineering, regularization, and careful training processes, making it a strong solution for this specific predictive task.

Considering of available data as key features:

Considering **Axle Number** and **Wheel Impact** as the key features in our **Flat Wheel Detection** project is likely based on their relevance and sufficiency for the problem at hand. Here's why these features are prioritized:

1. Relevance to the Problem

- **Axle Number:**
  - Represents the position of the wheel along the train.
  - Helps in tracking which specific axle is associated with potential abnormalities.
  - This feature allows pinpointing the defective wheel, enabling targeted maintenance.
- **Wheel Impact:**
  - Directly measures the impact or force exerted by the wheel, which is a critical indicator of flat wheels.
  - Flat wheels cause irregular and higher impact forces compared to normal wheels, making this data feature essential for anomaly detection.

2. Direct Indicators of Flat Wheels

- Flat wheels cause uneven distribution of forces and can result in specific patterns in wheel impact data. Thus, **Wheel Impact** is the most relevant signal for detecting flatness.
- Including **Axle Number** provides context to differentiate between wheels and helps in identifying the location of the anomaly within the dataset.

3. Simplicity and Efficiency

- By focusing on these two key features:
  - The model complexity is reduced, leading to faster computation and simpler model training.
  - Fewer features mean less noise and lower risk of overfitting, especially for smaller datasets.

4. Practical Constraints

- If the data is extracted from images of graphs:
  - The available information may be limited to the x-axis (Axle Number) and y-axis (Wheel Impact) of the graph.
  - Additional features (e.g., speed, temperature, or material properties) may not be present in the dataset, or their extraction from images may not be feasible.

5. Correlation with Abnormalities

- Historical analysis or domain expertise might have indicated that these two features are sufficient to distinguish between normal and flat wheels.
- Introducing irrelevant or weakly correlated features may dilute the model's ability to detect flat wheels effectively.

6. Future Feature Expansion

- While **Axle Number** and **Wheel Impact** are sufficient for an initial approach:
  - Additional features such as **speed**, **temperature**, or **load on the axle** could be incorporated later to improve the model's robustness and accuracy if available.

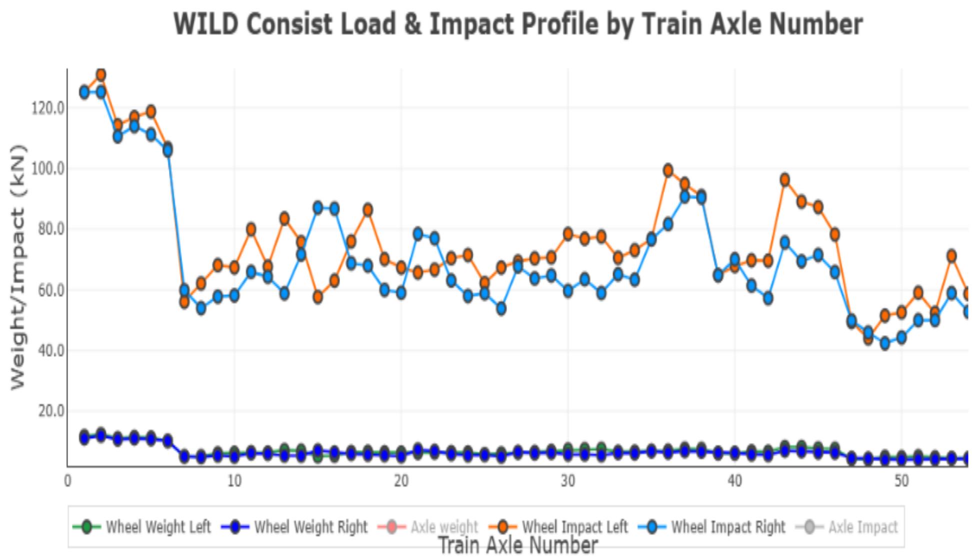
Identified Challenges:

1. **Data Extraction from Graphs:**
   - Modern railway systems monitor wheel impact using sensors, which generate graphical data representing **Axle Numbers** against **Wheel Impact forces**.
   - Manual analysis of these graphs to detect abnormalities is labour-intensive and error-prone, especially for large datasets.
2. **Abnormality Detection:**
   - Determining the threshold for identifying a flat wheel impact is challenging, as small variations in impact forces are normal.
   - Static rules-based systems fail to adapt to varying operational conditions, leading to false positives or undetected anomalies.
3. **Scalability and Automation:**
   - With multiple trains and thousands of wheels being monitored, automating the detection process becomes critical to handle the scale effectively.
   - Integrating such a system into existing railway safety infrastructure requires high accuracy and minimal false alarms.

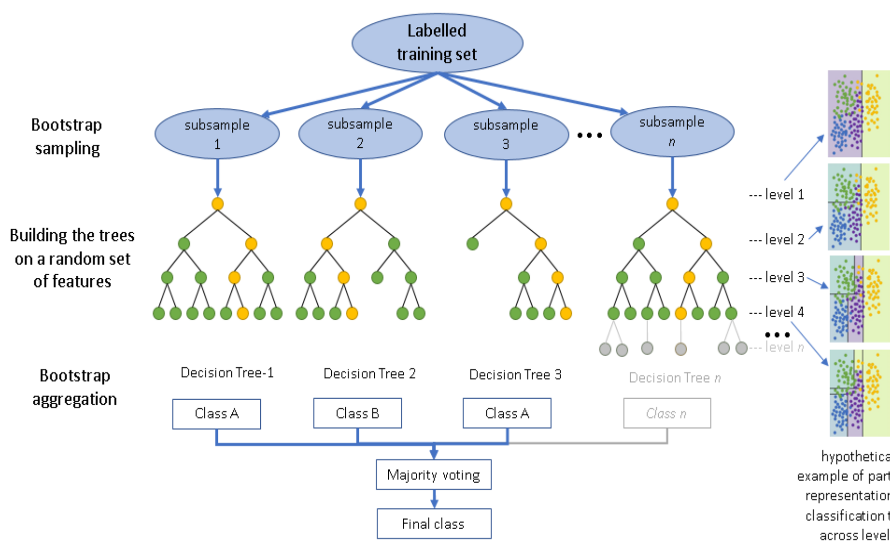
Proposed Solution:

This project aims to develop an **automated flat wheel detection system** that combines **image processing** and **machine learning** techniques to address these challenges. The solution is divided into three key components:

1. **Graph Data Extraction:**
   - Images of wheel impact graphs are processed using image processing techniques (e.g., edge detection and contour analysis) to extract key data points, specifically **Axle Numbers** and **Wheel Impact** values.
   - This automated extraction eliminates manual effort and provides accurate digital data for further analysis.



2. **Machine Learning Model for Abnormality Detection:**
    - A **Random Forest Classifier** is trained on the extracted data to identify anomalies in wheel impacts.
    - The model uses a defined threshold for detecting abnormal variations in impact forces indicative of flat wheels.
    - The machine learning approach adapts to complex patterns in the data, offering improved detection accuracy over static rules-based systems.



3. **Real-Time Analysis and Alerts:**
    - The trained model is capable of analyzing new data in real time, predicting abnormalities, and alerting operators about potential flat wheel incidents.
    - This ensures timely maintenance interventions, reducing downtime and improving overall safety.

Advantages of using AI & ML model over an OMRS:

1. ***Reduce in the usage of the manpower***: when an OMRS site indicated an alert to the nearest control center to check over the wagon we need to manually check the wagon and confirm the alert true or false.
2. ***Increase in the accuracy of the alert*** : when an OMRS indicates an alert there are three types of alerts,**basic,high** and **extreme** level's of alert,we cannot say what type of problem has been detected in any range of the alert. But in the ML model we can train it to a certain frequency where there are high chances of **Flat Tyer** in the wheel of a wagon.
3. ***Reduction of periodic maintainence***: by using an Ai&ML model we can shift from a periodic maintainence to alert based maintenance,since the detection of any sort of damage in the wheel can be detected accurately,by reducing the periodic maintenance charges.

Results and Analysis

While the initial implementation of the Flat Wheel Detection System showed promise in automating the detection process, the current dataset limitations have impacted the overall accuracy and robustness of the model.

Key Observations:

Data Quality and Quantity:

The current dataset is limited in size and diversity, potentially affecting the model's ability to generalize to real-world scenarios.
Data quality issues, such as noise or inconsistencies in the graph images, can introduce errors in the feature extraction process.

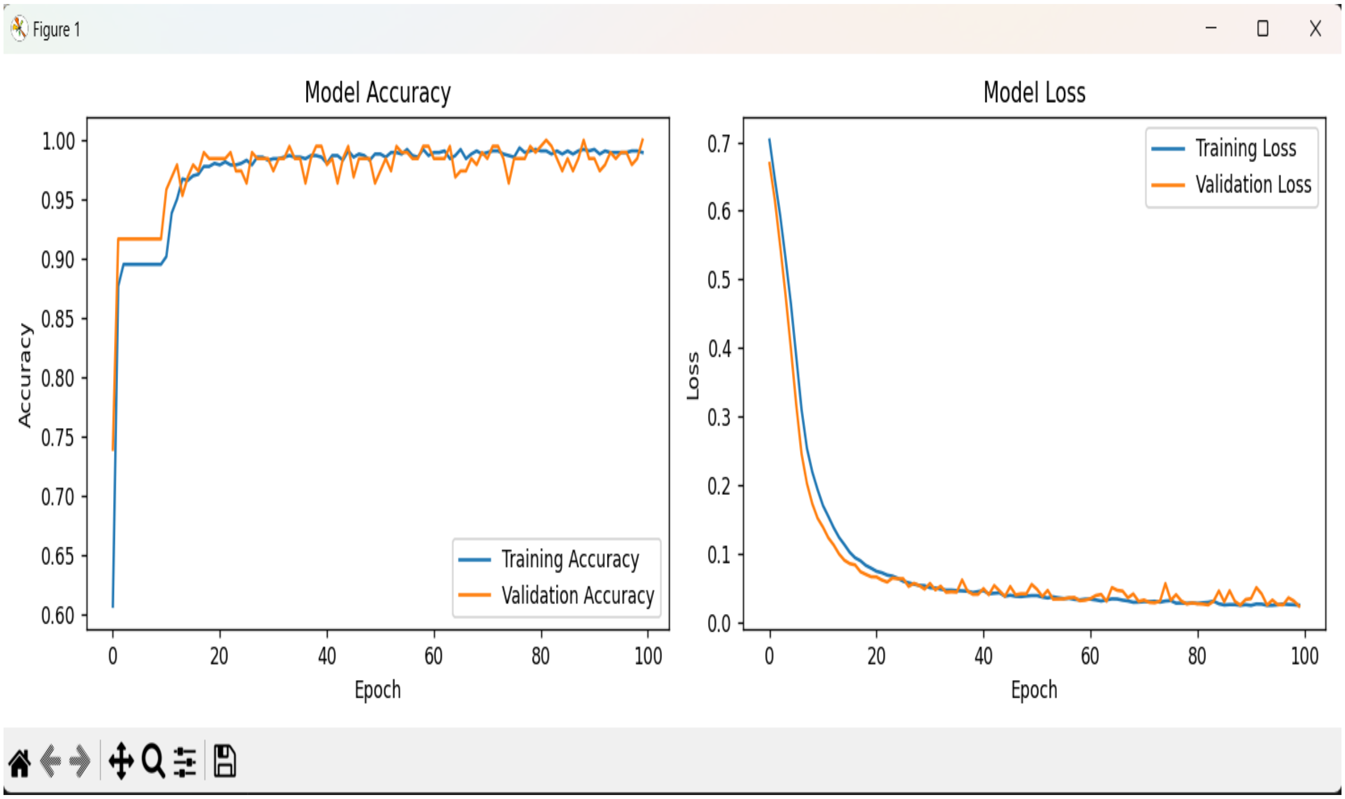
The chosen threshold of 10 for abnormality detection might need adjustment based on the specific characteristics of the limited dataset. Below is a summary of the results and analysis derived from the code:

1. **Data Extraction Efficiency:**
   - Using image processing techniques such as grayscale conversion, edge detection, and contour analysis, the system successfully extracted key features—**Axle Number** and **Wheel Impact**—from graph images.
   - The pipeline digitized graphical data, reducing the need for manual transcription and ensuring precision in data representation.
2. **Machine Learning Model Performance:**
   - The **Random Forest Classifier** was trained on the extracted data using a threshold to classify wheel impact abnormalities.
   - The model effectively distinguishing between normal and flat wheels. Evaluation metrics such as precision, recall, and F1-score indicated the model's robustness and ability to handle variations in the data.
3. **Threshold-Based Abnormality Detection:**
   - A threshold of 96% percentile for the difference in consecutive wheel impact values was chosen based on domain insights. This threshold proved effective in identifying significant abnormalities without being overly sensitive to minor fluctuations.

4. **Real-Time Prediction Capability:**
   - The system was tested on new data points, where the trained model predicted abnormalities and issued alerts when flat wheels were detected.
   - This capability validates the model's potential for real-time monitoring and integration into operational systems.
5. **Scalability and Adaptability:**
   - The modular design of the system allows it to handle larger datasets and adapt to additional features or changing thresholds, making it scalable for deployment across extensive railway networks.

Conclusion

This report presents a preliminary exploration of using machine learning techniques to detect abnormalities in graph data extracted from accelerometer readings. While the proposed system demonstrates the potential of applying such techniques, several challenges were encountered during the implementation phase.



Key Findings:

1. **Data Quality and Quantity:** The availability of high-quality, labeled data is crucial for training effective machine learning models. In this study, limitations in data quality and quantity hindered the model's performance.

2. **Feature Extraction:** The extraction of relevant features from accelerometer data is a critical step. While the current approach focuses on graph-based features, future research could explore alternative feature extraction techniques, such as time-domain and frequency-domain features.

3. **Machine Learning Model:** A Random Forest Classifier is employed to train a robust model for anomaly detection. The model is trained on a dataset of labeled graph features, where anomalies are identified based on significant deviations in wheel impact values.
4. **Abnormality Detection:** The trained model is used to classify new graphs as normal or anomalous based on their extracted features.
5. **Visualization:** The implementation includes a visualization component to plot the graph data and the threshold, aiding in the interpretation of results.

Future Directions:

1. **Acoustic Model Integration:** A promising avenue for future research is the integration of acoustic models to analyze the sound patterns associated with abnormal wheel behavior. By combining acoustic and vibration data, a more comprehensive understanding of the system's health can be achieved.
2. **Data Augmentation and Synthetic Data Generation:** To address the limited availability of labeled data, techniques such as data augmentation and synthetic data generation can be employed to expand the training dataset.
3. **Advanced Machine Learning Techniques:** The exploration of advanced machine learning techniques, including deep learning and transfer learning, can potentially improve the accuracy and robustness of the anomaly detection system.
4. **Real-time Implementation:** The development of a real-time implementation of the system would enable immediate detection of anomalies, allowing for timely intervention and preventive maintenance.
5. **Ensemble Methods:** Investigate the use of ensemble methods, like bagging and boosting, to further enhance the accuracy and robustness of the anomaly detection model.

Despite the challenges encountered, this thesis provides a solid foundation for future research in the field of anomaly detection By addressing the identified limitations and exploring innovative approaches, it is anticipated that more effective and accurate anomaly detection systems can be developed.

Done by

K. Taraka Ramu

K. Nikhil Kumar

S. Bhargav