# Behavioral Analysis and Risk Assessment of Two-Wheeler Drivers

Submitted in partial fulfillment of the requirements of the degree

**BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING**

By

**Simran Ahuja 02**

**Jesica Bijju 10**

**Sejal Datir 14**

**Sania Khan 36**

Name of the Mentor

**Prof.** Dr. Mrs. Nupur Giri

V.E.S.

Since 1962

**Vivekanand Education Society's Institute of Technology,**

**An Autonomous Institute affiliated to University of Mumbai**

**HAMC, Collector's Colony, Chembur,**

**Mumbai-400074**

**University of Mumbai (AY 2023-24)**

# CERTIFICATE

This is to certify that the Mini Project entitled **" Behavioral Analysis and Risk Assessment of Two-Wheeler Drivers"** is a bonafide work of **Simran Ahuja(02), Jesica Bijju(10), Sejal Datir(14) and Sania Khan(36)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

**(Prof.__________________)**

Mentor

**(Prof.__________________)**

Head of Department

**(Prof.__________________)**

Principal

# Mini Project Approval

This Mini Project entitled "Behavioral Analysis and Risk Assessment of Two-Wheeler Drivers " by **Simran Ahuja(02), Jesica Bijju(10), Sejal Datir(14) and Sania Khan(36)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

**Examiners**

**1……………………………………….**

(Internal Examiner Name & Sign)

**2……………………………………….**

(External Examiner name & Sign)

Date:

Place:

**INDEX**

## Abstract

This project focuses on the driving behavior of two wheeler riders, who are particularly vulnerable to road accidents. Due to the absence of a surrounding frame and airbags, even minor collisions can have severe consequences for these riders. Through the analysis of time-series data collected from real-world driving scenarios, this study aims to develop machine learning models for risk assessment and anomaly detection. The paper starts by outlining the alarming statistics of road accidents in India, emphasizing the urgent need for effective solutions and mitigating the risk of accidents. The study explores the effectiveness of five distinct models: Long Short-Term Memory -Residual (LSTM-R), Gated Recurrent Unit (GRU), Adaboost , XGBoost, Multilayer Perceptron (MLP) with Random Forest (RF) ensemble and compares the performance of the given models for 9 parameters that include Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch of the bike.

## Acknowledgments

We extend our heartfelt gratitude to Vivekanand Education Society's Institute of Technology for their unwavering support and assistance throughout our project. In particular, we are deeply appreciative of Professor Dr. Mrs. Nupur Giri for her invaluable guidance and advice during the project's inception and development.

We are indebted to Dr. (Mrs.) Nupur Giri,  the Head of the Computer Department, Mr. Yogesh Pandit, and Dr. (Mrs.) J.M. Nair, our Principal, for granting us this valuable opportunity to undertake this project. Their trust in us has been instrumental in our success.

Our sincere thanks go to all the teaching and non-teaching staff whose constant encouragement and selfless assistance were instrumental in completing both the project synopsis and the project review successfully. We also acknowledge the Department of Computer Engineering for their support and guidance.

We are profoundly thankful to everyone who contributed to our project by providing information and resources. Our families also deserve recognition for their unwavering moral support and encouragement throughout this journey.

# Chapter 1 : Introduction

## 1.1 Introduction

In recent years, India has witnessed an escalation in traffic accidents. The Ministry of Road Transport and Highways' annual report for the fiscal year 2021–2022 shows a startling statistic: there were 4,12,432 road accidents registered nationwide, resulting in 1,53,972 fatalities and 3,84,448 injuries. According to NH road accident and fatality data, two-wheelers accounted for the most number of accidents (52,416) and deaths (22,786) in 2021 [1]. The leading cause of more than half of these accidents was overspeeding, with dangerous and careless driving following closely. These alarming statistics highlight the importance of developing effective solutions to reduce road accidents and elevate road safety standards.

In response to this critical and escalating public health challenge, our research aims to address the urgent need for enhanced road safety measures, especially for two-wheeler riders. Through careful examination of the actions and habits of two-wheeler drivers, our research endeavors to uncover the factors contributing to accidents.

## 1.2 Motivation

With the growth of various start-ups promising 10 minute deliveries the strain of these targets is obviously borne by the delivery partners, who often drive hastily to meet the demands. The primary transportation used by these delivery partners is 2 wheelers. In 2019, the Mumbai Traffic Police had to summon the executives of delivery partners like Swiggy, Zomato, Uber-Eats to address the issue of rash and negligent driving by the delivery drivers who are perpetually in a tearing hurry to meet their targets. Moreover, maintenance of such huge fleets of vehicles is also a challenge. Thus, to address these concerns behavior analysis of these driving patterns is the need of the hour with primary focus on 2 wheelers.

## 1.3 Problem Statement

The project aims to develop a comprehensive behavioral analysis and risk assessment AI-model using time-series data for 2-wheeler motorbikes that would incorporate parameters to identify abnormal driving behaviors. Hence, help improve road safety, and create a safer environment for all road users in India.

## 1.4 Existing Systems

Several existing systems and technologies aim to address road safety concerns and analyze driving behavior. These systems include:

1. **Telematics Systems:** Telematics systems utilize GPS technology and onboard diagnostics to monitor vehicle performance and driver behavior. These systems can track parameters such as speed, acceleration, braking, and location in real-time, providing insights into driving habits and vehicle condition.

2. **Dashboard Cameras:** Dashboard cameras, commonly known as dashcams, record video footage of the road ahead and inside the vehicle. They can capture incidents such as accidents, near misses, and driver behavior. Some dash cams also feature accelerometers and gyroscopes to detect sudden movements or impacts.

3. **Black Box Recorders:** Similar to those used in aircraft, black box recorders for vehicles store data on speed, acceleration, braking, and other parameters. In the event of an accident, this data can be retrieved and analyzed to determine the cause and contributing factors.

4. **Driver Monitoring Systems:** These systems use sensors and cameras to monitor the driver's behavior, including eye movement, head position, and fatigue. They can alert drivers to dangerous behaviors such as drowsiness or distraction.

5. **Mobile Applications:** There are numerous mobile applications available that aim to improve road safety by providing features such as speed limit alerts, journey tracking, and driving behavior analysis. Some applications also offer gamification elements to encourage safe driving practices.

## 1.5 Lacuna of the Existing Systems

Despite the availability of these systems, there are still gaps in effectively addressing road safety challenges, particularly for two-wheeler riders. Some of the shortcomings include:

1. **Limited Focus on Two-Wheeler Safety:** Many existing systems are designed primarily for four-wheeled vehicles and may not adequately address the unique challenges faced by two-wheeler riders, such as balancing, maneuverability, and vulnerability to impacts.

2. **Data Analysis Complexity:** While data collection technologies have advanced, analyzing the vast amounts of data generated by these systems can be complex and time-consuming. There is a need for more efficient methods of extracting meaningful insights from driving data.

3. **Integration and Compatibility:** Integrating different systems and technologies into a cohesive solution can be challenging, particularly in heterogeneous environments with a mix of vehicle types and communication protocols.

4. **Cost and Accessibility:** Some advanced safety systems may be prohibitively expensive for individual riders or small-scale operators, limiting their adoption and impact on overall road safety.

## 1.6 Relevance of the Project

Given the increasing number of road accidents and fatalities, especially involving two-wheeler riders, there is a pressing need for innovative solutions that can effectively analyze driving behavior, identify risk factors, and ultimately improve road safety. This project addresses these challenges by leveraging machine learning algorithms and time-series data analysis to develop a comprehensive behavioral analysis and risk assessment model specifically tailored for two-wheeler motorbikes. By focusing on key parameters and abnormal driving behaviors, the project aims to contribute to the creation of a safer environment for all road users in India.

# Chapter 2 : Literature Survey

## A. Brief Overview of Literature Survey

The literature survey conducted for this research project highlights a comprehensive array of studies addressing driving behavior analysis and safety measures, particularly focusing on two-wheeler riders. These studies encompass various methodologies, including machine learning algorithms, smartphone sensor data analysis, and systematic literature reviews, to understand and mitigate the risks associated with two-wheeler transportation. Notably, the research papers explore diverse aspects such as event detection, abnormal driving behavior identification, and driver behavior classification. Despite the advancements in driver assistance systems, a noticeable gap exists in providing tailored, proactive solutions specifically designed for two-wheeler safety. Existing systems primarily cater to car-centric features, leaving two-wheeler riders vulnerable on the roads. Furthermore, while AI technology is prevalent in contemporary systems, its full potential remains untapped for two-wheeler applications, particularly in comprehensive diagnostics, risk assessment, and behavioral analysis. Thus, there is a pressing need for innovative, AI-driven solutions aimed at proactively enhancing two-wheeler safety to reduce road accidents and improve overall road safety standards.

## B. Related Works

### 2.1 Research Papers Referred

| SR NO. | Title Of The Paper | Abstract |
|---|---|---|
| 1 | Driving-Pattern Identification and Event Detection Based on an Unsupervised Learning Framework: Case of a Motorcycle-Riding Simulator | This paper suggests a method to study how people drive in a detailed way. They have a process with several steps. The main step uses computer programs to figure out how people usually drive and to find unusual or risky driving moments. When these risky moments are found, they are analyzed further by looking at the most important aspects using graphs. |

| | | |
|---|---|---|
| 2 | (PDF) Detection of Two wheeler Driver Safety Using Machine Learning | In this paper a method for spotting people who break the law by not wearing a helmet while riding a bike. This new tool can also help the police find these rule-breakers even in tough weather conditions. These experiments show how well it can recognize the bikers and identify the violations. A system was also proposed that could adapt to different situations with a few adjustments if needed. |
| 3 | Safety of motorised two wheelers in mixed traffic conditions: Literature review of risk factors - ScienceDirect | This research looks at other studies that have examined the things that make motorized two-wheelers (MTWs) less safe, especially when they're riding in diverse traffic without clear lanes. The goal of this paper is to gather the findings from these studies and emphasize the latest insights. The paper also talks about what we still don't know and need to research more about to improve the safety of MTWs in mixed traffic. |
| 4 | D 3 : Abnormal driving behaviors detection and identification using smartphone sensors | This paper focuses on enhancing driving safety by monitoring and identifying specific abnormal driving behaviors using smartphone sensors. It can pinpoint actions like weaving, swerving, and sudden braking. The system, called "D3," extracts unique patterns from smartphone sensor data and employs machine learning to identify these behaviors with an impressive 95.36% accuracy, as shown in real-world tests. It aims to improve drivers' awareness and help prevent accidents. |
| 5 | https://www.academia.edu/23985750/Detecting_Powered_Two_Wheeler_incidents_from_high_resolution_naturalistic_data | This paper focuses on how motorcycle riders (Powered-Two-Wheeler or PTW drivers) change their behavior during risky situations. The paper suggests a method to identify these changes by looking at detailed driving data. It finds irregular behavior by spotting values that stand out in the data. These irregularities can be seen as critical moments (incidents) that relate to common driving situations. |

| | | |
|---|---|---|
| 6 | Driver Behavior Classification System Analysis Using Machine Learning Methods | This paper is used to classify driver behavior based on data from lane detection and traffic conditions. It tested various models and found that gradient boosting works best. It also analyzes important features for classification and suggests future improvements, considering factors like speed limits and mental workload, and a hybrid classification system. |
| 7 | Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals - ScienceDirect | This paper introduces LSTM-R, a real-time algorithm for detecting abnormal driving behavior. It's tested using smartphone-collected vehicle data and outperforms other methods, even with limited abnormal driving data. This approach is cost-effective and can improve road safety by assessing driving risk and behavior. |
| 8 | CNN-LSTM Driving Style Classification Model Based on Driver Operation Time Series Data | This paper presents a method for classifying driving styles using data from a simulator environment. A CNN+LSTM network is trained to detect driving styles, which shows high generalization and cost-efficiency when tested with real car data. The system can signal drivers and surrounding vehicles to improve driving route planning. |
| 9 | Driver Behavior Classification: A Systematic Literature Review | This review studied driver behavior classification systems from 2015 to 2022, finding 93 relevant studies. Field driving studies are common, and driver behavior categories include abnormal and aggressive behavior. Machine learning algorithms, especially SVM, LR, and LSTM, dominate this field. |
| 10 | Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring | This paper presents SenseFleet, a mobile driver profiling and scoring app. It uses sensors and GPS to detect driving events and considers context like weather and time. Experimental results show its accurate detection of risky driving and driver differentiation. Future work includes refining calibration and fuzzy set definitions. |

## 2.4 Comparison with the Existing System

In the landscape of driver assistance systems, a noticeable gap exists when it comes to providing comprehensive and proactive solutions tailored specifically for two-wheeler riders. Many existing systems are primarily designed for cars, leaving a void in holistic support for two-wheeler safety. These systems, while proficient in their respective tasks, often function reactively rather than proactively. Instead of assisting riders in avoiding potential risks, they typically alert them after detecting hazards. Furthermore, these systems may offer features such as navigation or vehicle diagnostics, but they often lack the integration of multimodal datasets, which could combine camera data with information from other sensors to enhance safety.

AI technology, although prominent in contemporary systems, is not fully harnessed for two-wheeler applications, especially concerning comprehensive diagnostics, risk assessment, and behavioral analysis. Furthermore, these systems rarely provide specific alerts tailored to two-wheelers regarding hazardous road conditions, such as wet or slippery surfaces. Unlike car-centric solutions, which include advanced fleet management tools, two-wheeler-specific solutions are less developed and may lack advanced functionalities. Maintenance tracking, typically offered as a standalone feature, is often disconnected from real-time diagnostics and AI-driven risk analysis.

# Chapter 3: Requirement Gathering for the Proposed System

## 3.1 Introduction to requirement gathering

The requirement gathering phase of our project on driving behavior analysis of two-wheeler riders is pivotal for delineating the objectives, scope, and constraints of the study. This phase involves meticulous interaction with stakeholders to identify and document the essential features, functionalities, and data requirements for developing machine learning models. Through interviews, surveys, and data collection from real-world driving scenarios, we aim to elucidate the driving behaviors contributing to road accidents. By engaging closely with domain experts and stakeholders, we strive to ensure alignment with the overarching goal of enhancing road safety measures, particularly for vulnerable two-wheeler riders. Transparency, flexibility, and responsiveness characterize our approach during this phase to accommodate evolving needs and expectations effectively.

By monitoring the rider's driving patterns and comparing them to a database of safe driving behaviors, the system can assess the risk associated with the rider's style. Over time, it can identify tendencies that might lead to accidents or violations and offer corrective feedback. For example, if a rider frequently brakes hard or takes turns aggressively, the system might provide tips or tutorials to encourage safer habits.

## 3.2 Functional Requirements

Functional requirements in our project delineate the specific behaviors and capabilities of the machine learning models developed for risk assessment and anomaly detection in two-wheeler driving behavior. These requirements elucidate the features and interactions expected from the models, such as analyzing accelerometer and gyro rotation data to detect abnormal driving patterns. Through detailed documentation and validation techniques, including use cases and performance metrics, we specify the system's functionalities to ensure alignment with stakeholder needs and project objectives. Functional requirements serve as the blueprint for model design, development, and validation, guiding the implementation of features to mitigate road accident risks effectively.

## 3.3 Non-Functional Requirements

Non-functional requirements in our project outline the quality criteria and operational characteristics essential for the effectiveness and reliability of the machine learning models. These requirements encompass performance, reliability, security, usability, and scalability aspects critical for ensuring the models' success in real-world applications. Examples include response time, model accuracy, data privacy measures, user interface design, and compliance with regulatory standards. By addressing non-functional requirements, we aim to enhance the overall quality, efficiency, and user experience of the developed models, thereby contributing to improved road safety standards.

The aim of this project is to analyze the behavior of two-wheeler riders and make predictions regarding the normalcy or abnormality of their driving patterns. This section outlines the methodologies and procedures employed to achieve this objective.

## 3.4 Hardware, Software , Technology and tools utilized

**Hardware:**

- Sensors for capturing vehicle data.

**Software:**

- Python (Tensorflow, Keras, Sklearn)
- Google Colab

## 3.5 Constraints

Constraints inherent in our research paper encompass factors such as time limitations, data availability, and computational resources. The limited availability of real-world driving data and the complexity of anomaly detection algorithms pose challenges during model development. Furthermore, resource constraints, including budgetary limitations and hardware limitations, influence the scope and scalability of the study. By acknowledging and managing these constraints proactively, we aim to optimize resource utilization, mitigate risks, and deliver meaningful insights into driving behavior analysis, despite the inherent limitations and challenges posed by the research context.

# Chapter 4 : Proposed Design

## 4.1 Block diagram of the system

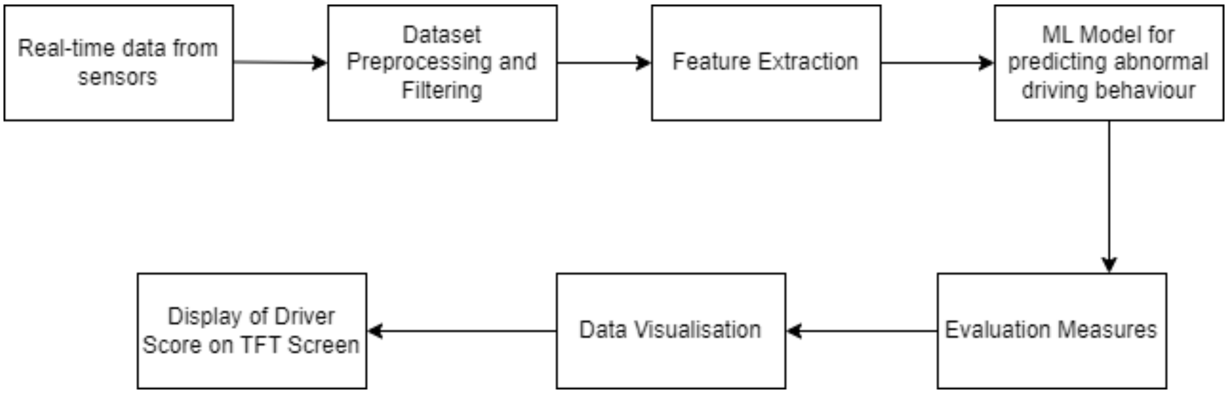



*Fig. 1: Block diagram*

## 4.3 Detailed Design

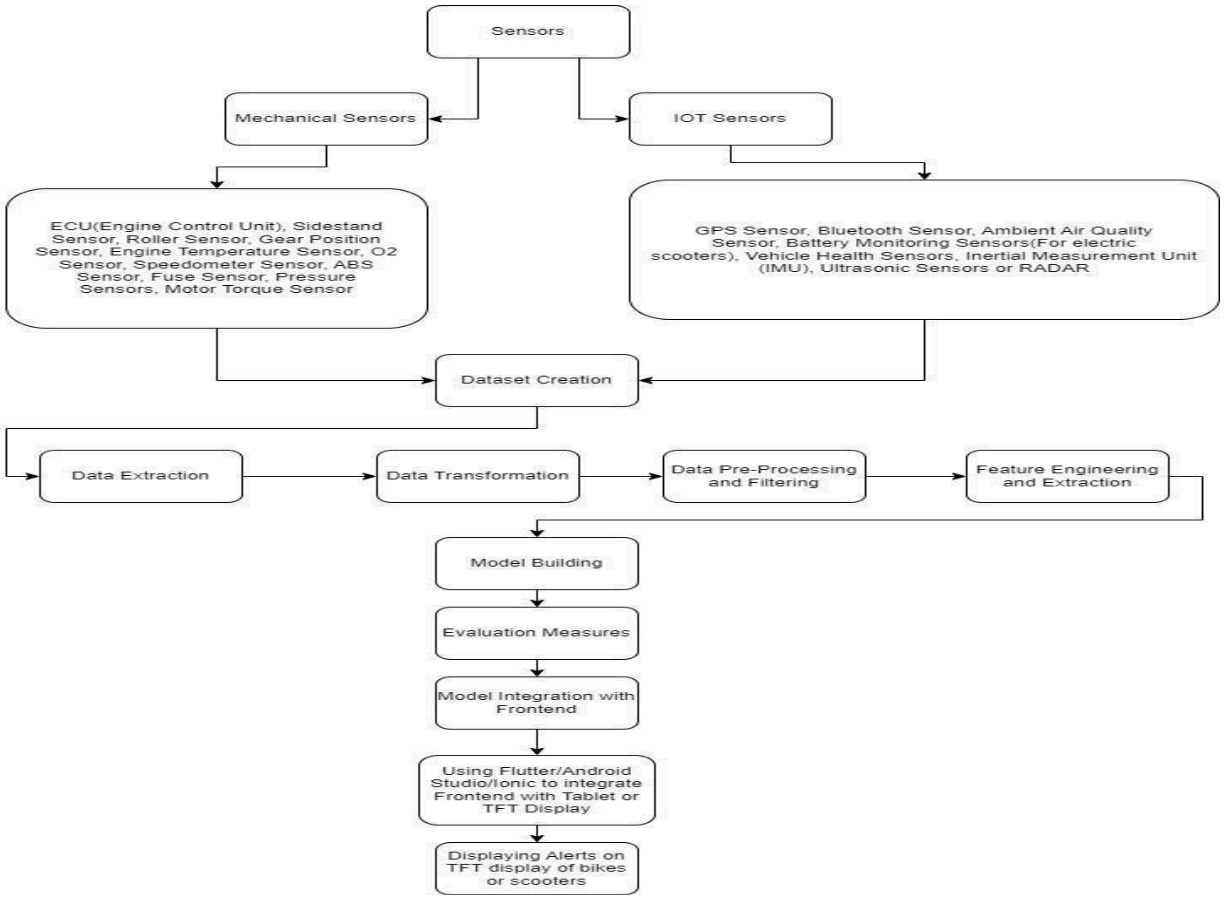



*Fig. 2: Detailed design*

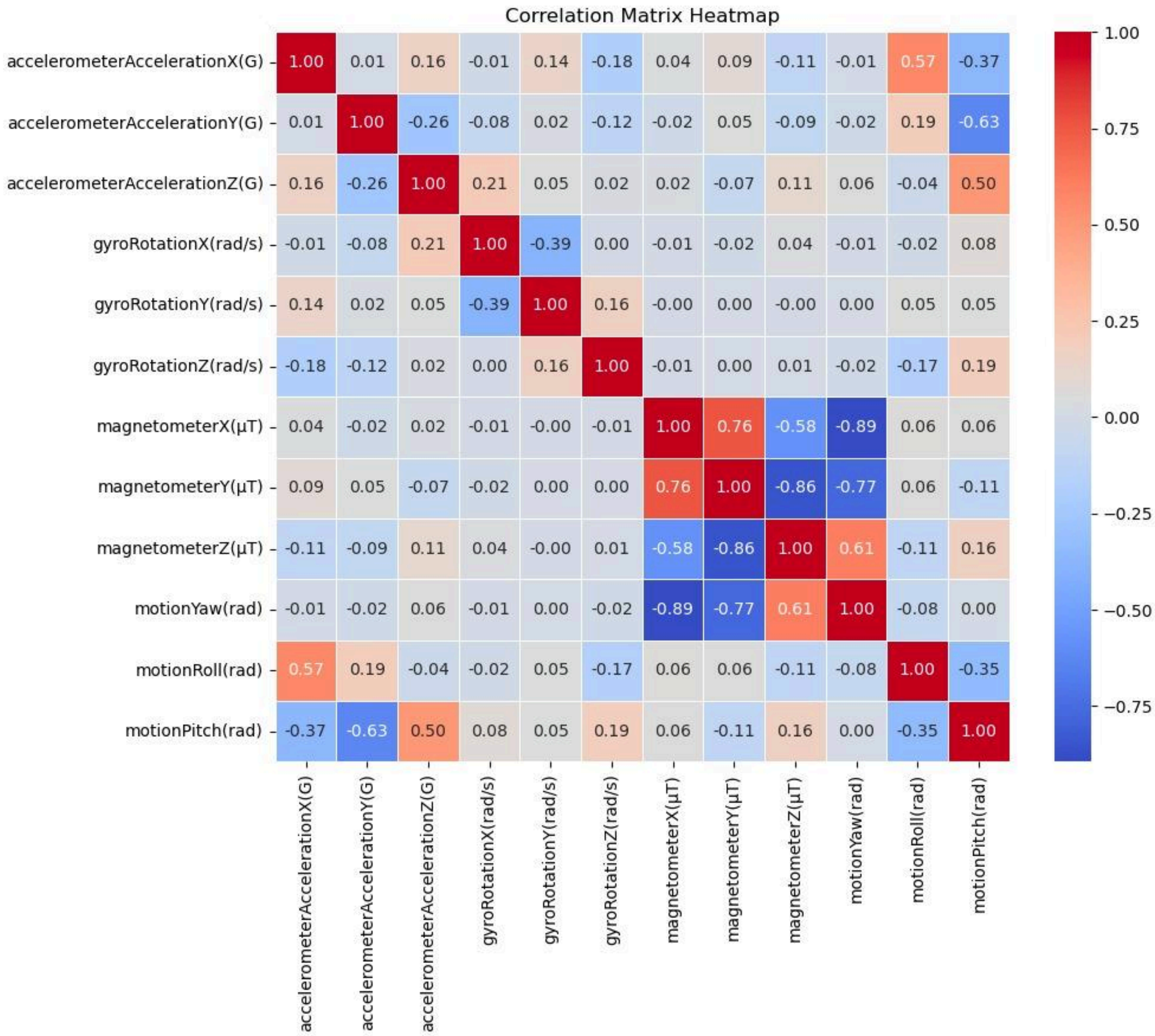
# Chapter 5 : Implementation of the Proposed System

## 5.1 Methodology employed for development

1. A series of test drives was conducted over a 4 km route in Chembur, Mumbai, utilizing Sensorlog, an iOS application, to collect vehicle kinematic data in real time with a sampling frequency of 10 Hz. The data collection process involved two individuals: a rider and a passenger equipped with a handheld iOS mobile device running the Sensorlog app. The application continuously recorded the rider's movements and behavior, capturing parameters including Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch of the bike. Additionally, Magnetometer readings (X, Y, Z) were initially included but later excluded due to redundancy. The data collection occurred over a period of 7 days, during which 81,480 tuples of data were gathered, with each ride averaging 20 minutes in duration. The dataset encompassed data collected from both normal driving behavior and rash driving behavior exhibited on the same route.

2. Preprocessing:
Before fitting the model, the input features are reshaped for all models. The reshaping is performed to flatten the input features, converting the 2D array into a 1D array, making it compatible with the model's expectations. The dataset was split into training, validation, and test sets.

3. Model selection: Five models were trained on the nine parameters. Neural network models- LSTM-R and GRU were trained first, followed by the boosting algorithms- AdaBoost and XGBoost. The final model that was trained was a combination of both neural networks and boosting- the ensemble of MLP with RF.




*Correlation Matrix*

Fig. 1 depicts the correlation between the recorded parameter

## 5.2 Algorithms and flowcharts for the respective modules developed

Five models were trained on the nine parameters. Neural network models- LSTM-R and GRU were trained first, followed by the boosting algorithms- AdaBoost and XGBoost. The final model that was trained was a combination of both neural networks and boosting- the ensemble of MLP with RF. The model creation process involves the training of five models: long short-term memory-residual (LSTM-R), gated recurrent unit (GRU), AdaBoost, XGBoost, and ensemble of multilayer perceptron (MLP) with random forest. The dataset is split into three parts: 70 % for training, 15 % for testing , and 15 % for validation. The input layer is structured to accept data in windows of a specific size.

### 5.2.1 LSTM - Residual

#### 5.2.1.1 LSTM

LSTM is a type of recurrent neural network (RNN) that analyzes sequential data [ 12]. LSTM's ability to capture temporal dependencies made it a suitable choice for the two wheeler time-series dataset used in this study. In an LSTM network, there are three gates (input, forget, and output gates) that regulate the flow of information, and the memory cell stores long-term dependencies[25].

#### 5.2.1.2 LSTM-Residual (LSTM-R)

The LSTM-R model calculates residuals, which are the differences between the observed data and the data predicted by the model. It uses a time window-based approach, with the residual values computed over a window of 100 milliseconds. The LSTM units act as intelligent memory cells, recognizing patterns across multiple time steps as data patterns evolve over various temporal scales.

In this model, layers are organized sequentially which begins with an input layer to handle sequences with nine features. The core is an LSTM layer with 64 units, designed to capture patterns over time. After that, a Dense layer with eight units and Rectified Linear Unit (ReLU) activation is added for non-linear transformations. The final layer is another Dense layer with nine units, using a linear activation.

### 5.2.2 GRU

Following the training of LSTM-R, Gated Recurrent Units (GRU) was chosen for its simpler architecture. GRU has two gates: reset and update gates as opposed to LSTM which has three gates[23]. GRU is a type of RNN like LSTM with a potential for faster training[13 ]. This choice aims to explore the strengths of both models and improve performance on the time series dataset. The structure of the GRU model was determined through repeated testing and comparison of the number of hidden layers, activation functions, number of units in each layer, batch size, dropout rate and epochs.

The model incorporated three GRU hidden layers with decreasing units 64, 32, and 16 respectively. This structure further includes two dense layers to map the extracted features from the hidden layers to the output space with 9 units each that uses the Rectified Linear Unit(ReLU) and the other that uses the Leaky Rectified Linear Unit(LeakyReLU) activation function. The addition of dropout layers as a regularization technique was avoided since GRU layers already have internal mechanisms in the form of reset and update gates to control the information flow and mitigate vanishing gradient problems, thereby avoiding overfitting. [24] For model optimization and training, we utilized Mean Absolute Error

(MAE) as the loss function, paired with the Adam optimizer set at a learning rate of 0.001.

### 5.2.3 Adaboost

After training the model with neural networks like LSTM and GRU, the dataset was trained on boosting algorithms to see how they perform in comparison to neural networks for the multi output regression task. The Adaboost model follows the boosting ensemble learning method. It makes use of 'adaboost regressor' as the weak learner and Decision Trees as the default base estimator [ 26 ]. It is then followed by Weighted Training where weights are assigned to each training instance. Initially, all weights are set equally and during the training process, the weights are adjusted based on the performance of the weak learners. The final prediction is a weighted sum of the predictions from individual weak learners.
Grid Search CV was employed to discover the optimal hyperparameters for the model, utilizing a 'param grid' dictionary[ 28 ]. Notably, the grid focused on three key hyperparameters: number of estimators (number of weak learners), learning rate, and loss function. The best parameters identified through this process were the number of estimators equal to 100, learning rate of 0.01, and loss set to 'exponential.' After training, the Adaboost Regressor is used to make predictions on the training, validation, and test sets. The Root Mean Square Error (RMSE) values are calculated for each parameter and Mean Absolute Error (MAE) is calculated for each set by comparing the predicted values to the actual labels. This metric is used to evaluate the model's performance on different subsets of the data.

### 5.2.4 XGBoost

It was observed that the predictions did not fit well with the actual data points. This observation prompted a transition to XG-Boost, a gradient boosting algorithm, known for its sequential tree construction [29 ]. XGBoost offers distinct advantages, including effective regularization methods and enhanced control over finetuning—features absent in Adaboost [16]. The XGBoost model adopts a structured approach, processing data in specific windows. This strategy proves effective in isolating and characterizing driving events based on their duration, providing valuable contextual insights for analysis and intervention [ 30 ].
Central to XGBoost's functionality is its foundation in gradient boosting, a methodology that optimizes a loss function through the calculation of gradients concerning the model's predictions; this approach allows the algorithm to make corrections that minimize overall prediction errors. Hyperparameters were experimented with to fine-tune the configuration of

the XGBoost model. The objective was to identify the optimal settings that would minimize the Root Mean Square Error (RMSE) values, ensuring the model's ability to make accurate and reliable predictions. Specifically, the selected key hyperparameters were- learning rate set at 0.2, maximum depth of 3 for each individual tree, alpha value of 10 for regularization strength , and 100 decision trees in the ensemble.

#### 5.2.5 Ensemble of Multilayer Perceptron(MLP) and Random Forest

After a separate analysis of neural networks and boosting algorithms, training a model on the combination of both could offer a different perspective.

##### 5.2.5.1 Multilayer Perceptron Neural Network

The Multilayer Perceptron has a wide variety of classification and regression applications in many fields: pattern recognition, voice and classification problems. [18] A general network consists of a layered architecture, an input layer, one or more hidden layers and an output layer [31]. A common artificial neural network used to address a variety of issues, such as pattern recognition and interpolation, is the Multilayer Perceptron (MLP) [32][33].

##### 5.2.5.2 Random Forest

The random forest algorithm is commonly used to handle regression and classification problems [ 20]. It can be defined as a regression technique that combines the performance of numerous decision trees and then averaging their predictions for the final outcome [22]. Each tree in the random forest is created based on a random subset of the input variables, and each tree branching is created based on a random number of selected input features. [11]

##### 5.2.5.3 Ensemble of MLP with Random Forest

The Ensemble model was created using the combination of Multi-Layer Perceptron as the neural network, and Random Forest for the boosting. A voting ensemble employs multiple models instead of a single one to enhance the system's performance, as discussed in [17 ]. The Voting Regressor, which combines regression models of MLP and RF by assigning weights to each model. The MLP network was trained with the ReLU activation function and the Adam optimizer. After trial and error, the best neural network architecture was obtained with 3 hidden layers with 100, 50, and 25 neurons in each hidden layer respectively. Other hyperparameters include an L2 regularization term of 0.0001, a constant learning rate of 0.001, and a batch size that dynamically adjusts during training. The RandomForestRegressor consists of 200 decision trees, employing the Mean Squared Error criterion. The ensemble model, created using VotingRegressor, combines predictions from both models. The model's evaluation involves assessing its performance on both training and test datasets using metrics such as Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

## 5.3 Datasets source and utilization

The dataset utilized in this study was systematically collected over a series of test rides covering a 4 km route in Chembur, Mumbai. The data collection process leveraged the SensorLog iOS mobile application, which was installed on a handheld device carried by both the rider and a passenger during the test drives. The application recorded vehicle kinematic data in real-time at a sampling frequency of 10 Hz.

Data Collection Process:

- Route and Distance: The rides were conducted over a consistent 4 km route in Chembur, Mumbai.
- Duration and Frequency: Each ride lasted approximately 20 minutes, with data recorded at a frequency of 10 Hz, resulting in a substantial volume of data points.
- Sensor Parameters Recorded: Various parameters were recorded during the rides, including Accelerometer readings (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch of the bike.
- Exclusion of Redundant Data: Initially, Magnetometer readings (X, Y, Z) were included but later excluded due to redundancy in the captured information.

Data Volume and Collection Period:

- Total Data Tuples: A total of 81,480 tuples of data were collected over the course of 7 days of data collection.
- Data Variability: The dataset encompasses a diverse range of driving behaviors, capturing both normal driving and instances of rash driving.

# Chapter 6 : Testing of the Proposed System

## 6.1. Introduction to testing

Testing is a crucial phase in the development of any system, ensuring its functionality, reliability, and performance meet the desired standards. In the context of behavioral analysis and risk assessment of two-wheeler drivers, rigorous testing becomes imperative to validate the effectiveness of the proposed models in identifying anomalies in driving behavior. This section outlines the testing procedures employed to evaluate the performance of five distinct models: LSTM-Residual, GRU, Adaboost, XGBoost and the ensemble of Multilayer Perceptron (MLP) and Random Forest (RF). Each model underwent testing on unseen data to assess its ability to detect deviations from expected driving patterns. The analysis of test results, including the examination of residuals, provides insights into the models' accuracy and efficacy in identifying driving anomalies.

## 6.2. Types of tests Considered

Various tests can be considered to evaluate the performance and robustness of the proposed machine learning models for two-wheeler driver behavior analysis.

1. Unit Testing:
    - Test individual components or functions of your machine learning models, such as the preprocessing steps, feature extraction, and individual model layers.
    - Ensure that each unit performs as expected and handles edge cases or invalid inputs correctly.
2. Performance Testing:
    - Test the computational performance and efficiency of your models, including training time, inference time, and resource utilization (e.g., memory and CPU usage).
    - Evaluate the models' scalability and ability to handle large volumes of data or real-time scenarios.
3. Regression Testing:
    - After making changes or updates to your models, perform regression testing to ensure that existing functionality is not adversely affected.
    - This can include testing with previously used datasets or test cases to verify consistent behavior.

4. Accuracy and Robustness Testing:
    - Test the accuracy and robustness of your models using various test case scenarios, as mentioned in your report (e.g., normal driving, rash driving, environmental conditions, sensor noise).
    - Evaluate the models' ability to generalize, handle edge cases, and maintain performance under different conditions.

## 6.3 Various test case scenarios considered

Test cases scenarios that could be considered for evaluating the performance and robustness of your machine learning models for two-wheeler driver behavior analysis:

1. Normal Driving Scenarios:
    - Test the models with data representing normal driving patterns, such as smooth acceleration, gentle turns, and appropriate braking.
    - Evaluate the models' ability to correctly identify these scenarios as normal behavior.
2. Abnormal Driving Scenarios:
    - Test the models with data representing various types of abnormal or rash driving behaviors, such as sudden acceleration, hard braking, and aggressive cornering.
    - Assess the models' ability to accurately detect and classify these abnormal behaviors.
3. Environmental Conditions:
    - Test the models with data collected under different environmental conditions, such as wet or slippery roads, strong winds, or varying light conditions.
    - Assess the models' ability to adapt and maintain accurate predictions in these varying conditions.

# Chapter 7 : Results and Discussion

## 7.2. Performance Evaluation measures

**1. Root Mean Squared Error (RMSE):**

The RMSE is a popular metric used to evaluate the performance of regression models. It represents the square root of the average squared difference between the predicted values and the actual values. The squaring of the errors in the RMSE formula gives more weight to larger errors, making it sensitive to outliers. The RMSE has the same units as the target variable, which makes it easier to interpret.

A lower RMSE value indicates better model performance, as it means the predicted values are closer to the actual values. However, the interpretation of a "good" RMSE value depends on the context and the scale of the target variable.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

**2. Mean Squared Error (MSE):**

The MSE is the average squared difference between the predicted values and the actual values. It is calculated by squaring the residuals. The squaring operation in the MSE formula amplifies the effect of larger errors, making it sensitive to outliers. This property of the MSE makes it useful for situations where large errors are particularly undesirable.

Like the RMSE, lower MSE values indicate better model performance. However, the MSE is not as interpretable as the RMSE because it is expressed in squared units of the target variable.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

**3. Residuals:**

Residuals are the differences between the actual values and the predicted values for each data point. They represent the errors made by the model for each individual prediction.

$$\mathbf{Residual} = \mathbf{y} - \hat{\mathbf{y}}$$

Residuals are useful for:

1. Assessing the goodness of fit: A good regression model should have residuals that are randomly distributed around zero, with no discernible patterns or trends. If the residuals exhibit patterns or trends, it may indicate that the model is missing important variables or that the assumptions of the regression are violated.
2. Identifying outliers and influential data points: Large residuals can indicate the presence of outliers or influential data points that may be affecting the model's performance.
3. Diagnosing model assumptions: Residual plots can be used to check for violations of assumptions, such as homoscedasticity (constant variance of errors) or normality of errors.
4. Identifying areas for model improvement: By analyzing the patterns in the residuals, you can identify areas where the model is performing poorly and potentially make improvements to the model.

Anomaly Detection- When the absolute value of a residual is larger than a predefined threshold, it may indicate that the corresponding data point is an anomaly or outlier. The threshold can be determined based on the distribution of the residuals or domain knowledge.

## 7.3. Input Parameters / Features considered

1. Accelerometer (X, Y, Z): Measures the acceleration forces along the three perpendicular axes (X, Y, and Z) due to the motion or gravity.
    - X: Measures the acceleration force along the horizontal left-right axis.
    - Y: Measures the acceleration force along the vertical up-down axis.
    - Z: Measures the acceleration force along the depth front-back axis.
2. Gyro Rotation (X, Y, Z): Measures the rate of rotation around the three perpendicular axes (X, Y, and Z) using a gyroscope sensor.
    - X: Measures the rate of rotation around the horizontal left-right axis.
    - Y: Measures the rate of rotation around the vertical up-down axis.
    - Z: Measures the rate of rotation around the depth front-back axis.

3. Motion Yaw: Measures the angle of rotation around the vertical axis, representing the heading or direction of the object's motion.
4. Motion Roll: Measures the angle of rotation around the front-to-back axis, representing the tilt or roll of the object's motion.
5. Motion Pitch: Measures the angle of rotation around the side-to-side axis, representing the up-and-down tilt or pitch of the object's motion.

## 7.4. Graphical and statistical output

### 7.4.1 LSTM- R

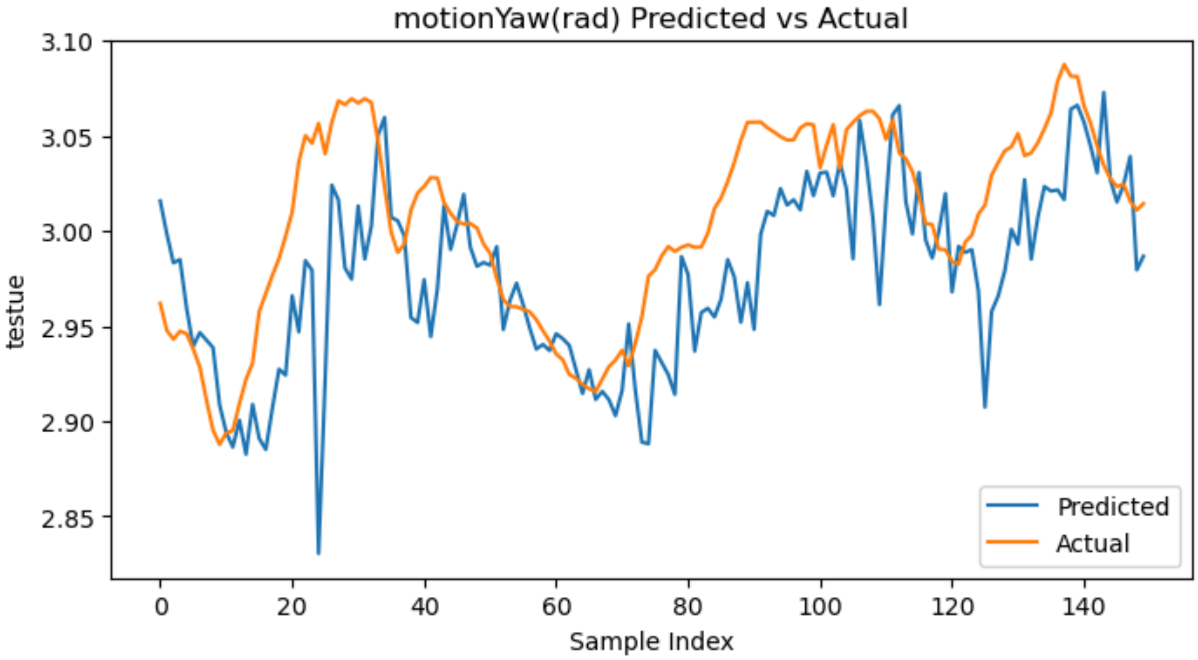



Fig. 2 : Motion Yaw Predicted vs Actual

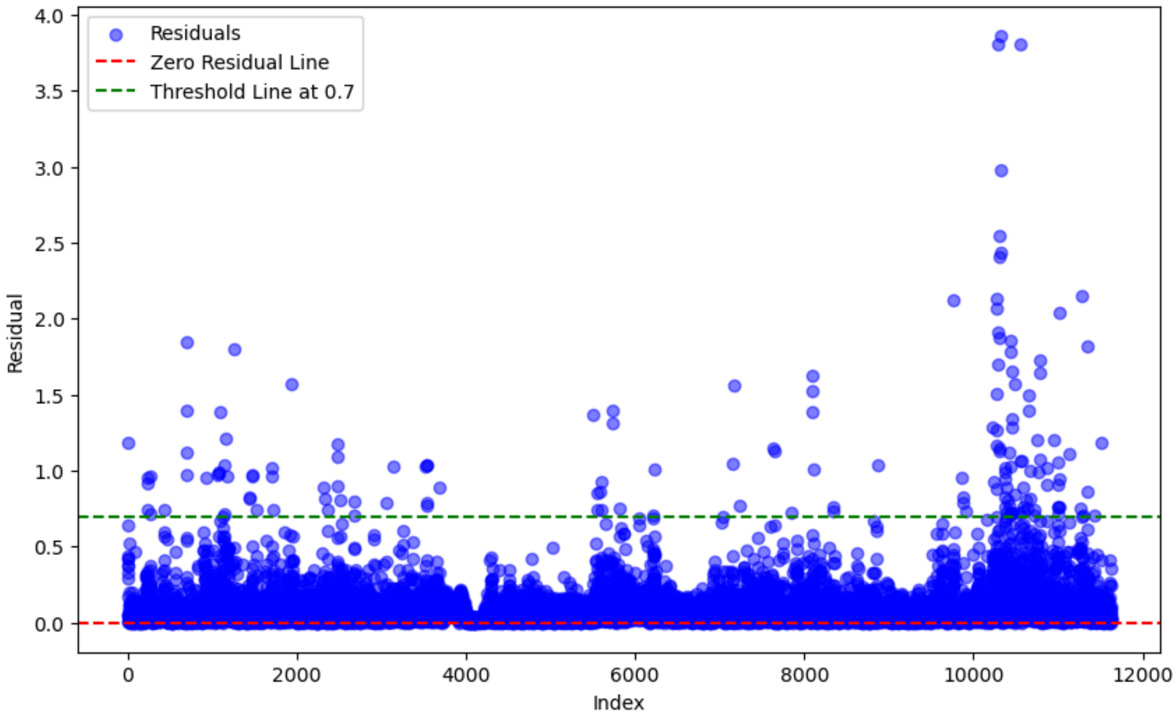



Fig. 3 : Residual Plot

### 7.4.2 GRU

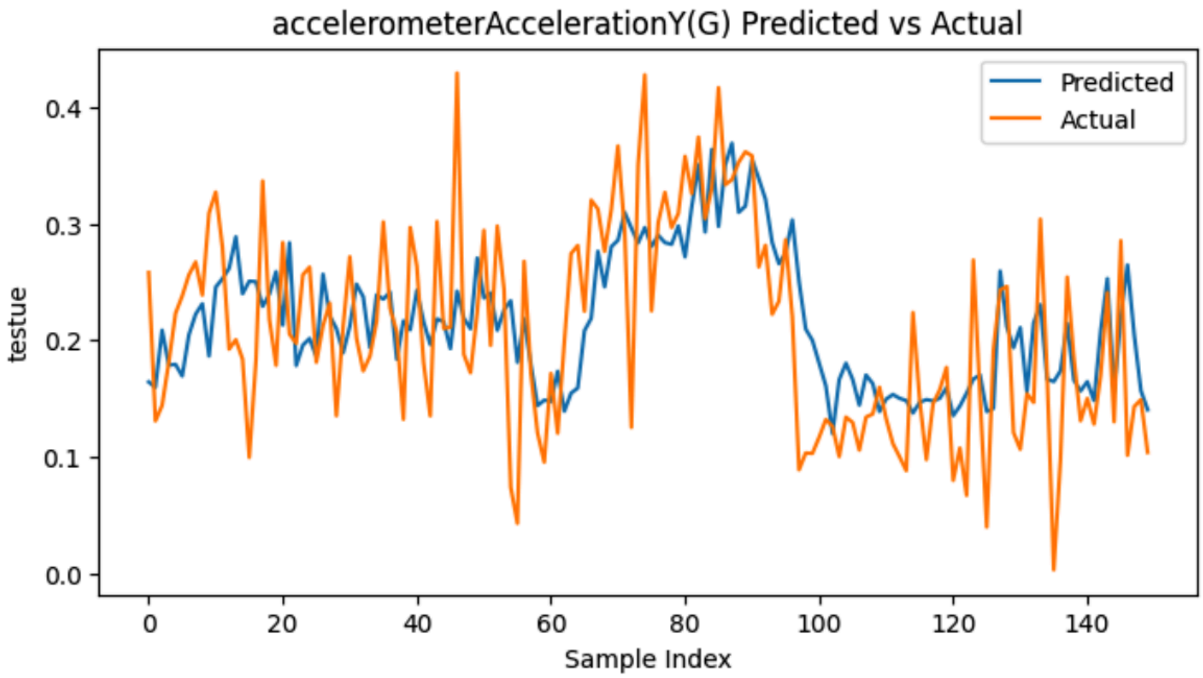



Fig. 4 : AccelerometerY Predicted vs Actual

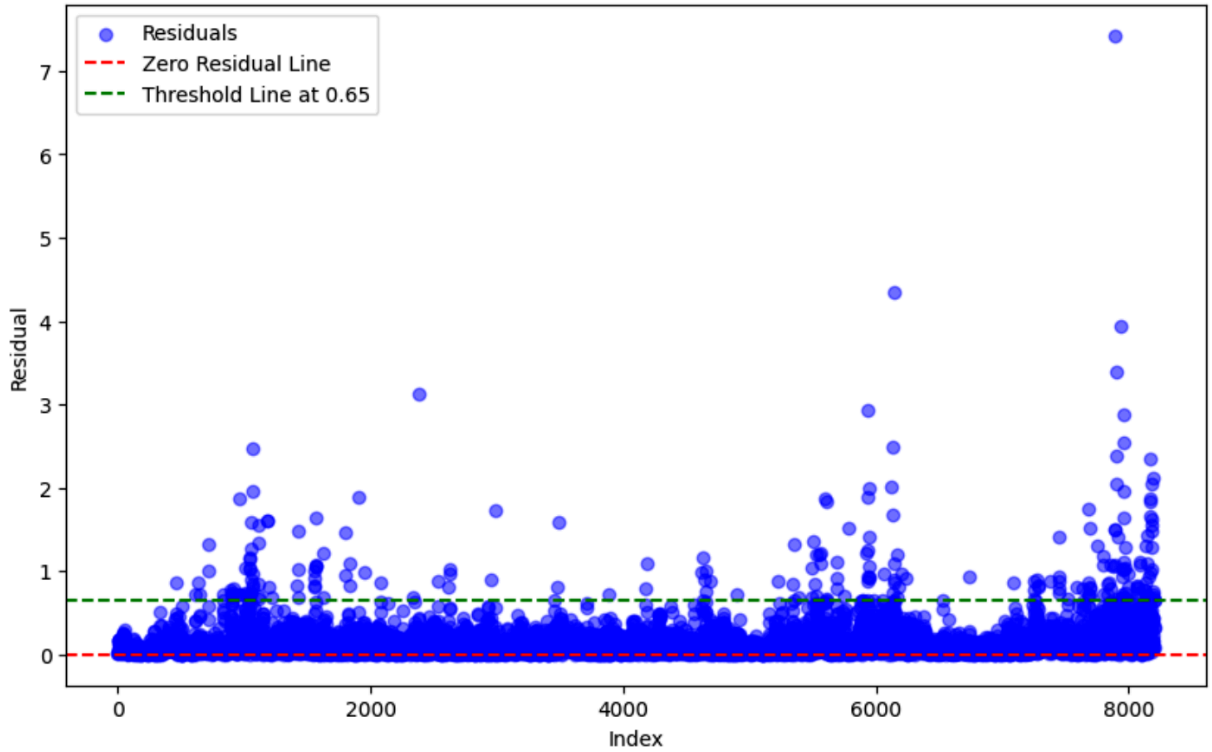



Fig. 5 : Residual Plot

### 7.4.3 AdaBoost

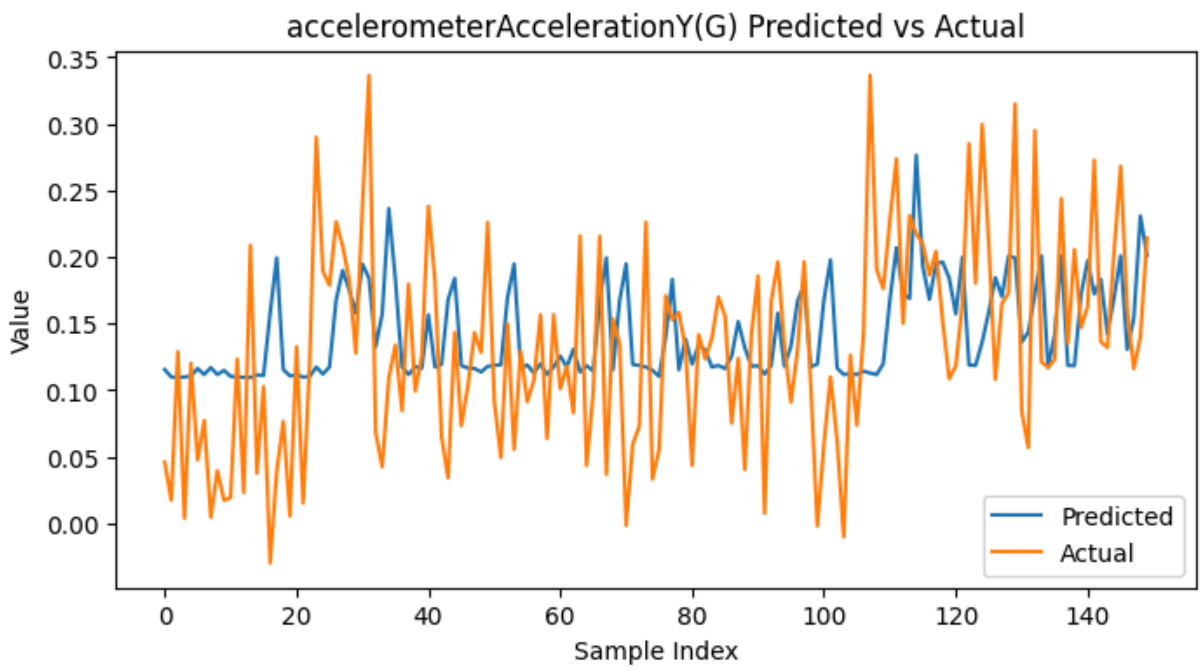



Fig. 6 : Accelerometer(Y) Predicted vs Actual

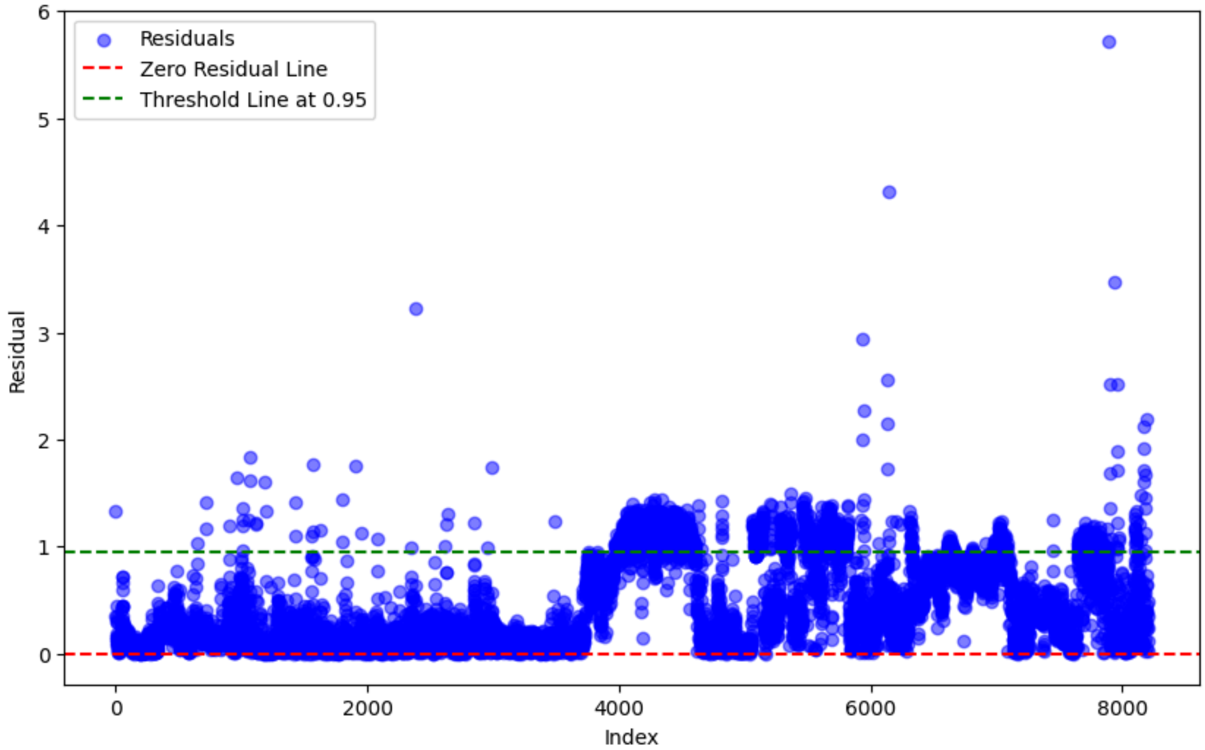



Fig. 7 : Residual Plot

### 7.4.4 XGBoost

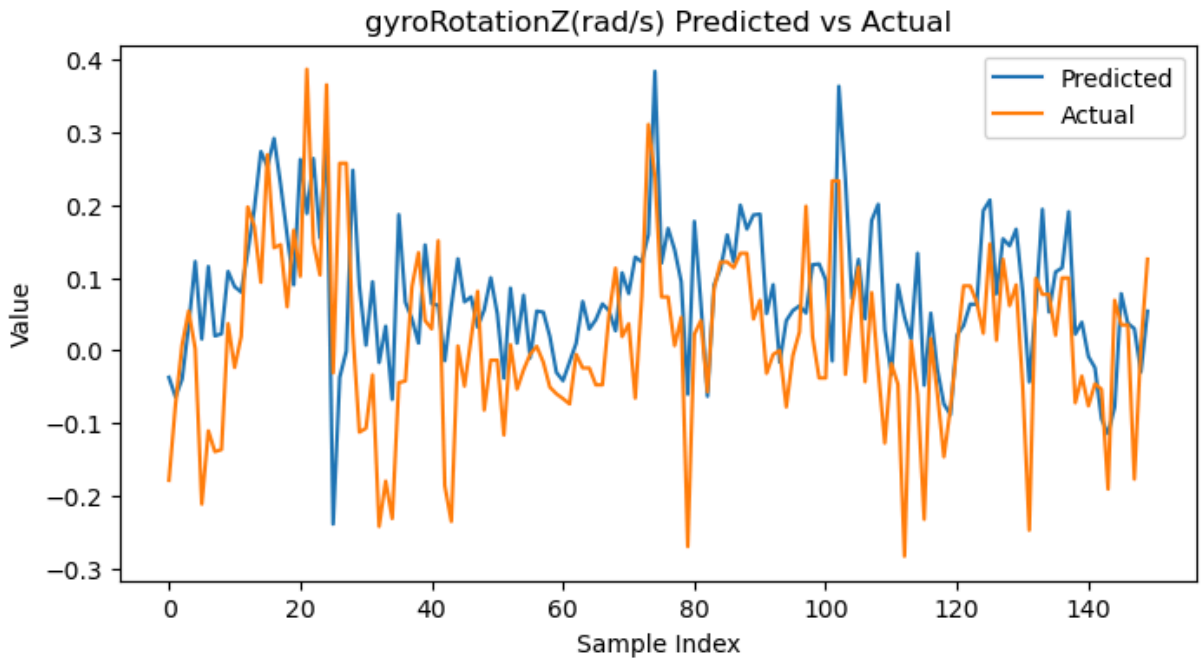



Fig. 8 : GyroRotationZ Predicted vs Actual

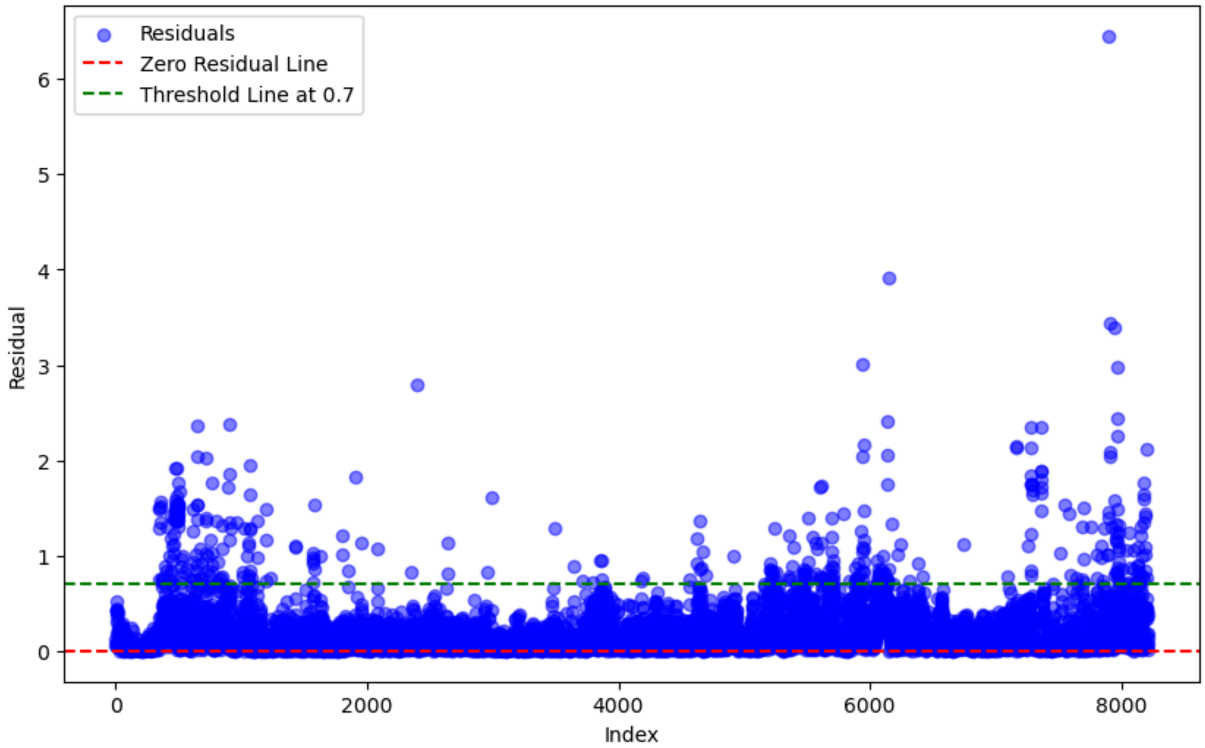



Fig. 9 : Residual Plot

### 7.4.5 Ensemble of MLP and RF

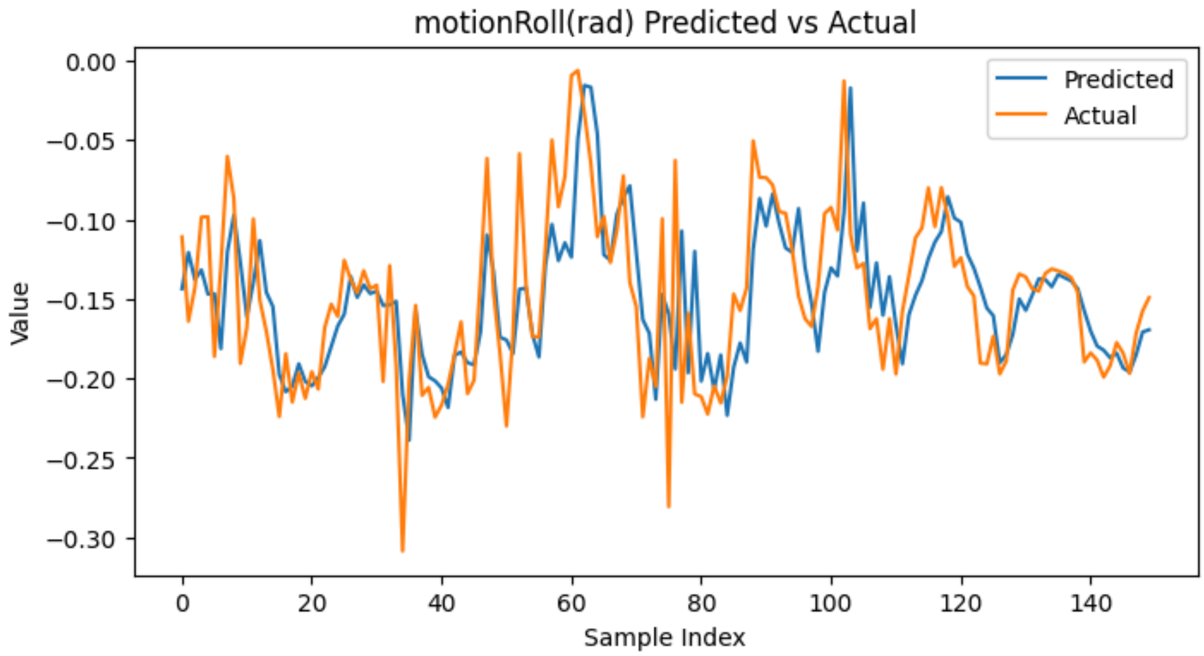



Fig. 2 Motion Roll Predicted vs Actual

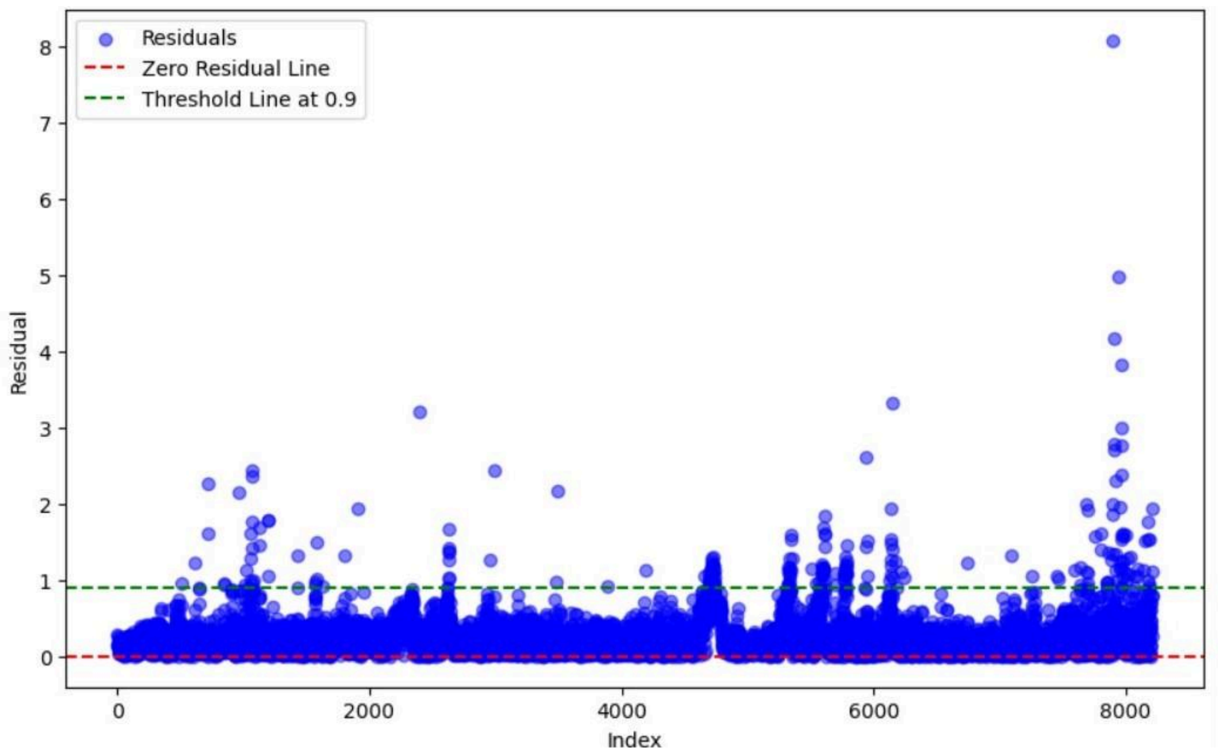



Fig. 3 : Residual Plot

## 7.5. Comparison of results with existing systems

1. Model Selection:
    - Ensemble models like MLP with Random Forest (RF), as well as recurrent neural networks like GRU and LSTM, aligns with common practices in this domain with variations in datasets, sensor configurations, and problem formulations
    - Ensemble models and RNNs have shown strong performance in capturing complex patterns in sensor data, making them suitable for driver behavior analysis.
    - Other techniques like Convolutional Neural Networks (CNNs) and attention mechanisms have also been explored for multi-task learning of driver behavior, including drowsiness and distraction detection, using vehicle sensor data.
2. Performance Metrics:
    - RMSE and MAE are widely used metrics for evaluating regression models in driver behavior studies.
    - Ensemble models perform better on RMSE and MAE for accelerometer and

gyro data align with some existing literature while proposed an ensemble model combining CNNs and LSTMs for driver behavior analysis, outperforming individual models on the SPMD dataset.

- GRU and LSTM's proficiency in capturing motion patterns like yaw, pitch, and roll is also consistent with their ability to model temporal dependencies.

3. Residual Analysis and Anomaly Detection:
   - Computing residuals as the difference between predicted and actual values is a common approach for anomaly detection in driver behavior studies.
   - Using static thresholds on residuals for anomaly flagging is a straightforward technique, but more advanced methods like dynamic thresholding, density-based approaches, and deep learning-based anomaly detection have also been explored.

## 7.6. Inference drawn

| RMSE Values for Multi Output Regression of all models | | | | | |
|---|---|---|---|---|---|
| **Parameters** | **Models** | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Accelerometer X | **0.0485** | 0.0755 | 0.0636 | 0.0499 | 0.1191 |
| Accelerometer Y | **0.0388** | 0.0944 | 0.0811 | 0.0611 | 0.0913 |
| Accelerometer Z | **0.0659** | 0.1465 | 0.1236 | 0.0903 | 0.1132 |
| Gyro Rotation X | **0.1019** | 0.2402 | 0.2043 | 0.1209 | 0.1824 |
| Gyro Rotation Y | **0.1454** | 0.3465 | 0.2850 | 0.1855 | 0.2859 |
| Gyro Rotation Z | 0.1007 | 0.1522 | 0.1057 | **0.0717** | 0.1074 |
| Motion Yaw | 0.0179 | 0.2339 | 0.1371 | **0.0073** | 0.0645 |
| Motion Roll | 0.0182 | 0.0246 | 0.0195 | **0.0087** | 0.0624 |
| Motion Pitch | 0.0195 | 0.0204 | 0.0299 | **0.0087** | 0.0243 |

Table 2 : RMSE Values for multi output regression for all models

Table-2 represents RMSE values for all the parameters with respect to each model for multi output regression. The MLP with RF Ensemble model performs best for Accelerometer(X, Y, Z)and Gyrorotation (X, Y). Meanwhile GRU demonstrates proficiency in capturing patterns for Gyro Rotation Z, Motion Yaw, Motion Pitch and Motion Roll, highlighting its suitability for motion-related features. All the five models measure better predictions for Motion Roll and Motion Pitch.

| MAE Values for Train, Validation and Test Samples | | | | | |
|---|---|---|---|---|---|
| **Mean Absolute Error** | **Models** | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Train | 0.0812 | 0.0914 | 0.0723 | 0.0705 | 0.0857 |
| Validation | 0.0477 | 0.0953 | 0.0839 | 0.0759 | 0.0876 |
| Test | 0.1157 | 0.1422 | 0.1325 | 0.1175 | 0.1179 |

Table 2 : MAE values for train, validation and test samples

The MLP with RF Ensemble model performs better than the other models in terms of Mean Absolute Error (MAE) across Validation, and Test samples, while XGBoost performs better on the Train set. Adaboost has higher MAE on the Test set. GRU and LSTM performs competitively but has slightly higher MAE values compared to the Ensemble.\newline

To evaluate the performance of our predictive model, the predicted test vectors and actual test vectors were calculated by computing the magnitude of the nine chosen parameters. The residual value for each sample was computed as the absolute difference between the magnitude of the predicted vector for that sample and the magnitude of the actual vector for the same sample. Following the residual analysis, a specific static threshold value was set up for each model and anomalies were detected when the residuals exceeded these thresholds.

# Chapter 8 : Conclusion

## 8.1 Limitations

While the proposed models offer promising avenues for enhancing driving assistance systems, several limitations are acknowledged. First, the reliability and accuracy of the models heavily depend on the quality and diversity of the training datasets. Ensuring comprehensive coverage of various driving scenarios, environmental conditions, and edge cases is crucial but can be challenging and resource-intensive.

Another limitation lies in the computational requirements for training and deploying these models, particularly for real-time applications. Efficient hardware acceleration and optimization techniques may be necessary to enable practical deployment in resource-constrained environments, such as embedded systems or mobile devices.

Furthermore, the interpretability and explainability of these models remain a concern, especially in safety-critical applications like autonomous driving. Ensuring transparency and accountability in decision-making processes is essential for building trust and enabling effective human-machine collaboration. The integration of these models into existing driving assistance systems may require significant engineering efforts and careful consideration of system architectures, communication protocols, and safety standards. Overcoming these challenges will be crucial for seamless and reliable operation in real-world conditions.

## 8.2 Conclusion

The proposed models built on a comprehensive analysis of driver behavior and road conditions, hold the potential to be a transformative step towards enhancing road safety and the overall driving experience in India. The data-driven approach, leveraging five models: LSTM-R, Ensemble of Multilayer Perceptron(MLP) and Random Forest, Adaboost Regressor, XGBoost regressor and GRU offering a robust foundation for this endeavor.

In conclusion, the models provide a potential to enhance road safety in India substantially. The data-driven, AI-powered approach, combined with the integration of various data sources, makes it a valuable tool for both individual drivers and authorities responsible for road safety. It represents a vital step towards a safer and more efficient road ecosystem in India.

The project involved an extensive literature survey, which highlighted the need for proactive solutions tailored specifically for two-wheeler safety. Existing systems were found to be primarily designed for cars, leaving a gap in holistic support for two-wheeler riders.

This project represents a significant step toward improving road safety and addressing the pressing issue of two-wheeler rider accidents. The methodologies and findings presented here lay the foundation for future research and initiatives aimed at creating a safer environment for all road users in India.

## 8.3 Future Scope

The MLP has a multi-layered architecture which enables it to learn hierarchical features. With further research, the model can be used to learn more complex patterns like specific driving maneuvers or sequences of actions. Dataset creation can be focused on the collection of various other parameters such as weather conditions, traffic data, road conditions, rider behavior, and route information. This Ensures a wide range of driving scenarios. The datasets will be instrumental in enhancing the accuracy and robustness of the model predictive and analytical capabilities. It could be used for logistic planning, Risk assessment, Insurance companies etc for secondary analytics to further include different business models.

The proposed models can be used for the implementation of real-time visualizations and anomaly detection in driver assistance systems. The applications can process and analyze streaming data from sensors and cameras to provide instant insights into driving behaviors. Real-time features can include speed, steering patterns, and other relevant parameters.

## References

1] Ministry of Road Transport and Highways. Road accidents in india. pages 1163–1168, 2023. URL: https://morth.nic.in/sites/default/files/RA_2021_Compressed.pdf, doi:11Apr2023.

[2] Xian-Hong Huang, Chen Zhi-Hao, Ahamad Afaroj, and Sun Chi-Chia. Adas e-bike: Auxiliary adas module for electric power-assisted bicycle. pages 1–2, 2022. doi:10.1109/IET-ICETA56553.2022.9971513.

[3] B. Ball and Tang J.E. Machine learning and embedded computing in advanced driver assistance systems (adas). June 2019. URL: https://www.mdpi.com/2079-9292/8/7/748, doi:10.3390/electronics8070748.

[4] De-Las-Heras G., Sánchez-Soriano J., and Puertas E. Advanced driver assistance systems (adas) based on machine learning techniques for the detection and transcription of variable message signs on roads. July 2021. URL: https://www.mdpi.com/1424-8220/21/17/5866, doi:10.3390/s21175866.

[5] Tselentis Dimitrios I., Eleonora Papadimitriou, and Pieter van Gelder. The usefulness of artificial intelligence for safety assessment of different transport modes. 2023. URL: https://www.sciencedirect.com/science/article/pii/S0001457523000817?via%3Dihub, doi:10.1016/j.aap.2023.107034.

[6] Yongfeng Ma, Zhuopeng Xie, Shuyan Chen, Fengxiang Qiao, and Zeyang Li. Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals. 2023. URL: http://www.sciencedirect.com/science/article/pii/S0968090X22003965, doi:10.1016/j.trc.2022.103983.

[7] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li. D3: Abnormal driving behaviors detection and identification using smartphone sensors. 2015. URL: https://ieeexplore.ieee.org/document/7338354, doi:http://dx.doi.org/10.1109/SAHCN.2015.7338354.

[8] G.K. Sahoo, S.K. Das, and P. Singh. Two-layered gated recurrent stacked long short-term memory networks for driver's behavior analysis. S ¯adhan ¯a, 48, 2023. URL: https://www.ias.ac.in/article/fulltext/sadh, doi:10.1007/s12046023-02126-y.

[9] R. Kumar and A. Jain. Driving behavior analysis and classification by vehicle obd data using machine learning. Journal of Supercomputing, 79:18800–18819, 2023. doi:10.1007/s11227-023-05364-3.

[10] Xiupeng Shi, Yiik Wong, Michael Li, Chandrasekar Palanisamy, and Chen Chai. A feature learning approach based on xgboost for driving assessment and risk prediction. Accident Analysis & Prevention, 129:170–179, 2019. URL: https://www.sciencedirect.com/science/article/abs/pii, doi:10.1016/j.aap.2019.05.005.

[11] Mehdi Ghatee Amin Fattahi, Amir Golroo. Driver behavior assessment using multi-layer perceptron and random forest via smartphone sensors and obd ii. pages 1163–1168, 2023. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4416107, doi:11Apr2023.

[12] Nan Ding, HaoXuan Ma, Huanbo Gao, YanHua Ma, and GuoZhen Tan. Real-time anomaly detection based on long short-term memory and gaussian mixture model. Computers & Electrical Engineering, 79(1):106458, January 2021. URL: https://www.sciencedirect.com/science/article/pii/S0045790618334372, doi:10. 1016/j.compeleceng.2019.106458.

[13] Hassan Bousnguar, Amal Battou, and Lotfi Najdi. Gated recurrent units (gru) for time series forecasting in higher education. International Journal of Engineering Research & Technology (IJERT), 12(03):To be determined, March 2023. doi:10. 17577/IJERTV12IS030091.

[14] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. pages 1597–1600, 2017. doi:10.1109/MWSCAS.2017.8053243.

[15] Devon Barrow. A comparison of adaboost algorithms for time series forecast combination. International Journal of Forecasting, 32:1103–1119, 06 2016. doi: 10.1016/j.ijforecast.2016.01.006.

[16] Zeravan Ali, Ziyad Abduljabbar, Hanan Tahir, Amira Sallow, and Saman Almufti. Exploring the power of extreme gradient boosting algorithm in machine learning: a review. 12:320–334, 05 2023. doi:10.25007/ajnu.v12n2a1612.

[17] BabekHassa Erdebilli and Burcu Devrim-İçtenbaş. Ensemble voting regression based on machine learning for predicting medical waste: A case from turkey. Mathematics, 10(14), 2022. URL: https://www.mdpi.com/2227-7390/10/14/2466, doi:10.3390/math10142466.

[18] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. International Journal of Interactive Multimedia and Artificial Inteligence, 4:26–30, 01 2016. doi:10.9781/ijimai.2016.415.

[19] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. Pattern Recognition Letters, 31:2225–2236, 10 2010.

doi:10.1016/j.patrec.2010.03.014.
[20] Fereidoon Moghadas Nejad Mehrdad Ehsani and Pouria Hajikarimi. Developing an optimized faulting prediction model in jointed plain concrete pavement using artificial neural networks and random forest methods. International Journal of Pavement Engineering, 24(2):2057975, 2023. arXiv:https://doi.org/10.1080/10298436.2022.2057975, doi:10.1080/10298436.2022.2057975.
[21] Amir Samimi Mahdi Zarei Yazd, Iman Taheri Sarteshnizi and Majid Sarvi. A robust machine learning structure for driving events recognition using smartphone motion sensors. Journal of Intelligent Transportation Systems, 28(1):54–68, 2024. arXiv:https://doi.org/10.1080/15472450.2022.2101109, doi:10.1080/15472450.2022.2101109.
[22] Leo Breiman. Random forests. Machine learning, 45:5–32, 2001.
[23] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences, pages 771–785. 08 2023. doi:10.1007/978-3-031-37963-5_53.
[24] Michael Phi. Illustrated guide to lstm's and gru's: A step by step explanation.2018.
[25] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. Artificial Intelligence Review, 53, 12 2020. doi:10.1007/s10462-020-09838-1.
[26] Y. Ding, Hongyang Zhu, Ruyun Chen, and Ronghui Li. An efficient adaboost algorithm with the multiple thresholds classification. Applied Sciences, 2022. URL: https://api.semanticscholar.org/CorpusID:249562694.
[27] Tu Chengsheng, Liu Huacheng, and Xu Bing. Adaboost typical algorithm and its application research. MATEC Web of Conferences, 139:00222, 01 2017. doi:10.1051/matecconf/201713900222.
[28] Muhammad Adnan, Alaa Alarood, M. Irfan Uddin, and Izaz Rehman. Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. PeerJ Computer Science, 8:e803, 02 2022. doi:10.7717/peerj-cs.803.
[29] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), pages 785–794, 2016. doi:10.1145/2939672.2939785.
[30] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of xgboost, 11 2019.
[31] Sirilak Areerachakul and Siripun Sanguansintukul. Classification and regression trees

and mlp neural network to classify water quality of canals in bangkok, thailand. 2010. URL: https://api.semanticscholar.org/CorpusID:132729585.
[32] S.Ridella D. Anguita and F.Rivieccio. K-folds generalization capability assessment for support vector classifiers. URL: https://api.semanticscholar.org/CorpusID:132729585.
[33] Li hua Chen and Xiao yun Zhang. Application of artificial neural network to classify water quality of the yellow river. URL: https://api.semanticscholar.org/CorpusID:132729585.
[34] S.Areerachakul and S.Sanguansintukul. “water classification using neural network: A case study of canals in bangokok, thailand. URL: https://api.semanticscholar.org/CorpusID:132729585.7

# Comparative Analysis of Machine Learning Models for classifying abnormal driving behavior in two-wheeler kinematics

Nupur Giri
nupur.giri@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Yogesh Pandit
yogesh.pandit@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Simran Ahuja
2021.simran.ahuja@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Sania Khan
2021.sania.khan@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Jesica Bijju
2021.jesica.bijju@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Sejal Datir
2021.sejal.datir@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

**ACM Reference Format:**
Nupur Giri, Yogesh Pandit, Simran Ahuja, Sania Khan, Jesica Bijju, and Sejal Datir. 2024. Comparative Analysis of Machine Learning Models for classifying abnormal driving behavior in two-wheeler kinematics. In . , 7 pages.

## 1 ABSTRACT

This research paper focuses on the driving behaviour of two wheeler riders who are particularly vulnerable to road accidents. Through the analysis of time-series data collected from real-world driving scenarios, the study aims to develop machine learning models for risk assessment and anomaly detection. The paper starts by outlining the alarming statistics of road accidents in India, emphasizing the urgent need for effective solutions. The study explores the effectiveness of five distinct models: Long Short-Term Memory -Residual (LSTM-R), Gated Recurrent Unit (GRU), Adaboost , XGBoost, Multilayer Perceptron (MLP) with Random Forest (RF) ensemble.

## 2 INTRODUCTION

In recent years, India has witnessed an escalation in traffic accidents. The Ministry of Road Transport and Highways' annual report for the fiscal year 2021–2022 shows a startling statistic: there were 4,12,432 road accidents registered nationwide, resulting in 1,53,972 fatalities and 3,84,448 injuries. According to NH road accident and fatality data, two-wheelers accounted for the most number of accidents (52,416) and deaths (22,786) in 2021 [1]. The leading cause of more than half of these accidents was overspeeding, with dangerous and careless driving following closely. These alarming statistics highlight the importance of developing effective solutions to reduce road accidents and elevate road safety standards.

In response to this critical and escalating public health challenge, our research aims to address the urgent need for enhanced road safety measures, especially for two-wheeler riders. Through careful examination of the actions and habits of two-wheeler drivers, our research endeavors to uncover the factors contributing to accidents.

## 3 LITERATURE REVIEW

One of the primary causes of accidents on the roads is abnormal driving behaviour. Most studies on anomalous driving behaviour classify reconstruction or prediction residuals using unsupervised learning techniques, or they identify and evaluate the driving state using direct classification algorithms. Data on unusual driving behaviour are challenging to find and categorise. Due to this limited data of the anomalous driving behavior, there is an inherent class imbalance problem in the algorithm training process. Furthermore, the majority of the existing residual analysis studies concentrate on single points, which makes it difficult to capture the continuous feature of abnormal driving behavior.

Several papers have been proposed on Advanced Driver Assistance System that help drivers in routine navigation by leveraging computer networks to enable more data-driven and safer driving experiences. Advanced Driver Assistance Systems (ADAS) module that is based on AI neural networks is proposed in [2] to improve road safety. John et al. in [3] presents Advanced machine learning algorithms based ADAS system to detect objects, obstacles, other vehicles, pedestrians, and lanes, and the estimation of object trajectories and intents. The work described in [4] introduces a functional prototype of a machine learning-driven system for Variable Message Sign (VMS) interpretation. This system is capable of recognizing traffic signals, extracting their textual information, and converting it into spoken output..

Paper [5] proposes crash prediction based on, incident/failure detection, pattern identification, driver/operator or route assistance, using ANN, SVM, Hidden Markov Models and Bayesian models. In Paper [6], YongFeng Ma et al. suggests a long short-term memory-residual (LSTM-R) algorithm to identify abnormal driving behavior. This algorithm consists of two parts: firstly, an LSTM network is used to analyze historical vehicle kinematic data, yielding root mean square residuals at each time point. Secondly, a time window-based residual algorithm is applied to identify abnormal driving behaviors, calculating the magnitude and continuity of these residuals. Paper[7] presents real-time high-accurate abnormal driving behaviors monitoring using smartphone sensors and uses SVM. Paper [8] implements a two-layer GRU stacked long short-term memory (LSTM) network to categorize driving characteristics. OBD-II protocol was implemented to collect real-time vehicle performance in paper [9] using SVM, AdaBoost, and Random Forest. Paper [10] utilizes XGBoost to create associations between behavior features and risk levels. Random Forest for feature selection and Multilayer Perceptron (MLP) neural network for maneuver classification in Paper [11] achieving an 89% overall accuracy in identifying aggressive driving maneuvers using smartphone sensors and OBD

*We acknowledge IIIT-H/IHub/Project/Mobility/2023-24/M2-018 for financial assistance, 2024*
.

II data. Paper [12] proposes the LGMAD, a real-time anomaly detection algorithm based on Long-Short Term Memory (LSTM) and Gaussian Mixture Model (GMM) whereas paper [13] explores the use of a gated neural network for time series forecasting in the context of higher education. Three variations of the Gated Recurrent Unit (GRU) evaluated in paper [14] by retaining the structure and methodically decreasing parameters in the update and reset gates. Rigorous empirical evaluations were conducted in paper [15] of seven variants of the popular AdaBoost algorithm. Paper [16] uses one of the most well-known machine learning algorithms for supervised and semi-supervised learning (SSL), such as XGBoost, in a Python environment. The paper referenced by [17] introduces an ensemble voting regression algorithm leveraging machine learning techniques like random forests, gradient boosting machines, and adaptive boosting. Paper [18] introduces a new approach to optimize the network architecture of a MLP by using the genetic algorithm and a back-propagation algorithm. while paper [19] focuses on random forests to provide some experimental insights about the behavior of the variable importance index. Paper [20] has used random forest and artificial neural network techniques to create an optimal faulting prediction model for jointed plain concrete pavement. A two-phase Machine Learning (ML) method using high-pass, low-pass, and wavelet filters to detect driving brakes and turns using Random Forest and Artificial Neural Network classifiers was proposed in paper [21]. Article [22] discusses how random forests, consisting of tree predictors, make use of random feature selection to create an ensemble that converges to a stable generalization error. In the paper[23], the memorization techniques of RNN's, specifically LSTM and GRU was explored concluding that an increase in depth does not always improve memorization. The article [24] provides a guide to Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), explaining their internal mechanisms and role in mitigating short-term memory issues in recurrent neural networks. The paper [25], highlights the impact of LSTM on machine learning and neuro-computing. A novel approach to forecast the value of gold (XAU/USD) using a regression ensemble, linear regression, decision tree regression, and stacking regression, with stacking regression was discussed in paper [26].

## 4 NOVELTY OF WORK

Data was systematically collected from a consistent route over seven rides, covering a 4 km distance. Nine parameters were recorded: Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch using the SensorLog iOS mobile application. The methodology contributes in addressing the multi-output regression tasks involving time-series data. We trained LSTM-R, GRU, AdaBoost, XGBoost, and Ensemble of MLP and Random Forest for prediction, aiming to understand connections among different sensor variables simultaneously. We integrated information from accelerometers, gyroscopes, and motion angles, emphasizing a direct combination of insights from various sensors. The evaluation process involved visualizing predicted versus actual values, calculating RMSE, and conducting a residual analysis, providing a practical understanding of the model's performance and areas for improvement.

## 5 METHODOLOGY

1. A series of test drives was conducted over a 4 km route in Chembur, Mumbai, utilizing Sensorlog, an iOS application, to collect vehicle kinematic data in real time with a sampling frequency of 10 Hz. The data collection process involved two individuals: a rider and a passenger equipped with a handheld iOS mobile device running the Sensorlog app. The application continuously recorded the rider's movements and behavior, capturing parameters including Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch of the bike. Additionally, Magnetometer readings (X, Y, Z) were initially included but later excluded due to redundancy. The data collection occurred over a period of 7 days, during which 81,480 tuples of data were gathered, with each ride averaging 20 minutes in duration. The dataset encompassed data collected from both normal driving behavior and rash driving behavior exhibited on the same route.

2. Preprocessing: Before fitting the model, the input features are reshaped for all models. The reshaping is performed to flatten the input features, converting the 2D array into a 1D array, making it compatible with the model's expectations. The dataset was split into training, validation, and test sets.

3. Model selection: Five models were trained on the nine parameters. Neural network models- LSTM-R and GRU were trained first, followed by the boosting algorithms- AdaBoost and XGBoost. The final model that was trained was a combination of both neural networks and boosting- the ensemble of MLP with RF.

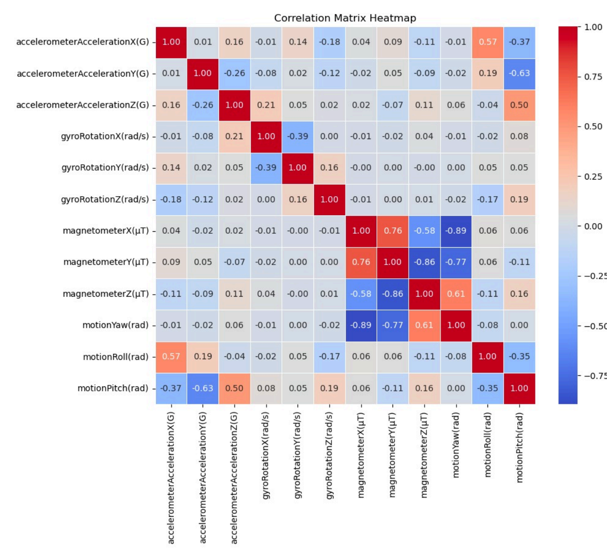



Fig. 1 : Confusion matrix

Fig. 1 depicts the correlation between the recorded parameters

## 6 MODEL TRAINING

The model creation process involves the training of five models: long short-term memory-residual (LSTM-R), gated recurrent unit (GRU), AdaBoost, XGBoost, and ensemble of multilayer perceptron (MLP) with random forest. The dataset is split into three parts: 70

% for training, 15 % for testing , and 15 % for validation. The input layer is structured to accept data in windows of a specific size.

## 6.1 LSTM - Residual:

6.1.1 LSTM

LSTM is a type of recurrent neural network (RNN) that analyzes sequential data [12]. LSTM's ability to capture temporal dependencies made it a suitable choice for the two wheeler time-series dataset used in this study. In an LSTM network, there are three gates (input, forget, and output gates) that regulate the flow of information, and the memory cell stores long-term dependencies[25].

6.1.2 LSTM-Residual (LSTM-R)

The LSTM-R model calculates residuals, which are the differences between the observed data and the data predicted by the model. It uses a time window-based approach, with the residual values computed over a window of 100 milliseconds. The LSTM units act as intelligent memory cells, recognizing patterns across multiple time steps as data patterns evolve over various temporal scales.
In this model, layers are organized sequentially which begins with an input layer to handle sequences with nine features. The core is an LSTM layer with 64 units, designed to capture patterns over time. After that, a Dense layer with eight units and Rectified Linear Unit (ReLU) activation is added for non-linear transformations. The final layer is another Dense layer with nine units, using a linear activation.

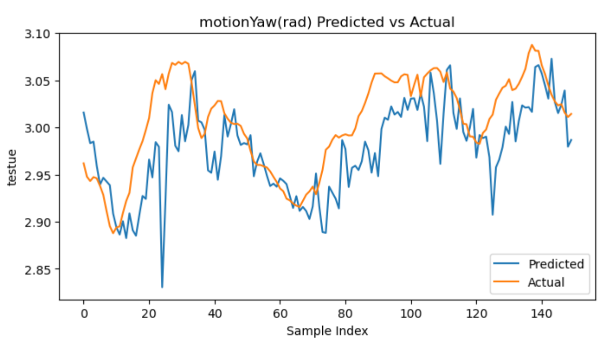



Fig. 2 : Motion Yaw Predicted vs Actual

Fig. 2 depicts the Actual vs. Predicted values of Motion Yaw(rad) trained on LSTM-R model.

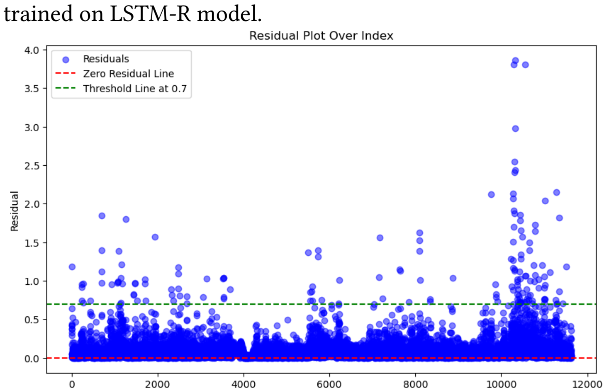



Fig. 3 : Residual Plot

Fig. 3 depicts the residual values for the LSTM-R model.

## 6.2 GRU

Following the training of LSTM-R, Gated Recurrent Units (GRU) was chosen for its simpler architecture. GRU has two gates: reset and update gate as opposed to LSTM which has three gates[23]. GRU is a type of RNN like LSTM with a potential for faster training[13]. This choice aims to explore the strengths of both models and improve performance on the time series dataset.
GRU consists of parameters that control the memory states and aims to solve the exploding gradient problem of traditional RNNs.[14] It involves a gating mechanism that helps selectively update and reset the hidden states. The reset gate controls the information from the previous time step to be forgotten, while the update gate regulates how much of the new information should be added to the cell state.
The GRU architecture consists of an input layer and uses a time window-based approach. The structure of the GRU model was determined through repeated testing and comparison of the number of hidden layers, activation functions, number of units in each layer, batch size, dropout rate and epochs.
The model incorporated three GRU hidden layers with decreasing units 64, 32, and 16 respectively to capture sequential dependencies. This structure is further extended with two dense layers to map the extracted features from the hidden layers to the output space with 9 units each, one that uses the Rectified Linear Unit(ReLU) and the other that uses the Leaky Rectified Linear Unit(LeakyReLU) activation function. The addition of dropout layers as a regularization technique was avoided since GRU layers already have internal mechanisms in the form of reset and update gates to control the information flow and mitigate vanishing gradient problems, thereby avoiding overfitting. [24]
For model optimization and training, we utilized Mean Absolute Error (MAE) as the loss function, paired with the Adam optimizer set at a learning rate of 0.001.

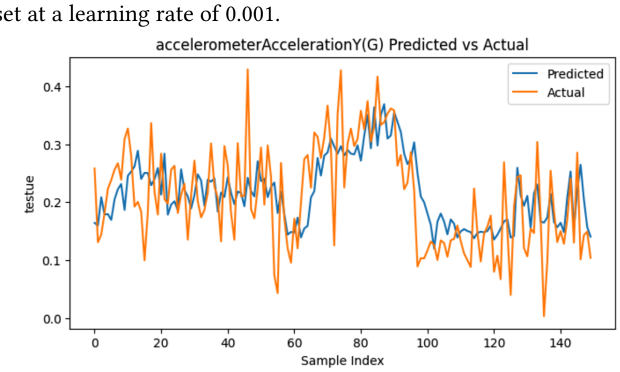



Fig. 4 : AccelerometerY Predicted vs Actual

Figure 4 compares the Actual and the Predicted values of AccelerometerY trained on GRU.

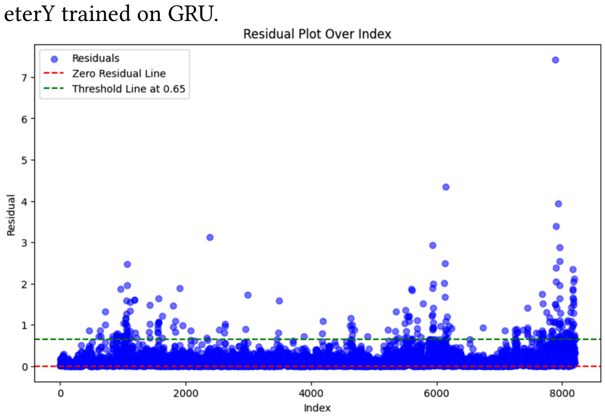



Fig. 5 : Residual Plot

Figure 5 showcases the Residual over time offering an insight to the anomalies from the expected behavior for GRU.

## 6.3 AdaBoost

After training the model with neural networks like LSTM and GRU, the dataset was trained on boosting algorithms to see how they perform in comparison to neural networks for the multioutput regression task. The Adaboost model follows the boosting ensemble learning method. It makes use of 'adaboost regressor' as the weak learner and Decision Trees as the default base estimator [27]. It is then followed by Weighted Training where weights are assigned to each training instance. Initially, all weights are set equally. However, during the training process, the weights are adjusted based on the performance of the weak learners. Misclassified instances are given higher weights to focus on them during subsequent iterations. Weak learners are then trained sequentially. Each subsequent learner focuses more on the instances that were misclassified by the previous learners [28]. This is performed for each output variable and a new adaboost regressor is trained for each variable. The final prediction is a weighted sum of the predictions from individual weak learners. The weights are determined based on the performance of each learner.

Grid Search CV was employed to discover the optimal hyperparameters for the model, utilizing a 'param grid' dictionary, an Adaboost Regressor as the base model, and a Grid Search CV object [29]. The param grid contained various combinations of hyperparameter values, evaluating each set to determine the most effective configuration. Notably, the grid focused on three key hyperparameters: number of estimators (number of weak learners), learning rate (scaling factor for each weak learner's contribution), and loss (the function impacting weight updates and weak learner performance measurement). The choice of loss function significantly influenced the boosting algorithm's behavior. The Adaboost Regressor served as the base model for the GridSearch CV, which, employing four-fold cross-validation, explored the parameter grid for detailed output. The best parameters identified through this process were number of estimators equal to 100, learning rate of 0.01, and loss set to 'exponential.'

After training, the Adaboost Regressor is used to make predictions on the training, validation, and test sets. The Root Mean Square Error (RMSE) values are calculated for each parameter and Mean Absolute Error (MAE) is calculated for each set by comparing the predicted values to the actual labels. This metric is used to evaluate the model's performance on different subsets of the data.

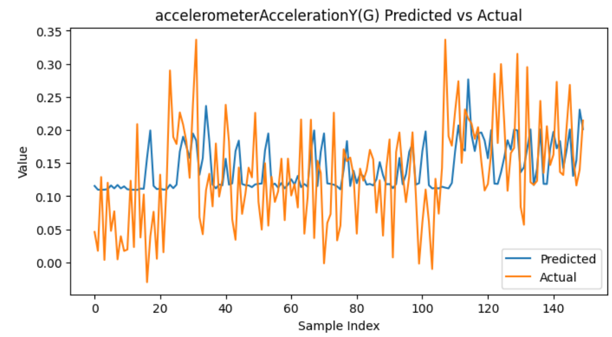



Fig. 6 : Accelerometer(Y) Predicted vs Actual

Figure 6 compares the predicted values generated by the ensemble of AdaBoost regressors with the actual values for motion pitch.

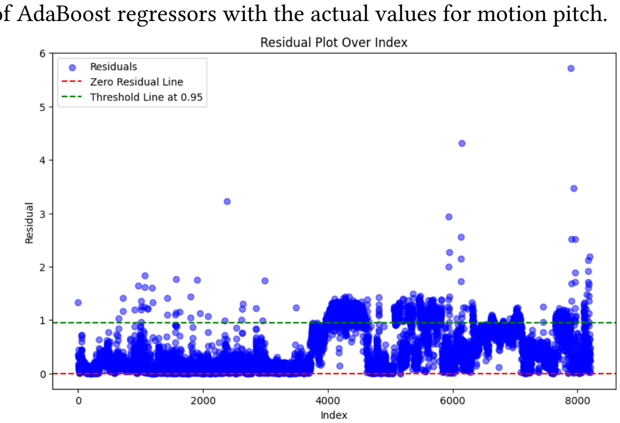



Fig. 7 : Residual Plot

## 6.4 XGBoost

Figure 10 shows the actual vs predicted graph for Adaboost model. It was observed that the predictions did not fit well with the actual data points. This observation prompted a transition to XGBoost, a gradient boosting algorithm, known for its sequential tree construction [30]. XGBoost offers distinct advantages, including effective regularization methods and enhanced control over fine-tuning—features absent in Adaboost [16].

The XGBoost model adopts a structured approach, processing data in specific windows. This strategy proves effective in isolating and characterizing driving events based on their duration, providing valuable contextual insights for analysis and intervention [31]. XGBoost iteratively constructs a sequence of decision trees, each addressing the errors of its predecessor. This iterative process steadily enhances the model's predictive capabilities. Central to XGBoost's functionality is its foundation in gradient boosting, a methodology that optimizes a loss function through the calculation of gradients

concerning the model's predictions, this approach allows the algorithm to make corrections that minimize overall prediction errors, contributing to the model's improved accuracy.
Hyperparameters were experimented with to fine-tune the configuration of the XGBoost model. The objective was to identify the optimal settings that would minimize the Root Mean Square Error (RMSE) values, ensuring the model's ability to make accurate and reliable predictions. Specifically, the selected key hyperparameters were- learning rate set at 0.2, maximum depth of 3 for each individual tree, alpha value of 10 for regularization strength , and 100 decision trees in the ensemble.

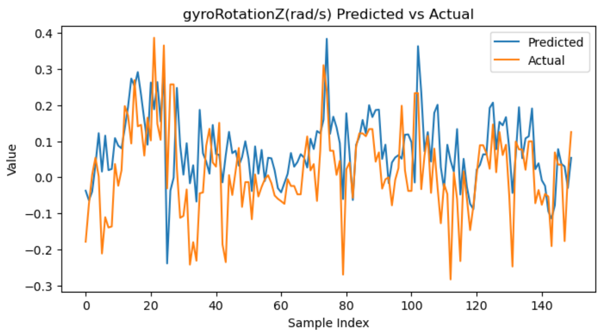



Fig. 8 : GyroRotationZ Predicted vs Actual

Figure 8 depicts the Actual vs. Predicted values of Motion Roll, GyroRotationZ.

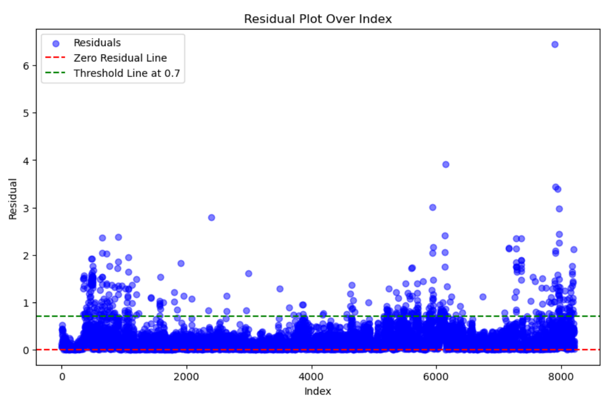



Fig. 9 : Residual Plot

## 6.5 Ensemble of Multilayer Perceptron(MLP) and Random Forest:

After a separate analysis of neural networks and boosting algorithms, training a model on the combination of both could offer a different perspective.

### *6.5.1 Multilayer Perceptron Neural Network:*

The Multilayer Perceptron has a large wide of classification and regression applications in many fields: pattern recognition, voice and classification problems. [18]. It is one of the most common and practical artificial neural networks in which each neuron is connected to several neighbor neurons [11]. A general network consists of a layered architecture, an input layer, one or more hidden layers and an output layer [32]. A common artificial neural network used to address a variety of issues, such as pattern recognition and interpolation, is the Multilayer Perceptron (MLP) [33][34]. Neurons make up each layer, and they are related to one another via weights. A unique mathematical function known as the activation function is present in every neuron and uses information from earlier levels to produce output for the layer after it. The hyperbolic tangent sigmoid transfer function is the activation function employed in the experiment [35].

### *6.5.2 Random Forest:*

The random forest algorithm is commonly used to handle regression and classification problems [20]. It can be defined as a regression technique that combines the performance of numerous decision trees independently and then averaging their predictions for the final outcome [22]. Random Forests offer efficient training and prediction processes from a computational perspective, relying on just a couple of tuning parameters, and can easily be implemented in parallel. Statistically, Random Forests provide additional features such as the measures of variable importance, differential class weighting. Each tree in the random forest is created based on a random subset of the input variables, and each tree branching is created based on a random number of selected input features. [11]

### *6.5.3 Ensemble of MLP with Random Forest:*

The Ensemble model was created using the combination of Multi-Layer Perceptron as the neural network, and Random Forest for the boosting. A voting ensemble employs multiple models instead of a single one to enhance the system's performance, as discussed in [17]. Regressor voting is beneficial for comparing many regression models and selecting the one that produces the best predictions. The predictions from all models are combined through either average voting (AV), or weighted voting (WV)[17]. The Voting Regressor, which combines regression models of MLP and RF by assigning weights to each model, developed a weighted strategy in this case so that the prediction results are based on the results of the selection of these models. The ensemble model is then implemented as a MultiOutputRegressor, allowing it to handle multi-output regression tasks.
Determining the optimal configuration of the MLP, including the number of hidden layers, the number of neurons, the type of activation function, and the optimization method was done through trial and error [19]. The MLP network was trained with the ReLU activation function and the Adam optimizer. After trial and error, the best neural network architecture was obtained with 3 hidden layers with 100, 50, and 25 neurons in each hidden layer respectively. Other hyperparameters include an L2 regularization term of 0.0001, a constant learning rate of 0.001, and a batch size that dynamically adjusts during training.
The RandomForestRegressor consists of 200 decision trees, employing the Mean Squared Error criterion. The ensemble model, created using VotingRegressor, combines predictions from both models. The model's evaluation involves assessing its performance on both training and test datasets using metrics such as Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

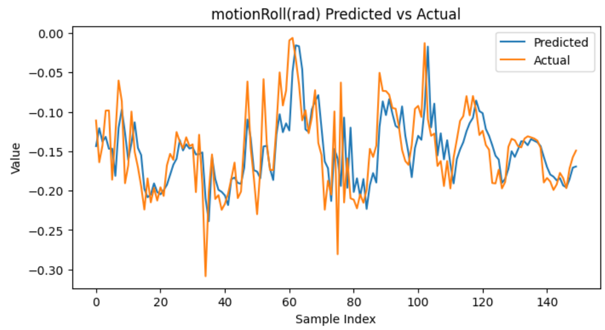



Fig. 10 : Motion Roll Predicted vs Actual

Figures 10 depicts the Actual vs. Predicted values of Motion Roll parameter generated by the Ensemble of MLP and Random Forest.

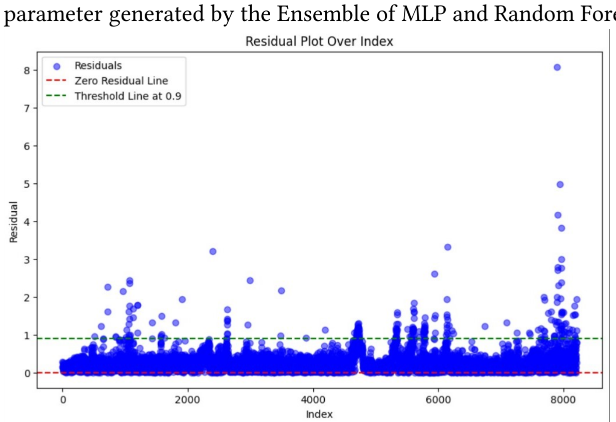



Fig. 11 : Residual Plot

Figure 11 showcases the Residual over time of the MLP-Random Forest Ensemble

| Models | Parameters | Parameter Range | Optimal Parameters |
|---|---|---|---|
| MLP - RF Ensemble | number of hidden layers $N_l$<br>number of neurons $N_n$<br>activation layers $A_l$<br>optimizer $O$<br>number of estimators $N_e$<br>learning rate $L_r$ | $N_l \in$ {1, 2, 3}<br>$N_n \in$ {100, (20,20), (100,50,25)}<br>$A_l \in$ {relu, logistic, hyperbolic tan}<br>$O \in$ {adam, sgd }<br>$N_e \in$ {100,200,300,400,500}<br>$L_r \in$ {0.001, 0.0001} | $N_l$ = 3<br>$N_n$ = (100,50,25)<br>$A_l$ = relu<br>$O$ = Adam<br>$N_e$ = 200<br>$L_r$ = 0.001 |
| Adaboost | number of estimators $N_e$<br>learning rate $L_r$<br>loss $L$ | $N_e \in$ {50, 100, 200}<br>$L_r \in$ {0.001, 0.01, 0.1, 0.2}<br>$L \in$ {linear ,square, exponential} | $N_e$ = 100<br>$L_r$ = 0.01<br>$L$ = exponential |
| XGBoost | objective $O_b$<br><br>colsample bytree $N_t$<br>learning rate $L_r$<br>maximum depth $N_d$<br>alpha $\alpha$<br>number of estimators $N_e$ | $O_b \in$ {reg:squarederror, reg:logistic, reg:gamma}<br>$N_t \in$ {[0.1 - 1.0]}<br>$L_r \in$ {[0.01 - 0.3]}<br>$N_d \in$ {2, 3, 4, 5, 6}<br>$\alpha \in$ {[1 - 20]}<br>$N_e \in$ {[50 - 200]} | $O_b$ = reg:squarederror<br><br>$N_t$ = 0.8<br>$L_r$ = 0.2<br>$N_d$ = 3<br>$\alpha$ = 10<br>$N_e$ = 100 |
| GRU | number of units $N_u$<br>learning rate $L_r$<br>number of hidden layers $N_l$<br>activation layers $A_l$<br>batch size $m$ | $N_u \in$ {8, 16, 32, 64, 128}<br>$L_r \in$ {0.001, 0.0001}<br>$N_l \in$ {2, 3, 4}<br>$A_l \in$ {relu, leakyrelu, softmax, tanh}<br>$m \in$ {0, 16, 32} | $N_u$ = 16, 32, 64<br>$L_r$ = 0.001<br>$N_l$ = 3<br>$A_l$ = relu, leakyrelu<br>$m$ = 0 |
| Residual LSTM | number of units $N_u$<br>learning rate $L_r$<br>activation layers $A_l$<br>batch size $m$ | $N_u \in$ {16, 32, 64, 128}<br>$L_r \in$ {0.001, 0.0001}<br>$A_l \in$ {relu, linear, softmax, tanh}<br>$m \in$ {0, 16, 32} | $N_u$ = 64<br>$L_r$ = 0.0001<br>$A_l$ = relu, linear<br>$m$ = 0 |

Table 1 : RMSE Values for multi output regression for all models

# 7 MODEL EVALUATION

| RMSE Values for Multi Output Regression of all models | | | | | |
|---|---|---|---|---|---|
| **Parameters** | **Models** | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Accelerometer X | **0.0485** | 0.0755 | 0.0636 | 0.0499 | 0.1191 |
| Accelerometer Y | **0.0388** | 0.0944 | 0.0811 | 0.0611 | 0.0913 |
| Accelerometer Z | **0.0659** | 0.1465 | 0.1236 | 0.0903 | 0.1132 |
| Gyro Rotation X | **0.1019** | 0.2402 | 0.2043 | 0.1209 | 0.1824 |
| Gyro Rotation Y | **0.1454** | 0.3465 | 0.2850 | 0.1855 | 0.2859 |
| Gyro Rotation Z | 0.1007 | 0.1522 | 0.1057 | **0.0717** | 0.1074 |
| Motion Yaw | 0.0179 | 0.2339 | 0.1371 | **0.0073** | 0.0645 |
| Motion Roll | 0.0182 | 0.0246 | 0.0195 | **0.0087** | 0.0624 |
| Motion Pitch | 0.0195 | 0.0204 | 0.0299 | **0.0087** | 0.0243 |

Table 2 : RMSE Values for multi output regression for all models

Table-2 represents RMSE values for all the parameters with respect to each model for multi output regression. The MLP with RF Ensemble model performs best for Accelerometer(X, Y, Z)and Gyrorotation (X, Y). Meanwhile GRU demonstrates proficiency in capturing patterns for Gyro Rotation Z, Motion Yaw, Motion Pitch and Motion Roll, highlighting its suitability for motion-related features. All the five models measure better predictions for Motion Roll and Motion Pitch.

| MAE Values for Train, Validation and Test Samples | | | | | |
|---|---|---|---|---|---|
| **Mean Absolute Error** | **Models** | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Train | 0.0812 | 0.0914 | **0.0723** | 0.0735 | 0.0857 |
| Validation | **0.0477** | 0.0953 | 0.0839 | 0.0759 | 0.0876 |
| Test | **0.1157** | 0.1422 | 0.1325 | 0.1175 | 0.1179 |

Table 2 : mae values for train, validation and test samples

The MLP with RF Ensemble model performs better than the other models in terms of Mean Absolute Error (MAE) across Validation, and Test samples, while XGBoost performs better on the Train set. Adaboost has higher MAE on the Test set. GRU and LSTM performs competitively but has slightly higher MAE values compared to the Ensemble.

To evaluate the performance of our predictive model, the predicted test vectors and actual test vectors were calculated by computing the magnitude of the nine chosen parameters. The residual value for each sample was computed as the absolute difference between the magnitude of the predicted vector for that sample and the magnitude of the actual vector for the same sample. Following the residual analysis, a specific static threshold value was set up for each model and anomalies were detected when the residuals exceeded these thresholds.

# 8 CONCLUSION

The proposed models built on a comprehensive analysis of driver behavior and road conditions, hold the potential to be a transformative step towards enhancing road safety and the overall driving experience in India. The data-driven approach, leveraging five models: LSTM-R, Ensemble of Multilayer Perceptron(MLP) and Random Forest, Adaboost Regressor, XGBoost regressor and GRU offering a robust foundation for this endeavor.

In conclusion, the models provide a potential to enhance road safety in India substantially. The data-driven, AI-powered approach, combined with the integration of various data sources, makes it a valuable tool for both individual drivers and authorities responsible for road safety. It represents a vital step towards a safer and more efficient road ecosystem in India.

# REFERENCES


[1] Ministry of Road Transport and Highways. Road accidents in india. pages 1163–1168, 2023. URL: https://morth.nic.in/sites/default/files/RA_2021_Compressed.pdf, doi:11Apr2023.

[2] Xian-Hong Huang, Chen Zhi-Hao, Ahamad Afaroj, and Sun Chi-Chia. Adas e-bike: Auxiliary adas module for electric power-assisted bicycle. pages 1–2, 2022. doi:10.1109/IET-ICETA56553.2022.9971513.

[3] B. Ball and Tang J.E. Machine learning and embedded computing in advanced driver assistance systems (adas). June 2019. URL: https://www.mdpi.com/2079-9292/8/7/748, doi:10.3390/electronics8070748.

[4] De-Las-Heras G., Sánchez-Soriano J., and Puertas E. Advanced driver assistance systems (adas) based on machine learning techniques for the detection and transcription of variable message signs on roads. July 2021. URL: https://www.mdpi.com/1424-8220/21/17/5866, doi:10.3390/s21175866.

[5] Tselentis Dimitrios I., Eleonora Papadimitriou, and Pieter van Gelder. The usefulness of artificial intelligence for safety assessment of different transport modes. 2023. URL: https://www.sciencedirect.com/science/article/pii/S0001457523000817?via%3Dihub, doi:10.1016/j.aap.2023.107034.

[6] Yongfeng Ma, Zhuopeng Xie, Shuyan Chen, Fengxiang Qiao, and Zeyang Li. Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals. 2023. URL: http://www.sciencedirect.com/science/article/pii/S0968090X22003965, doi:10.1016/j.trc.2022.103983.

[7] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li. D3: Abnormal driving behaviors detection and identification using smartphone sensors. 2015. URL: https://ieeexplore.ieee.org/document/7338354, doi:http://dx.doi.org/10.1109/SAHCN.2015.7338354.

[8] G.K. Sahoo, S.K. Das, and P. Singh. Two-layered gated recurrent stacked long short-term memory networks for driver's behavior analysis. *Sādhanā*, 48, 2023. URL: https://www.ias.ac.in/article/fulltext/sadh, doi:10.1007/s12046-023-02126-y.

[9] R. Kumar and A. Jain. Driving behavior analysis and classification by vehicle obd data using machine learning. *Journal of Supercomputing*, 79:18800–18819, 2023. doi:10.1007/s11227-023-05364-3.

[10] Xiupeng Shi, Yiik Wong, Michael Li, Chandrasekar Palanisamy, and Chen Chai. A feature learning approach based on xgboost for driving assessment and risk prediction. *Accident Analysis & Prevention*, 129:170–179, 2019. URL: https://www.sciencedirect.com/science/article/abs/pii, doi:10.1016/j.aap.2019.05.005.

[11] Mehdi Ghatee Amin Fattahi, Amir Golroo. Driver behavior assessment using multi-layer perceptron and random forest via smartphone sensors and obd ii. pages 1163–1168, 2023. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4416107, doi:11Apr2023.

[12] Nan Ding, HaoXuan Ma, Huanbo Gao, YanHua Ma, and GuoZhen Tan. Real-time anomaly detection based on long short-term memory and gaussian mixture model. *Computers & Electrical Engineering*, 79(1):106458, January 2021. URL: https://www.sciencedirect.com/science/article/pii/S0045790618334372, doi:10.1016/j.compeleceng.2019.106458.

[13] Hassan Bousnguar, Amal Battou, and Lotfi Najdi. Gated recurrent units (gru) for time series forecasting in higher education. *International Journal of Engineering Research & Technology (IJERT)*, 12(03):To be determined, March 2023. doi:10.17577/IJERTV12IS030091.

[14] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. pages 1597–1600, 2017. doi:10.1109/MWSCAS.2017.8053243.

[15] Devon Barrow. A comparison of adaboost algorithms for time series forecast combination. *International Journal of Forecasting*, 32:1103–1119, 06 2016. doi:10.1016/j.ijforecast.2016.01.006.

[16] Zeravan Ali, Ziyad Abduljabbar, Hanan Tahir, Amira Sallow, and Saman Almufti. Exploring the power of extreme gradient boosting algorithm in machine learning: a review. 12:320–334, 05 2023. doi:10.25007/ajnu.v12n2a1612.

[17] BabekHassa Erdebilli and Burcu Devrim-İçtenbaş. Ensemble voting regression based on machine learning for predicting medical waste: A case from turkey. *Mathematics*, 10(14), 2022. URL: https://www.mdpi.com/2227-7390/10/14/2466, doi:10.3390/math10142466.

[18] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Inteligence*, 4:26–30, 01 2016. doi:10.9781/ijimai.2016.415.

[19] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31:2225–2236, 10 2010. doi:10.1016/j.patrec.2010.03.014.

[20] Fereidoon Moghadas Nejad Mehrdad Ehsani and Pouria Hajikarimi. Developing an optimized faulting prediction model in jointed plain concrete pavement using artificial neural networks and random forest methods. *International Journal of Pavement Engineering*, 24(2):2057975, 2023. arXiv:https://doi.org/10.1080/10298436.2022.2057975, doi:10.1080/10298436.2022.2057975.

[21] Amir Samimi Mahdi Zarei Yazd, Iman Taheri Sarteshnizi and Majid Sarvi. A robust machine learning structure for driving events recognition using smartphone motion sensors. *Journal of Intelligent Transportation Systems*, 28(1):54–68, 2024. arXiv:https://doi.org/10.1080/15472450.2022.2101109, doi:10.1080/15472450.2022.2101109.

[22] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[23] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. *A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences*, pages 771–785. 08 2023. doi:10.1007/978-3-031-37963-5_53.

[24] Michael Phi. Illustrated guide to lstm's and gru's: A step by step explanation. 2018.

[25] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020. doi:10.1007/s10462-020-09838-1.

[26] Zeynep Kilimci. Ensemble regression-based gold price (xau/usd) prediction. *Journal of Emerging Market Finance*, 2:7, 06 2022.

[27] Y. Ding, Hongyang Zhu, Ruyun Chen, and Ronghui Li. An efficient adaboost algorithm with the multiple thresholds classification. *Applied Sciences*, 2022. URL: https://api.semanticscholar.org/CorpusID:249562694.

[28] Tu Chengsheng, Liu Huacheng, and Xu Bing. Adaboost typical algorithm and its application research. *MATEC Web of Conferences*, 139:00222, 01 2017. doi:10.1051/matecconf/201713900222.

[29] Muhammad Adnan, Alaa Alarood, M. Irfan Uddin, and Izaz Rehman. Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. *PeerJ Computer Science*, 8:e803, 02 2022. doi:10.7717/peerj-cs.803.

[30] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, 2016. doi:10.1145/2939672.2939785.

[31] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of xgboost, 11 2019.

[32] Sirilak Areerachakul and Siripun Sanguansintukul. Classification and regression trees and mlp neural network to classify water quality of canals in bangkok, thailand. 2010. URL: https://api.semanticscholar.org/CorpusID:132729585.

[33] S.Ridella D. Anguita and F.Rivieccio. K-folds generalization capability assessment for support vector classifiers. URL: https://api.semanticscholar.org/CorpusID:132729585.

[34] Li hua Chen and Xiao yun Zhang. Application of artificial neural network to classify water quality of the yellow river. URL: https://api.semanticscholar.org/CorpusID:132729585.

[35] S.Areerachakul and S.Sanguansintukul. "water classification using neural network: A case study of canals in bangokok, thailand. URL: https://api.semanticscholar.org/CorpusID:132729585.

# Comparative Analysis of Machine Learning Models

*by* Sania Khan

# Comparative Analysis of Machine Learning Models for classifying abnormal driving behavior in two-wheeler kinematics

Nupur Giri
nupur.giri@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Yogesh Pandit
yogesh.pandit@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Simran Ahuja
2021.simran.ahuja@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Sania Khan
2021.sania.khan@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Jesica Bijju
2021.jesica.bijju@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

Sejal Datir
2021.sejal.datir@ves.ac.in
VES Institute of Technology
Mumbai, Maharashtra, India

**ACM Reference Format:**
Nupur Giri, Yogesh Pandit, Simran Ahuja, Sania Khan, Jesica Bijju, and Sejal Datir. 2024. Comparative Analysis of Machine Learning Models for classifying abnormal driving behavior in two-wheeler kinematics. In . , 7 pages.

## 1 ABSTRACT

This research paper focuses on the driving behaviour of two wheeler riders who are particularly vulnerable to road accidents. Through the analysis of time-series data collected from real-world driving scenarios, the study aims to develop machine learning models for risk assessment and anomaly detection. The paper starts by outlining the alarming statistics of road accidents in India, emphasizing the urgent need for effective solutions. The study explores the effectiveness of five distinct models: Long Short-Term Memory -Residual (LSTM-R), Gated Recurrent Unit (GRU), Adaboost , XGBoost, Multilayer Perceptron (MLP) with Random Forest (RF) ensemble.

## 2 INTRODUCTION

In recent years, India has witnessed an escalation in traffic accidents. The Ministry of Road Transport and Highways' annual report for the fiscal year 2021–2022 shows a startling statistic: there were 4,12,432 road accidents registered nationwide, resulting in 1,53,972 fatalities and 3,84,448 injuries. According to NH road accident and fatality data, two-wheelers accounted for the most number of accidents (52,416) and deaths (22,786) in 2021 [1]. The leading cause of more than half of these accidents was overspeeding, with dangerous and careless driving following closely. These alarming statistics highlight the importance of developing effective solutions to reduce road accidents and elevate road safety standards.

In response to this critical and escalating public health challenge, our research aims to address the urgent need for enhanced road safety measures, especially for two-wheeler riders. Through careful examination of the actions and habits of two-wheeler drivers, our research endeavors to uncover the factors contributing to accidents.

## 3 LITERATURE REVIEW

One of the primary causes of accidents on the roads is abnormal driving behaviour. Most studies on anomalous driving behaviour classify reconstruction or prediction residuals using unsupervised learning techniques, or they identify and evaluate the driving state using direct classification algorithms. Data on unusual driving behaviour are challenging to find and categorise. Due to this limited data of the anomalous driving behavior, there is an inherent class imbalance problem in the algorithm training process. Furthermore, the majority of the existing residual analysis studies concentrate on single points, which makes it difficult to capture the continuous feature of abnormal driving behavior.

Several papers have been proposed on Advanced Driver Assistance System that help drivers in routine navigation by leveraging computer networks to enable more data-driven and safer driving experiences. Advanced Driver Assistance Systems (ADAS) module that is based on AI neural networks is proposed in [2] to improve road safety. John et al. in [3] presents Advanced machine learning algorithms based ADAS system to detect objects, obstacles, other vehicles, pedestrians, and lanes, and the estimation of object trajectories and intents. The work described in [4] introduces a functional prototype of a machine learning-driven system for Variable Message Sign (VMS) interpretation. This system is capable of recognizing traffic signals, extracting their textual information, and converting it into spoken output..

Paper [5] proposes crash prediction based on, incident/failure detection, pattern identification, driver/operator or route assistance, using ANN, SVM, Hidden Markov Models and Bayesian models. In Paper [6], YongFeng Ma et al. suggests a long short-term memory-residual (LSTM-R) algorithm to identify abnormal driving behavior. This algorithm consists of two parts: firstly, an LSTM network is used to analyze historical vehicle kinematic data, yielding root mean square residuals at each time point. Secondly, a time window-based residual algorithm is applied to identify abnormal driving behaviors, calculating the magnitude and continuity of these residuals. Paper[7] presents real-time high-accurate abnormal driving behaviors monitoring using smartphone sensors and uses SVM. Paper [8] implements a two-layer GRU stacked long short-term memory (LSTM) network to categorize driving characteristics. OBD-II protocol was implemented to collect real-time vehicle performance in paper [9] using SVM, AdaBoost, and Random Forest. Paper [10] utilizes XGBoost to create associations between behavior features and risk levels. Random Forest for feature selection and Multilayer Perceptron (MLP) neural network for maneuver classification in Paper [11] achieving an 89% overall accuracy in identifying aggressive driving maneuvers using smartphone sensors and OBD

*We acknowledge IIIT-H/IHub/Project/Mobility/2023-24/M2-018 for financial assistance, 2024*
.

II data. Paper [12] proposes the LGMAD, a real-time anomaly detection algorithm based on Long-Short Term Memory (LSTM) and Gaussian Mixture Model (GMM) whereas paper [13] explores the use of a gated neural network for time series forecasting in the context of higher education. Three variations of the Gated Recurrent Unit (GRU) evaluated in paper [14] by retaining the structure and methodically decreasing parameters in the update and reset gates. Rigorous empirical evaluations were conducted in paper [15] of seven variants of the popular AdaBoost algorithm. Paper [16] uses one of the most well-known machine learning algorithms for supervised and semi-supervised learning (SSL), such as XGBoost, in a Python environment. The paper referenced by [17] introduces an ensemble voting regression algorithm leveraging machine learning techniques like random forests, gradient boosting machines, and adaptive boosting. Paper [18] introduces a new approach to optimize the network architecture of a MLP by using the genetic algorithm and a back-propagation algorithm. while paper [19] focuses on random forests to provide some experimental insights about the behavior of the variable importance index. Paper [20] has used random forest and artificial neural network techniques to create an optimal faulting prediction model for jointed plain concrete pavement. A two-phase Machine Learning (ML) method using high-pass, low-pass, and wavelet filters to detect driving brakes and turns using Random Forest and Artificial Neural Network classifiers was proposed in paper [21]. Article [22] discusses how random forests, consisting of tree predictors, make use of random feature selection to create an ensemble that converges to a stable generalization error. In the paper[23], the memorization techniques of RNN's, specifically LSTM and GRU was explored concluding that an increase in depth does not always improve memorization. The article [24] provides a guide to Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), explaining their internal mechanisms and role in mitigating short-term memory issues in recurrent neural networks. The paper [25], highlights the impact of LSTM on machine learning and neuro-computing. A novel approach to forecast the value of gold (XAU/USD) using a regression ensemble, linear regression, decision tree regression, and stacking regression, with stacking regression was discussed in paper [26].

## 4 NOVELTY OF WORK

Data was systematically collected from a consistent route over seven rides, covering a 4 km distance. Nine parameters were recorded: Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch using the SensorLog iOS mobile application. The methodology contributes in addressing the multi-output regression tasks involving time-series data. We trained LSTM-R, GRU, AdaBoost, XGBoost, and Ensemble of MLP and Random Forest for prediction, aiming to understand connections among different sensor variables simultaneously. We integrated information from accelerometers, gyroscopes, and motion angles, emphasizing a direct combination of insights from various sensors. The evaluation process involved visualizing predicted versus actual values, calculating RMSE, and conducting a residual analysis, providing a practical understanding of the model's performance and areas for improvement.

## 5 METHODOLOGY

1. A series of test drives was conducted over a 4 km route in Chembur, Mumbai, utilizing Sensorlog, an iOS application, to collect vehicle kinematic data in real time with a sampling frequency of 10 Hz. The data collection process involved two individuals: a rider and a passenger equipped with a handheld iOS mobile device running the Sensorlog app. The application continuously recorded the rider's movements and behavior, capturing parameters including Accelerometer (X, Y, Z), Gyro Rotation (X, Y, Z), Motion Yaw, Motion Roll, and Motion Pitch of the bike. Additionally, Magnetometer readings (X, Y, Z) were initially included but later excluded due to redundancy. The data collection occurred over a period of 7 days, during which 81,480 tuples of data were gathered, with each ride averaging 20 minutes in duration. The dataset encompassed data collected from both normal driving behavior and rash driving behavior exhibited on the same route.

2. Preprocessing: Before fitting the model, the input features are reshaped for all models. The reshaping is performed to flatten the input features, converting the 2D array into a 1D array, making it compatible with the model's expectations. The dataset was split into training, validation, and test sets.

3. Model selection: Five models were trained on the nine parameters. Neural network models- LSTM-R and GRU were trained first, followed by the boosting algorithms- AdaBoost and XGBoost. The final model that was trained was a combination of both neural networks and boosting- the ensemble of MLP with RF.

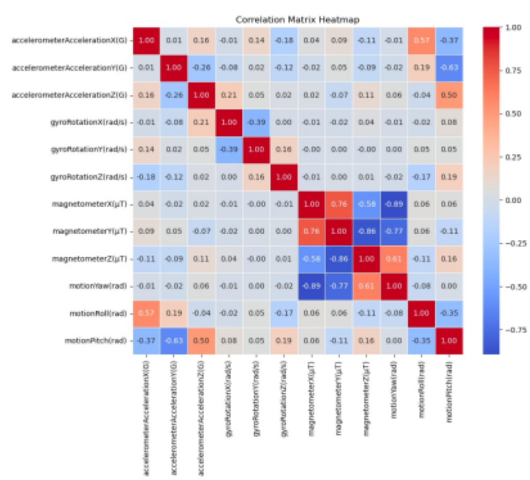



Fig. 1 : Confusion matrix

Fig. 1 depicts the correlation between the recorded parameters

## 6 MODEL TRAINING

The model creation process involves the training of five models: long short-term memory-residual (LSTM-R), gated recurrent unit (GRU), AdaBoost, XGBoost, and ensemble of multilayer perceptron (MLP) with random forest. The dataset is split into three parts: 70

% for training, 15 % for testing , and 15 % for validation. The input layer is structured to accept data in windows of a specific size.

## 6.1 LSTM - Residual:

6.1.1 LSTM

LSTM is a type of recurrent neural network (RNN) that analyzes sequential data [12]. LSTM's ability to capture temporal dependencies made it a suitable choice for the two wheeler time-series dataset used in this study. In an LSTM network, there are three gates (input, forget, and output gates) that regulate the flow of information, and the memory cell stores long-term dependencies[25].

6.1.2 LSTM-Residual (LSTM-R)

The LSTM-R model calculates residuals, which are the differences between the observed data and the data predicted by the model. It uses a time window-based approach, with the residual values computed over a window of 100 milliseconds. The LSTM units act as intelligent memory cells, recognizing patterns across multiple time steps as data patterns evolve over various temporal scales.
In this model, layers are organized sequentially which begins with an input layer to handle sequences with nine features. The core is an LSTM layer with 64 units, designed to capture patterns over time. After that, a Dense layer with eight units and Rectified Linear Unit (ReLU) activation is added for non-linear transformations. The final layer is another Dense layer with nine units, using a linear activation.

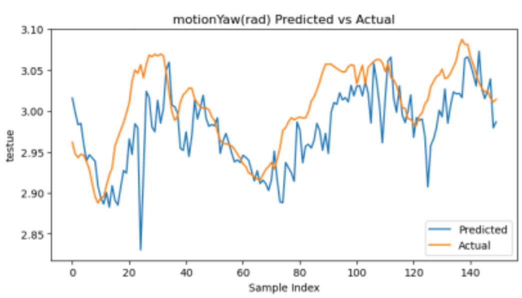



Fig. 2 : Motion Yaw Predicted vs Actual

Fig. 2 depicts the Actual vs. Predicted values of Motion Yaw(rad) trained on LSTM-R model.

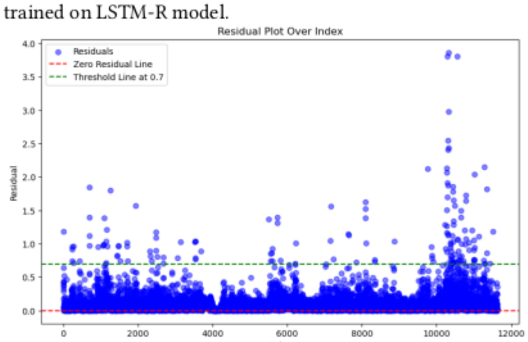



Fig. 3 : Residual Plot

Fig. 3 depicts the residual values for the LSTM-R model.

## 6.2 GRU

Following the training of LSTM-R, Gated Recurrent Units (GRU) was chosen for its simpler architecture. GRU has two gates: reset and update gate as opposed to LSTM which has three gates[23]. GRU is a type of RNN like LSTM with a potential for faster training[13]. This choice aims to explore the strengths of both models and improve performance on the time series dataset.
GRU consists of parameters that control the memory states and aims to solve the exploding gradient problem of traditional RNNs.[14] It involves a gating mechanism that helps selectively update and reset the hidden states. The reset gate controls the information from the previous time step to be forgotten, while the update gate regulates how much of the new information should be added to the cell state.
The GRU architecture consists of an input layer and uses a time window-based approach. The structure of the GRU model was determined through repeated testing and comparison of the number of hidden layers, activation functions, number of units in each layer, batch size, dropout rate and epochs.
The model incorporated three GRU hidden layers with decreasing units 64, 32, and 16 respectively to capture sequential dependencies. This structure is further extended with two dense layers to map the extracted features from the hidden layers to the output space with 9 units each, one that uses the Rectified Linear Unit(ReLU) and the other that uses the Leaky Rectified Linear Unit(LeakyReLU) activation function. The addition of dropout layers as a regularization technique was avoided since GRU layers already have internal mechanisms in the form of reset and update gates to control the information flow and mitigate vanishing gradient problems, thereby avoiding overfitting. [24]
For model optimization and training, we utilized Mean Absolute Error (MAE) as the loss function, paired with the Adam optimizer set at a learning rate of 0.001.

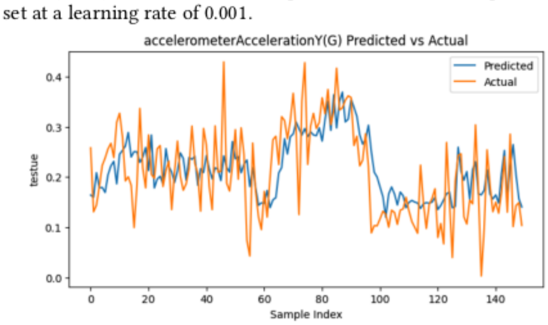



Fig. 4 : AccelerometerY Predicted vs Actual

Figure 4 compares the Actual and the Predicted values of AccelerometerY trained on GRU.

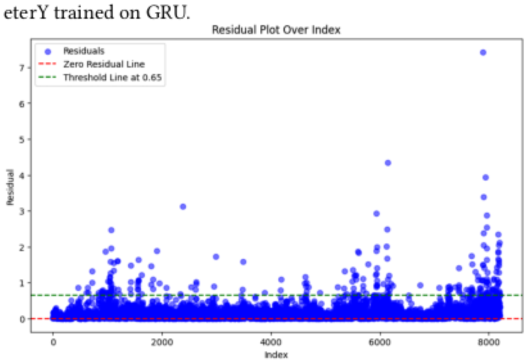



Fig. 5 : Residual Plot

Figure 5 showcases the Residual over time offering an insight to the anomalies from the expected behavior for GRU.

### 6.3 AdaBoost

After training the model with neural networks like LSTM and GRU, the dataset was trained on boosting algorithms to see how they perform in comparison to neural networks for the multioutput regression task. The Adaboost model follows the boosting ensemble learning method. It makes use of 'adaboost regressor' as the weak learner and Decision Trees as the default base estimator [27]. It is then followed by Weighted Training where weights are assigned to each training instance. Initially, all weights are set equally. However, during the training process, the weights are adjusted based on the performance of the weak learners. Misclassified instances are given higher weights to focus on them during subsequent iterations. Weak learners are then trained sequentially. Each subsequent learner focuses more on the instances that were misclassified by the previous learners [28]. This is performed for each output variable and a new adaboost regressor is trained for each variable. The final prediction is a weighted sum of the predictions from individual weak learners. The weights are determined based on the performance of each learner.

Grid Search CV was employed to discover the optimal hyperparameters for the model, utilizing a 'param grid' dictionary, an Adaboost Regressor as the base model, and a Grid Search CV object [29]. The param grid contained various combinations of hyperparameter values, evaluating each set to determine the most effective configuration. Notably, the grid focused on three key hyperparameters: number of estimators (number of weak learners), learning rate (scaling factor for each weak learner's contribution), and loss (the function impacting weight updates and weak learner performance measurement). The choice of loss function significantly influenced the boosting algorithm's behavior. The Adaboost Regressor served as the base model for the GridSearch CV, which, employing four-fold cross-validation, explored the parameter grid for detailed output. The best parameters identified through this process were number of estimators equal to 100, learning rate of 0.01, and loss set to 'exponential.'

After training, the Adaboost Regressor is used to make predictions on the training, validation, and test sets. The Root Mean Square Error (RMSE) values are calculated for each parameter and Mean Absolute Error (MAE) is calculated for each set by comparing the predicted values to the actual labels. This metric is used to evaluate the model's performance on different subsets of the data.

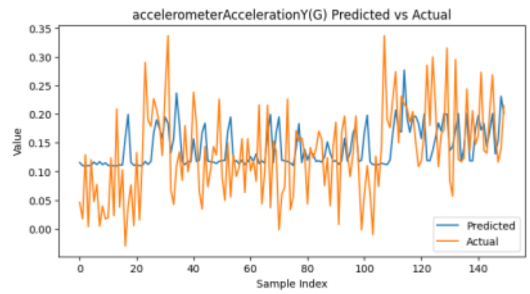



Fig. 6 : Accelerometer(Y) Predicted vs Actual

Figure 6 compares the predicted values generated by the ensemble of AdaBoost regressors with the actual values for motion pitch.

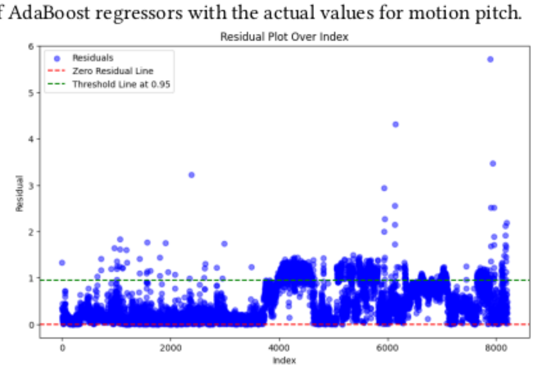



Fig. 7 : Residual Plot

### 6.4 XGBoost

Figure 10 shows the actual vs predicted graph for Adaboost model. It was observed that the predictions did not fit well with the actual data points. This observation prompted a transition to XGBoost, a gradient boosting algorithm, known for its sequential tree construction [30]. XGBoost offers distinct advantages, including effective regularization methods and enhanced control over fine-tuning—features absent in Adaboost [16].

The XGBoost model adopts a structured approach, processing data in specific windows. This strategy proves effective in isolating and characterizing driving events based on their duration, providing valuable contextual insights for analysis and intervention [31]. XGBoost iteratively constructs a sequence of decision trees, each addressing the errors of its predecessor. This iterative process steadily enhances the model's predictive capabilities. Central to XGBoost's functionality is its foundation in gradient boosting, a methodology that optimizes a loss function through the calculation of gradients

concerning the model's predictions, this approach allows the algorithm to make corrections that minimize overall prediction errors, contributing to the model's improved accuracy.
Hyperparameters were experimented with to fine-tune the configuration of the XGBoost model. The objective was to identify the optimal settings that would minimize the Root Mean Square Error (RMSE) values, ensuring the model's ability to make accurate and reliable predictions. Specifically, the selected key hyperparameters were- learning rate set at 0.2, maximum depth of 3 for each individual tree, alpha value of 10 for regularization strength , and 100 decision trees in the ensemble.

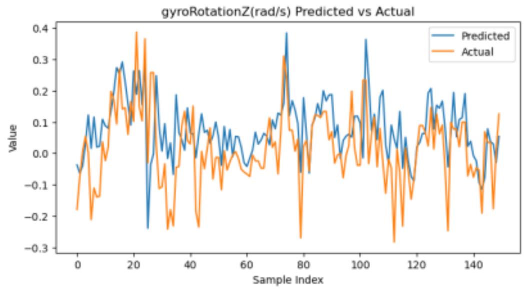



FIG. 8 : GYROROTATIONZ PREDICTED VS ACTUAL

Figure 8 depicts the Actual vs. Predicted values of Motion Roll, GyroRotationZ.

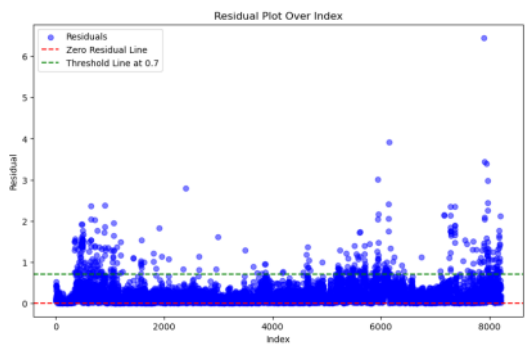



FIG. 9 : RESIDUAL PLOT

## 6.5 Ensemble of Multilayer Perceptron(MLP) and Random Forest:

After a separate analysis of neural networks and boosting algorithms, training a model on the combination of both could offer a different perspective.

### 6.5.1 *Multilayer Perceptron Neural Network:*

The Multilayer Perceptron has a large wide of classification and regression applications in many fields: pattern recognition, voice and classification problems. [18]. It is one of the most common and practical artificial neural networks in which each neuron is connected to several neighbor neurons [11]. A general network consists of a layered architecture, an input layer, one or more hidden layers and an output layer [32]. A common artificial neural network used to address a variety of issues, such as pattern recognition and interpolation, is the Multilayer Perceptron (MLP) [33][34]. Neurons make up each layer, and they are related to one another via weights. A unique mathematical function known as the activation function is present in every neuron and uses information from earlier levels to produce output for the layer after it. The hyperbolic tangent sigmoid transfer function is the activation function employed in the experiment [35].

### 6.5.2 *Random Forest:*

The random forest algorithm is commonly used to handle regression and classification problems [20]. It can be defined as a regression technique that combines the performance of numerous decision trees independently and then averaging their predictions for the final outcome [22]. Random Forests offer efficient training and prediction processes from a computational perspective, relying on just a couple of tuning parameters, and can easily be implemented in parallel. Statistically, Random Forests provide additional features such as the measures of variable importance, differential class weighting. Each tree in the random forest is created based on a random subset of the input variables, and each tree branching is created based on a random number of selected input features. [11]

### 6.5.3 *Ensemble of MLP with Random Forest:*

The Ensemble model was created using the combination of Multi-Layer Perceptron as the neural network, and Random Forest for the boosting. A voting ensemble employs multiple models instead of a single one to enhance the system's performance, as discussed in [17]. Regressor voting is beneficial for comparing many regression models and selecting the one that produces the best predictions. The predictions from all models are combined through either average voting (AV), or weighted voting (WV)[17]. The Voting Regressor, which combines regression models of MLP and RF by assigning weights to each model, developed a weighted strategy in this case so that the prediction results are based on the results of the selection of these models. The ensemble model is then implemented as a MultiOutputRegressor, allowing it to handle multi-output regression tasks.
Determining the optimal configuration of the MLP, including the number of hidden layers, the number of neurons, the type of activation function, and the optimization method was done through trial and error [19]. The MLP network was trained with the ReLU activation function and the Adam optimizer. After trial and error, the best neural network architecture was obtained with 3 hidden layers with 100, 50, and 25 neurons in each hidden layer respectively. Other hyperparameters include an L2 regularization term of 0.0001, a constant learning rate of 0.001, and a batch size that dynamically adjusts during training.
The RandomForestRegressor consists of 200 decision trees, employing the Mean Squared Error criterion. The ensemble model, created using VotingRegressor, combines predictions from both models. The model's evaluation involves assessing its performance on both training and test datasets using metrics such as Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

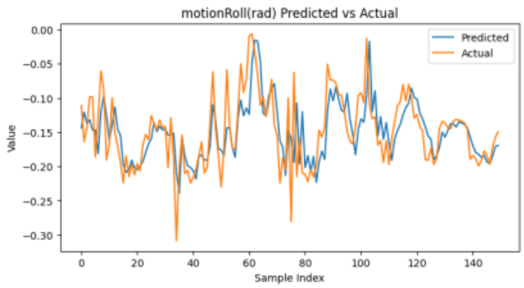



Fig. 10 : Motion Roll Predicted vs Actual

Figures 10 depicts the Actual vs. Predicted values of Motion Roll parameter generated by the Ensemble of MLP and Random Forest.

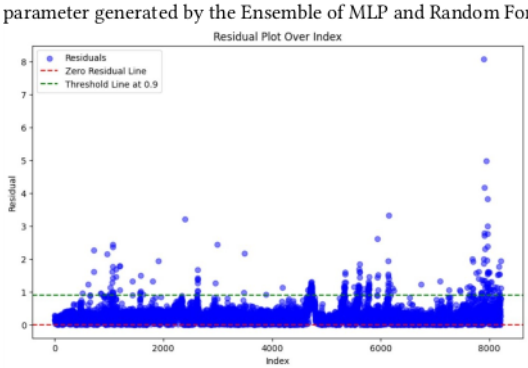



Fig. 11 : Residual Plot

Figure 11 showcases the Residual over time of the MLP-Random Forest Ensemble

| Models | Parameters | Parameter Range | Optimal Parameters |
|---|---|---|---|
| MLP - RF Ensemble | number of hidden layers $N_l$<br>number of neurons $N_n$<br>activation layers $A_l$<br>optimizer $O$<br>number of estimators $N_e$<br>learning rate $L_r$ | $N_l \in \{1, 2, 3\}$<br>$N_n \in$ {100, (20,20), (100,50,25)}<br>$A_l \in$ {relu, logistic, hyperbolic tan}<br>$O \in$ {adam, sgd }<br>$N_e \in$ {100,200,300,400,500}<br>$L_r \in \{0.001, 0.0001\}$ | $N_l = 3$<br>$N_n$ = (100,50,25)<br>$A_l$ = relu<br>$O$ = Adam<br>$N_e = 200$<br>$L_r = 0.001$ |
| Adaboost | number of estimators $N_e$<br>learning rate $L_r$<br>loss $L$ | $N_e \in \{50, 100, 200\}$<br>$L_r \in \{0.001, 0.01, 0.1, 0.2\}$<br>$L \in$ {linear ,square, exponential} | $N_e = 100$<br>$L_r = 0.01$<br>$L$ = exponential |
| XGBoost | objective $O_b$<br>colsample bytree $N_t$<br>learning rate $L_r$<br>maximum depth $N_d$<br>alpha $\alpha$<br>number of estimators $N_e$ | $O_b \in$ {reg:squarederror, reg:logistic, reg:gamma}<br>$N_t \in$ {[0.1 - 1.0]}<br>$L_r \in$ {[0.01 - 0.3]}<br>$N_d \in \{2, 3, 4, 5, 6\}$<br>$\alpha \in$ {[1 - 20]}<br>$N_e \in$ {[50 - 200]} | $O_b$ = reg:squarederror<br>$N_t = 0.8$<br>$L_r = 0.2$<br>$N_d = 3$<br>$\alpha = 10$<br>$N_e = 100$ |
| GRU | number of units $N_u$<br>learning rate $L_r$<br>number of hidden layers $N_l$<br>activation layers $A_l$<br>batch size $m$ | $N_u \in \{8, 16, 32, 64, 128\}$<br>$L_r \in \{0.001, 0.0001\}$<br>$N_l \in \{2, 3, 4\}$<br>$A_l \in$ {relu, leakyrelu, softmax, tanh}<br>$m \in \{0, 16, 32\}$ | $N_u$ = 16, 32, 64<br>$L_r = 0.001$<br>$N_l = 3$<br>$A_l$ = relu, leakyrelu<br>$m = 0$ |
| Residual LSTM | number of units $N_u$<br>learning rate $L_r$<br>activation layers $A_l$<br>batch size $m$ | $N_u \in \{16, 32, 64, 128\}$<br>$L_r \in \{0.001, 0.0001\}$<br>$A_l \in$ {relu, linear, softmax, tanh}<br>$m \in \{0, 16, 32\}$ | $N_u = 64$<br>$L_r = 0.0001$<br>$A_l$ = relu, linear<br>$m = 0$ |

Table 1 : RMSE Values for multi output regression for all models

## 7 MODEL EVALUATION

| RMSE Values for Multi Output Regression of all models | | | | | |
|---|---|---|---|---|---|
| Parameters | Models | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Accelerometer X | **0.0485** | 0.0755 | 0.0636 | 0.0499 | 0.1191 |
| Accelerometer Y | **0.0388** | 0.0944 | 0.0811 | 0.0611 | 0.0913 |
| Accelerometer Z | **0.0659** | 0.1465 | 0.1236 | 0.0903 | 0.1132 |
| Gyro Rotation X | **0.1019** | 0.2402 | 0.2043 | 0.1209 | 0.1824 |
| Gyro Rotation Y | **0.1454** | 0.3465 | 0.2850 | 0.1855 | 0.2859 |
| Gyro Rotation Z | 0.1007 | 0.1522 | 0.1057 | **0.0717** | 0.1074 |
| Motion Yaw | 0.0179 | 0.2339 | 0.1371 | **0.0073** | 0.0645 |
| Motion Roll | 0.0182 | 0.0246 | 0.0195 | **0.0087** | 0.0624 |
| Motion Pitch | 0.0195 | 0.0204 | 0.0299 | **0.0087** | 0.0243 |

Table 2 : RMSE Values for multi output regression for all models

Table-2 represents RMSE values for all the parameters with respect to each model for multi output regression. The MLP with RF Ensemble model performs best for Accelerometer(X, Y, Z)and Gyrorotation (X, Y). Meanwhile GRU demonstrates proficiency in capturing patterns for Gyro Rotation Z, Motion Yaw, Motion Pitch and Motion Roll, highlighting its suitability for motion-related features. All the five models measure better predictions for Motion Roll and Motion Pitch.

| MAE Values for Train, Validation and Test Samples | | | | | |
|---|---|---|---|---|---|
| Mean Absolute Error | Models | | | | |
| | MLP - RF Ensemble | Adaboost | XGBoost | GRU | Residual LSTM |
| Train | 0.0812 | 0.0914 | **0.0723** | 0.0735 | 0.0857 |
| Validation | **0.0477** | 0.0953 | 0.0839 | 0.0759 | 0.0876 |
| Test | **0.1157** | 0.1422 | 0.1325 | 0.1175 | 0.1179 |

Table 2 : MAE values for train, validation and test samples

The MLP with RF Ensemble model performs better than the other models in terms of Mean Absolute Error (MAE) across Validation, and Test samples, while XGBoost performs better on the Train set. Adaboost has higher MAE on the Test set. GRU and LSTM performs competitively but has slightly higher MAE values compared to the Ensemble.

To evaluate the performance of our predictive model, the predicted test vectors and actual test vectors were calculated by computing the magnitude of the nine chosen parameters. The residual value for each sample was computed as the absolute difference between the magnitude of the predicted vector for that sample and the magnitude of the actual vector for the same sample. Following the residual analysis, a specific static threshold value was set up for each model and anomalies were detected when the residuals exceeded these thresholds.

## 8 CONCLUSION

The proposed models built on a comprehensive analysis of driver behavior and road conditions, hold the potential to be a transformative step towards enhancing road safety and the overall driving experience in India. The data-driven approach, leveraging five models: LSTM-R, Ensemble of Multilayer Perceptron(MLP) and Random Forest, Adaboost Regressor, XGBoost regressor and GRU offering a robust foundation for this endeavor.

In conclusion, the models provide a potential to enhance road safety in India substantially. The data-driven, AI-powered approach, combined with the integration of various data sources, makes it a valuable tool for both individual drivers and authorities responsible for road safety. It represents a vital step towards a safer and more efficient road ecosystem in India.

## REFERENCES

[1] Ministry of Road Transport and Highways. Road accidents in india. pages 1163–1168, 2023. URL: https://morth.nic.in/sites/default/files/RA_2021_Compressed.pdf, doi:11Apr2023.

[2] Xian-Hong Huang, Chen Zhi-Hao, Ahamad Afaroj, and Sun Chi-Chia. Adas e-bike: Auxiliary adas module for electric power-assisted bicycle. pages 1–2, 2022. doi:10.1109/IET-ICETA56553.2022.9971513.

[3] B. Ball and Tang J.E. Machine learning and embedded computing in advanced driver assistance systems (adas). June 2019. URL: https://www.mdpi.com/2079-9292/8/7/748, doi:10.3390/electronics8070748.

[4] De-Las-Heras G., Sánchez-Soriano J., and Puertas E. Advanced driver assistance systems (adas) based on machine learning techniques for the detection and transcription of variable message signs on roads. July 2021. URL: https://www.mdpi.com/1424-8220/21/17/5866, doi:10.3390/s21175866.

[5] Tselentis Dimitrios I., Eleonora Papadimitriou, and Pieter van Gelder. The usefulness of artificial intelligence for safety assessment of different transport modes. 2023. URL: https://www.sciencedirect.com/science/article/pii/S0001457523000817?via%3Dihub, doi:10.1016/j.aap.2023.107034.

[6] Yongfeng Ma, Zhuopeng Xie, Shuyan Chen, Fengxiang Qiao, and Zeyang Li. Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals. 2023. URL: http://www.sciencedirect.com/science/article/pii/S0968090X22003965, doi:10.1016/j.trc.2022.103983.

[7] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li. D3: Abnormal driving behaviors detection and identification using smartphone sensors. 2015. URL: https://ieeexplore.ieee.org/document/7338354, doi:http://dx.doi.org/10.1109/SAHCN.2015.7338354.

[8] G.K. Sahoo, S.K. Das, and P. Singh. Two-layered gated recurrent stacked long short-term memory networks for driver's behavior analysis. *Sādhanā*, 48, 2023. URL: https://www.ias.ac.in/article/fulltext/sadh, doi:10.1007/s12046-023-02126-y.

[9] R. Kumar and A. Jain. Driving behavior analysis and classification by vehicle obd data using machine learning. *Journal of Supercomputing*, 79:18800–18819, 2023. doi:10.1007/s11227-023-05364-3.

[10] Xiupeng Shi, Yiik Wong, Michael Li, Chandrasekar Palanisamy, and Chen Chai. A feature learning approach based on xgboost for driving assessment and risk prediction. *Accident Analysis & Prevention*, 129:170–179, 2019. URL: https://www.sciencedirect.com/science/article/abs/pii, doi:10.1016/j.aap.2019.05.005.

[11] Mehdi Ghatee Amin Fattahi, Amir Golroo. Driver behavior assessment using multi-layer perceptron and random forest via smartphone sensors and obd ii. pages 1163–1168, 2023. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4416107, doi:11Apr2023.

[12] Nan Ding, HaoXuan Ma, Huanbo Gao, YanHua Ma, and GuoZhen Tan. Real-time anomaly detection based on long short-term memory and gaussian mixture model. *Computers & Electrical Engineering*, 79(1):106458, January 2021. URL: https://www.sciencedirect.com/science/article/pii/S0045790618334372, doi:10.1016/j.compeleceng.2019.106458.

[13] Hassan Bousnguar, Amal Battou, and Lotfi Najdi. Gated recurrent units (gru) for time series forecasting in higher education. *International Journal of Engineering Research & Technology (IJERT)*, 12(03):To be determined, March 2023. doi:10.17577/IJERTV12IS030091.

[14] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. pages 1597–1600, 2017. doi:10.1109/MWSCAS.2017.8053243.

[15] Devon Barrow. A comparison of adaboost algorithms for time series forecast combination. *International Journal of Forecasting*, 32:1103–1119, 06 2016. doi:10.1016/j.ijforecast.2016.01.006.

[16] Zeravan Ali, Ziyad Abduljabbar, Hanan Tahir, Amira Sallow, and Saman Almufti. Exploring the power of extreme gradient boosting algorithm in machine learning: a review. 12:320–334, 05 2023. doi:10.25007/ajnu.v12n2a1612.

[17] BabekHassa Erdebilli and Burcu Devrim-İçtenbaş. Ensemble voting regression based on machine learning for predicting medical waste: A case from turkey. *Mathematics*, 10(14), 2022. URL: https://www.mdpi.com/2227-7390/10/14/2466, doi:10.3390/math10142466.

[18] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Inteligence*, 4:26–30, 01 2016. doi:10.9781/ijimai.2016.415.

[19] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31:2225–2236, 10 2010. doi:10.1016/j.patrec.2010.03.014.

[20] Fereidoon Moghadas Nejad Mehrdad Ehsani and Pouria Hajikarimi. Developing an optimized faulting prediction model in jointed plain concrete pavement using artificial neural networks and random forest methods. *International Journal of Pavement Engineering*, 24(2):2057975, 2023. arXiv:https://doi.org/10.1080/10298436.2022.2057975, doi:10.1080/10298436.2022.2057975.

[21] Amir Samimi Mahdi Zarei Yazd, Iman Taheri Sarteshnizi and Majid Sarvi. A robust machine learning structure for driving events recognition using smartphone motion sensors. *Journal of Intelligent Transportation Systems*, 28(1):54–68, 2024. arXiv:https://doi.org/10.1080/15472450.2022.2101109, doi:10.1080/15472450.2022.2101109.

[22] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[23] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. *A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences*, pages 771–785. 08 2023. doi:10.1007/978-3-031-37963-5_53.

[24] Michael Phi. Illustrated guide to lstm's and gru's: A step by step explanation. 2018.

[25] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 12 2020. doi:10.1007/s10462-020-09838-1.

[26] Zeynep Kilimci. Ensemble regression-based gold price (xau/usd) prediction. *Journal of Emerging Market Finance*, 2:7, 06 2022.

[27] Y. Ding, Hongyang Zhu, Ruyun Chen, and Ronghui Li. An efficient adaboost algorithm with the multiple thresholds classification. *Applied Sciences*, 2022. URL: https://api.semanticscholar.org/CorpusID:249562694.

[28] Tu Chengsheng, Liu Huacheng, and Xu Bing. Adaboost typical algorithm and its application research. *MATEC Web of Conferences*, 139:00222, 01 2017. doi:10.1051/matecconf/201713900222.

[29] Muhammad Adnan, Alaa Alarood, M. Irfan Uddin, and Izaz Rehman. Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models. *PeerJ Computer Science*, 8:e803, 02 2022. doi:10.7717/peerj-cs.803.

[30] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pages 785–794, 2016. doi:10.1145/2939672.2939785.

[31] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of xgboost, 11 2019.

[32] Sirilak Areerachakul and Siripun Sanguansintukul. Classification and regression trees and mlp neural network to classify water quality of canals in bangkok, thailand. 2010. URL: https://api.semanticscholar.org/CorpusID:132729585.

[33] S.Ridella D. Anguita and F.Rivieccio. K-folds generalization capability assessment for support vector classifiers. URL: https://api.semanticscholar.org/CorpusID:132729585.

[34] Li hua Chen and Xiao yun Zhang. Application of artificial neural network to classify water quality of the yellow river. URL: https://api.semanticscholar.org/CorpusID:132729585.

[35] S.Areerachakul and S.Sanguansintukul. "water classification using neural network: A case study of canals in bangokok, thailand. URL: https://api.semanticscholar.org/CorpusID:132729585.

# Comparative Analysis of Machine Learning Models

ORIGINALITY REPORT

| 8% | 7% | 10% | 1% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Vivek Kumar Paswan, Akash Chandra Gupta, Ayesha Choudhary. "Computer Vision and Deep Learning Based Framework for Road Scene and Surface Segmentation in Unstructured Environment", 2022 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2022<br>Publication | 1% |
| 2 | www.researchgate.net<br>Internet Source | 1% |
| 3 | trid.trb.org<br>Internet Source | 1% |
| 4 | fastercapital.com<br>Internet Source | 1% |
| 5 | eprints.ums.ac.id<br>Internet Source | 1% |
| 6 | Submitted to Asia Pacific University College of Technology and Innovation (UCTI)<br>Student Paper | 1% |

| | | |
|---|---|---|
| 7 | "IET-ICETA 2022 Conference Proceedings", 2022 IET International Conference on Engineering Technologies and Applications (IET-ICETA), 2022<br>Publication | 1 % |
| 8 | www.ijert.org<br>Internet Source | 1 % |
| 9 | aclanthology.org<br>Internet Source | 1 % |
| 10 | scholar.archive.org<br>Internet Source | 1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 1% |
| Exclude bibliography | On | | |

1

**Industry / Inhouse:**
**Research / Innovation:** **Project Evaluation Sheet 2023-24** **Class: D12 C**

Title of Project (Group no): 1 - Behavioral Analysis and Risk Assessment for two wheelers vehicles.

Group Members: Sejal Dati, Sania Khan, Simran Ahuja, Jesica Bijju

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environ ment Friendly (2) | Ethics (2) | Team work (2) | Presentati on Skills (3) | Applied Engg & Mgmt principles (3) | Life - long learning (3) | Profess ional Skills (5) | Innov ative Appr oach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage I | 4 | 4 | 4 | 3 | 5 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 3 | 4 | 42 |

Comments: Read the base papers. Dynamic LSTM. Atleast 6 parameters comparison to be done.

Sanjay M.
Name & Signature Reviewer1

| | Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environ ment Friendly (2) | Ethics (2) | Team work (2) | Presentati on Skills (3) | Applied Engg & Mgmt principles (3) | Life - long learning (3) | Profess ional Skills (5) | Innov ative Appr oach (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Review of Project Stage I | 4 | 4 | 4 | 3 | 5 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 3 | 4 | 42 |

Comments: Correlation equ. should be implemented.

Dr. Nupur Giri
Name & Signature Reviewer2

Date: 13th September, 2023



Inhouse/ Industry
Innovation/Resear

**Project Evaluation Sheet 2023 - 24**

Class: D12 ~~A/B~~/C
Group No.: 1

Title of Project: Risk Assesment and Behavioral Analysis of Two Wheelers

Group Members: Simran Ahuja (2), Jesica Bijju (10), Sejal Dati (14), Sania Khan (36)

| Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environ ment Friendly (2) | Ethics (2) | Team work (2) | Presentati on Skills (2) | Applied Engg&M gmt principles (3) | Life - long learning (3) | Profess ional Skills (3) | Innov ative Appr oach (3) | Resear ch Paper (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 5 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 4 | 45 |

Comments: work carried out is satisfactory.

Sanjay M.
Name & Signature Reviewer1

| Engineering Concepts & Knowledge (5) | Interpretation of Problem & Analysis (5) | Design / Prototype (5) | Interpretation of Data & Dataset (3) | Modern Tool Usage (5) | Societal Benefit, Safety Consideration (2) | Environ ment Friendly (2) | Ethics (2) | Team work (2) | Presentati on Skills (2) | Applied Engg&M gmt principles (3) | Life - long learning (3) | Profess ional Skills (3) | Innov ative Appr oach (3) | Resear ch Paper (5) | Total Marks (50) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 3 | 2 | 5 | 2 | 2 | 2 | | 2 | 3 | 3 | 3 | 3 | 4 | 45 |

Comments: Publish Research papers in scopus indexed journals

Dr. Nupur Giri
Name & Signature Reviewer 2

Date: 9th March, 2024