Driving Behavior Classification Using Deep Learning in case of National Transport

Franko Efrem Olana

A Thesis Submitted to The Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September 2024

Adama, Ethiopia

Driving Behavior Classification Using Deep Learning in case of National Transport

Franko Efrem Olana

Advisor: Getinet Yilma, PhD

A Thesis Submitted to
The Department of Computer Science and Engineering
School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September 2024
Adama, Ethiopia

DECLARATION

I declare that this thesis/dissertation entitled "Driving Behavior Classification using Deep Learning in Case of National Transport" is my own work. That is, it has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Franko Efrem Olana	________________	______________
Name of student	Signature	Date

RECOMMENDATION OF ADVISORS/SUPERVISORS

I/we, the advisor(s) of this thesis, hereby certify that I/we have read the revised version of the thesis entitled "Driving Behavior Classification using Deep Learning in Case of National Transport" prepared under my/our guidance by Franko Efrem Olana submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering. Therefore, I/we recommend the submission of revised version of the thesis to the department following the applicable procedures.

Getinet Yilma, PhD	___________________	_______________
Major Advisor/Supervisor	Signature	Date
_______________________	__________________	_______________
Co-advisor/Co-supervisor	Signature	Date

Approval Page of M.Sc. Thesis

I/we, the advisors of the thesis entitled "Driving Behavior Classification using Deep Learning in Case of National Transport" and developed by Franko Efrem Olana, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Getinet Yilma, PhD	______________	______________
Major Advisor	Signature	Date
____________________	______________	______________
Co-advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Franko Efrem Olana have read and evaluated the thesis entitled "Driving Behavior Classification using Deep Learning in Case of National Transport" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Mater's of Science in Computer Science and Engineering.

___________________________	_______________	______________
Chairperson	Signature	Date
___________________________	_______________	______________
Internal Examiner	Signature	Date
___________________________	_______________	______________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

___________________________	_______________	______________
Department Head	Signature	Date
___________________________	_______________	______________
School Dean	Signature	Date
________________________________	_______________	_____________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGEMENT

I must first and foremost give thanks to the all-powerful God and his mother, Saint Mary. Additionally, I would like to thank my adviser, Dr. Getinet Yilma, for all of his help and inspiration during my academic career. I am grateful to him for his insightful remarks and his critical reading of my thesis.

I also thank my friends for their unwavering encouragement and support throughout my academic career.

TABLE OF CONTENTS

Contents

List of Tables

List of Figures and Illustrations

List of Acronyms and Abbreviations

ACC- Accuracy
ADAS- advanced driver assistance systems
CNN - Convolutional Neural Network (neural network architecture)
DNN- Deep Neural Network
FPR- False Positive Rate
GPS- Global Positioning System
GRU - Gated Recurrent Unit (neural network architecture)
IMU- Inertial Measurement Unit
LSTM - Long Short-Term Memory (neural network architecture)
ReLU- Rectified Linear Unit (activation function)
RNN- Recurrent Neural Network
TPR- True Positive Rate (Recall)

Abstract

The classification of driver behavior is critical area of research with significant implication of road safety, autonomous driving and intelligent transportation system. There have been researches that related with driver behavior classification that focus on real time driver drowsiness detection using CNN-LSTM model, drowsiness classification system using CNN with EEG signals and abnormal behavior detection using various machine learning methods like Random Forest,KNN,Naïve Bayes and SVM.which lacks model reliance only with facial features which lead limitation in its robustness, which require manual data validation, making the process time consuming and limitation with small sample size respectively. This thesis explores the application of deep learning techniques to classify driver behavior using dataset from transport company that is extracted from GPS device configured to the trucks includes various violation records over six-month period. The dataset comprises time-stamped records of violation categorized into different severity levels, labeled as less-severe, severe, and fatal. Our approach involves rearranging and pre-processing the data to handle missing values, standardizing features, and encoding categorical variables. We constructed custom CNN, LSTM, and GRU models to capture complex patterns from driver behavior dataset. Besides, we compared the performance of the deep learning models with selected conventional machine learning methods. From the experimental results, it is clear that deep learning method LSTM have shown good performance compared to other methods. Future work could extend this research by incorporating environmental factors such as traffic and weather conditions during violation to further enhance the ability of the models effectiveness on classification capability.

Keywords: *Driver Behavior, Classification, LSTM, Gated Recurrent Unit (GRU), CNN, GPS Data*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the study

The FDRE Ministry of Transport and logistics (MoTL) with other stakeholders and partners has prepared strategy based on the understanding of local context and international best practices. The strategy aims to reduce fatal road traffic injuries due to road traffic crashes by 50% in the next ten years (2021-2030). according to 2017 report by the WHO 1.35 Million people die each year as result of road traffic crashes,20-50 million more people suffer non-fatal injuries with many incurring a disability as a result of their injurie and also 90% of the world's fatalities on the road occur in low and middle income countries, even though the country has less than 54% of world vehicles.

In the modern world, maintaining road safety is mostly dependent on driver behavior. The danger of accidents can be greatly increased by driving while distracted, intoxicated, or hostile. Deep learning-based driver behavior classification has shown promise as a solution to this problem. And also, Road safety is still a major worldwide concern. Driver behavior plays a crucial role driving when inattentive, sleepy, or aggressively increases the likelihood of an accident. Deep learning-based driver behavior classification has become a power full technique to address this problem.

Driver behavior classification involves identifying and categorizing different driving patterns and action based on data from variety of sensor imbedded in vehicle. The sensors can capture wide range of information including vehicle dynamics (e.g., road type, weather), and driver specific action (e.g. Braking, lane change). the challenge lies in processing this multi-dimensional data and accurately classifying it into meaningful categories that reflect the driver behavior.

One of the most important areas of research in intelligent transportation systems is driver behavior, with the goal of improving road user efficiency and safety. In order to prevent injuries, it can be helpful to identify risky riding habits and create early warning systems through the systematic analysis of driver behavior. Recent advancements in deep learning techniques have demonstrated an incredibly high ability to accurately and efficiently identify driving behavior from a variety of data sources, such as GPS devices, cameras, and car sensors. (Zhang et al., 2019).

The neural network classifies fresh statistics entirely on the basis of the patterns it has found, and it learns the underlying styles and capacities within the facts. This method has been effective in identifying exceptional driving behaviors such as fatigued driving, distracted driving, and competitive driving. (Mohammed et al., 2021).

The literature has looked into a number of approaches to deal with the problem of classifying driving behavior. Based on the sensor data used, these approaches can be categorized into two main groups: driver dynamics-based approaches and vehicle dynamics-based approaches. Vehicle dynamics-based methods use many sensor data on the dynamics of the vehicle during traveling to determine a set of driving behaviors or styles. Vehicle dynamics sensor data includes things like vehicle orientation, speed, acceleration, and braking events. Additionally, provided are inertial measures from the gyroscope and accelerometer. On the other hand, driver dynamics-based techniques employ data from active sensors (often visual cameras) in the cabin to analyze driver traits like posture, eye contact, and EEG activity. (G.A.M Mering and H.C Myburgh,2015).

An additional recent study investigates spatiotemporal characteristics and their influence on forecasting the hazardous and hostile driving conduct of taxi drivers. Their study sought to use machine learning and geographical analytic techniques to thoroughly evaluate various traffic infractions in Liuzhou, China. The research area's most common infraction category, according to the results, was over speeding. Hotspot maps for various violation kinds were created using frequency-based nearest neighborhood cluster algorithms in the Arc map Geographic information system (GIS). These maps are essential for effectively prioritizing and carrying out treatment choices.

Ultimately, several machine learning (ML) techniques were used to predict and categorize each sort of violation, including decision trees, stack models, and AdaBoost with a base estimator choice trees. A variety of evaluation criteria, including accuracy, F-1 measure, specificity, and log loss, were used to compare the suggested techniques. The outcomes of the predictions showed how effective and reliable the suggested machine learning (ML) technique was. (Zahid, M., Chen, Y., Khan, 2020).

The work that tackles the issue of Gathering and Analyzing a Self-Driving Dataset presents an intriguing approach to dataset creation. They offer a step-by-step tutorial in their paper that covers everything from setting up the hardware to getting a clean dataset. They go over the design of the data gathering scenario, the hardware and software used, the difficulties that need to be taken into account, the data filtering and validation step. They think that by using a streamlined and effective method to gather a small but valuable dataset, benchmarking autonomous driving solutions could be improved by capturing the unique characteristics of local environments. (A. M., & Bacue, 2019, May).

Recent advancement in deep learning, particularly in the area of CNNs and recurrent neural network(RNN) such as long short-term memory(LSTM) networks, have shown significant promise in handling large-scale high dimensional data for classification tasks.

Cars today have improved networking capabilities and can connect to other devices and networks, such as telephones, databases, other cars (vehicle-to-vehicle communication), emergency call centers, roadways, and more, all thanks to the rapid growth of ICT. Even the oldest car models can have their data sent to external apps using on-board diagnostics (OBD) devices.

the categorization of driver behavior by deep learning has a great deal of potential to improve road safety. Research in this area is still in progress, and advancements in deep learning technology will likely make it possible in the future to create more accurate and effective Driver Behavior conduct category models.

1.2 Motivation

Road transport sector known to be crucial for the overall development of our country, if it is not managed in organized coordinated manner, it can have negative impact as much as its positive services. The number of fatalities dues to traffic accident has been reduced by an average six percent per year from 65/10,000 in 2014/15 to 34.4 /10,000 in 2019/20. However, this is still high compared to planned 27/10000 in 2019/2020 fiscal year.in terms of international performance 6.4 death per 10000 vehicles in 2016 and similar period in our countries performance is 61.4.

Analyzing the trend of traffic accident over past five years, the number of serious injuries has increased by average of 13.7% per year, as a result a number of accident increased from 6484, in 2014/15 to 6,929 in 2019/20. on other hand, number of people killed in traffic accident increased from 3495 in 2014/15 to 4133 in 2019/20 an average increases of 3% per year.

According to a 2014/16-2016/2017 study, the main cause of traffic accident in country are driver's incompetence and misconduct (68.4%), vehicle technical incompetence (14.5%), pedestrian traffic problem (4.3%), climate and road design gap (11.5%), and other related factors (1. 4%).Below is paichart for above analytics. (Transport, 2020).

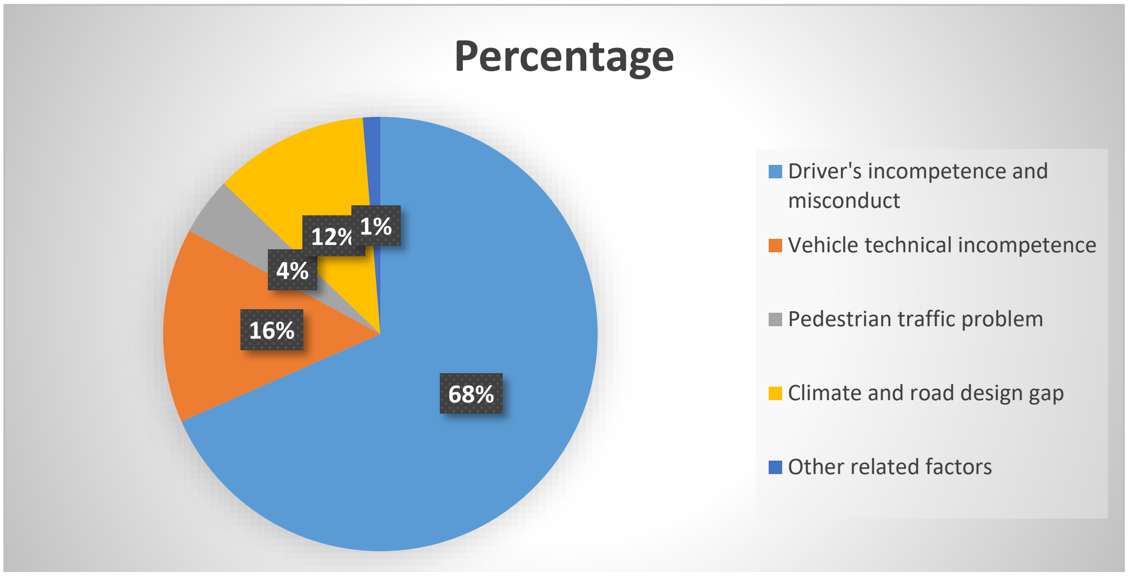


Figure 1 Ethiopian Road Traffic accident main causes

Driving assistance systems' (ADAS) capabilities can be greatly improved by deep learning-based driving behavior classification. Through precise identification of diverse driving behaviors, such as aggressive driving, fatigue, or distraction, ADAS can offer customized alerts or support to the driver, consequently decreasing the probability of collisions.

Robust perception systems are essential for autonomous cars to drive safely in intricate situations. By effectively reading other drivers' behavior on the road, deep learning models for driver behavior classification can help improve autonomous vehicles' perceptive capabilities. Consequently, this makes the interactions between autonomous and human-driven cars safer and more effective.

The goal of research on deep learning-based driver behavior classification is to advance artificial intelligence and machine learning, address real-world problems, enable autonomous vehicles, enable driver assistance systems, improve road safety, and optimize transportation networks. (Holc et al., 2017).

1.3 Statement of Problem

Most of the researches focus on visual data based such images, and videos for driver behavior classification. However, images have limited information of the actions of the driver, it is difficult to capture complex driving scenarios using image or video e.g. for a day driving video. (Dipu, 2021)

Current driver behaviors and drowsiness detection models, which suffer from various limitations. Existing models such as those that rely solely on facial features for drowsiness detection fails to capture the full range of driver behavior leading to incomplete in accurate statement. Other methods, like those utilizing EEG data, face issues with signal artifacts, which require time consuming manual validation process making them inefficient for real time applications. Additionally, some study uses simulated driving data in instead of real-time dataset, limiting their practical applicability, while other use small sample size that reduce reliability generalizability of their findings. Moreover, some approach depends on self-reported data through surveys and questionnaires which introduce biases and may not accurately reflect actual driver behavior.

Driver behavior analysis is crucial in various application including traffic safety, insurance and autonomous vehicle development. However existing research method often relay on visual data such as image and videos to classify driver behavior. While these methods have provided insight. they present significant challenges for capturing complexity of driving behavior. For instance, videos from single camera may not provide comprehensive understanding driver actions over extended periods, such as during full day of driving. this limitation makes it difficult to identify subtle but critical driver patterns.

Moreover, visual data alone often fails to account contextual factors vehicle dynamics, or real time decision making process that are essential for analyzing driving behavior accurately. factors can be better captured using sensor generated data such as data generated from IOT device (accelerometer, GPS, gyroscope) which provide richer and more detailed picture of driving characteristics. (Ward Ahmed Al-Hussein L. Y., 2022)

Apart from the above gaps driver behavior analysis bases on small data size, and diversity. Now a day to better understand about the driver characteristics it is possible to leverage the potential of IoT devices such as sensors. The use of sensor generated data has been paramount for studying driving characteristics. So in this thesis we aim to use sensor generated driving information for driver behavior analysis.

To improve road safety, the problem announcement is the means by which an appropriate deep learning model for the class of driver behavior may be developed. This model tackles the aforementioned issues and can be used in real-world, international applications. Firstly, there are variations in driver behavior across different contexts and environments, requiring the development of context-specific models. Secondly, the interpretation of deep learning models and explaining the factors contributing to driver behavior remains a challenge, potentially leading to limited adoption by stakeholders such as regulators and insurers. Therefore, the problem statement is to develop better deep learning models for the classification of driver behavior that take into account the contextual factor affecting driving behavior.

1.4 Research Question

1.4.1. Questions and/or hypotheses

Questions:

RQ1. What are the determining factor or attribute that influence driver behavior?
RQ2. Which classification model is best suited to driver behavior classification?
RQ3. To what extend the performance of the model effective in classifying driver behavior?

1.5 Objectives of the Study

1.5.1 General Objective

General objective of this study is classifying driver behavior using deep learning and drive insight that revolutionize driver safety and fleet management.

1.5.2 Specific Objective

The specific objectives of the study are:

- ✓ Examine relevant studies on the analysis of driver behavior
- ✓ Collect and preprocess sensor based driving dataset for driver behavior analysis
- ✓ Develop a deep learning model that for driver behavior classification.
- ✓ Explore the impact of different violation data combinations on model performance.
- ✓ Evaluate the performance of the proposed method

1.6 Significance of the study

The classification of driver behavior using deep learning has great significance in the field of autonomous driving and driver assistance systems. With the increasing number of vehicles being equipped with advanced driver assistance systems (ADAS), it is becoming more important to accurately classify driver behavior to ensure the safety of passengers and pedestrians. Some of the significant implications of this study are:

- ✓ **Improved safety**: By accurately classifying driver behavior, ADAS can provide appropriate responses to potential hazards on the road, such as sudden lane changes or collisions, thereby reducing the risk of accidents.
- ✓ **Enhanced comfort:** By understanding driver behavior, ADAS can adjust vehicle settings such as speed, lighting and air conditioning to create a more comfortable driving experience for passengers.
- ✓ **Increased efficiency:** By analyzing driver behavior in real-time, ADAS can optimize energy consumption and fuel efficiency by adjusting the vehicle's speed and acceleration according to the traffic conditions.
- ✓ **Cost savings:** By reducing the likelihood of accidents and improving fuel efficiency, the cost of vehicle maintenance and repair can be significantly reduced.

Overall, the classification of driver behavior using deep learning has the potential to revolutionize the automotive industry, making driving safer, more comfortable and efficient.

1.7 Scope and Limitation of the Study

1.7.1 Scope of The Study

Deep learning has been used to create driver behavior classification in a variety of ways, and this field of study has seen significant growth recently. Therefore, this study's primary objective is to create an accurate classification system by the application of novel methods and a deep learning methodology. The scope of the thesis work within the given time and resource includes:

- ✓ On this study the model will be trained only on violation reports gained from one company (National Transport PLC).
- ✓ It works only to classify 3 driver behaviors only
- ✓ It works with nominal dataset only
- ✓ The dataset does not include specific period and distance of events that may include in future work
- ✓ Environmental factors has not considered that may include in future work.

1.7.2 Limitation of The Study

The main objective of the study was to investigate how well deep learning approaches can identify trends in driver behavior and classify driver behavior in order to help transport companies to be more productive and effective in driver performance evaluation. This research concerned with answering the question indicated above.

Hence, we have nominal dataset which was obtained from violation reports of trucks, we haven't considered other behavior detection and deterministic activities of drivers like facial recognition .and also our research mainly focuses on truck drivers because, our dataset is obtained from transport company trucks violations report that may not effectively applied automobile's and other non-truck cars,

Furthermore, the classification model was build based on these models, CNN, LSTM, GRU.The classification result is evaluated using Accuracy, Precision, Recall and F1-score. Also The impact of outside variables like weather, traffic, or road conditions and its impact on driver behavior will not be covered by the study.

A more precise and trustful identification of particular driving behaviors based on violation report data from trucks is the anticipated result of driver behavior analysis employing Deep Learning

Techniques. This may result in better driver performance evaluations and enhanced car safety systems, which together may help to lower the overall number of traffic accidents. The use of Deep Learning Techniques in driver behavior analysis can also reveal information about how driver's violation record has an impact on severity of trucks damage. This knowledge canbe applied to enhance driver training programs, traffic management techniques, and roadway design. Overall, the study of driver behavior using Deep Learning Techniques has the potential to dramatically increase driving efficiency and road safety.

1.8 Organization of the Thesis

This research work was divided into seven chapters, which are briefly discussed below.

Chapter One**:** This chapter provides an outline of the research work as well as justifications for how and what will be accomplished.

Chapter Two**:** This chapter explains the background literature and related works regarding, Driver behavior classification, deep learning, time series classification, and RNN.

Chapter Three**:** This chapter covers research methodologies such as data collection, data preprocessing, design tools, development frameworks and platforms, and evaluation methods.

Chapter Four**:** Using different block diagrams, pseudo-codes, and flow charts of the implementation with corresponding mathematical descriptions, this chapter aims to provide comprehensive information on how the proposed solution attempts to address the problem.

Chapter Five: This chapter explains how to implement the proposed solution to the problem into action. It also contains a code snippet for the proposed solutions as well as the environment setup.

Chapter Six**:** The outcomes of the proposed solution are discussed in this chapter, along with a comparison of the results utilizing figures, graphs, and tables.

Chapter Seven: The research is summarized in this chapter. The conclusion of the work is given, as well as suggested future works.

CHAPTER TWO

2. LITERATURE REVIEW

2.1 Introduction

The classification of driver behavior is useful with the rapid development of deep learning techniques; various studies have proposed deep learning-based approaches to classify driver behavior

2.2 Time Series Classification

A time series (seconds, minutes, hours, days, months) is a data presentation that is organized chronologically. In order to create predictions, time series analysis also makes it possible to describe and explain a phenomenon over time. An observational set that has been measured progressively throughout time is called a time-series. Either a continuous record of these observations is kept over time, or they are kept at distinct, equal intervals. Machine learning methods like ANN and SVM are used to forecast time series. In the field of time series classification, research shows that the LSTM model performs better than traditional techniques.

Time series analysis is an attractive field of study with a wide range of applications in computer science, business, economics, finance, and the classification of driver behavior. A time sequence study aims to forecast the time series' future values, examine the journey observations, and create a model that explains the data's structure. Deep learning is a contemporary method for time series forecasting.

2.3. Deep Learning Method for Driver

2.3.1 CNN

type of deep learning model that has been used for Driver behavior classification is the convolutional neural network (CNN), which has traditionally been used for image analysis but can also be applied to driver behavior classification text data. Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that have been widely used for image recognition and computer vision tasks. CNNs are designed to automatically learn features from raw input data by employing convolutional filters, which allow the network to extract spatial and temporal patterns from the input signal. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers are responsible for extracting local features from the input signal, while pooling layers reduce the spatial dimensions of the features. Fully connected layers are used to classify the input based on the extracted features.

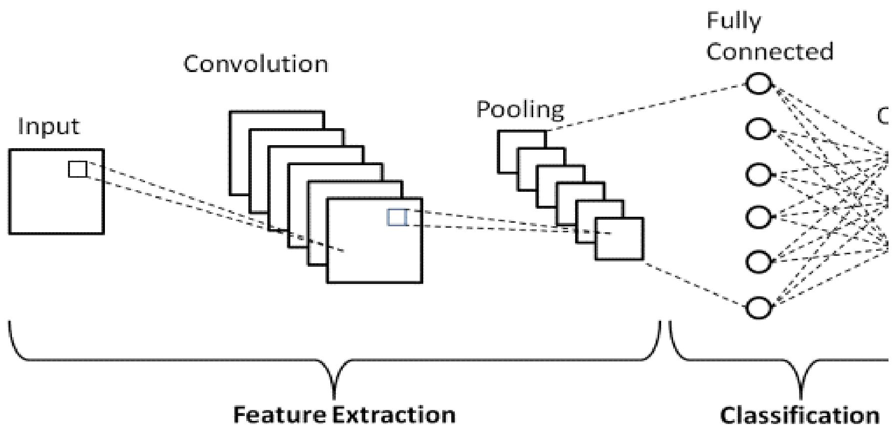


Figure 1 CNN layers adopted

2.3.2Recurrent Neural Network (RNN)

Another type of deep learning model for Driver Behavior classification is the recurrent neural network (RNN). Deep learning architectures known as recurrent neural networks (RNNs) are made to handle sequential data. RNNs can capture temporal dependencies in the data, making them particularly well-suited for processing temporal data, such as time series. The foundation of RNNs is the notion of keeping a “memory” of previous inputs as they process a stream of data. Each time step updates the memory, which is kept in a hidden state. The network can detect temporal

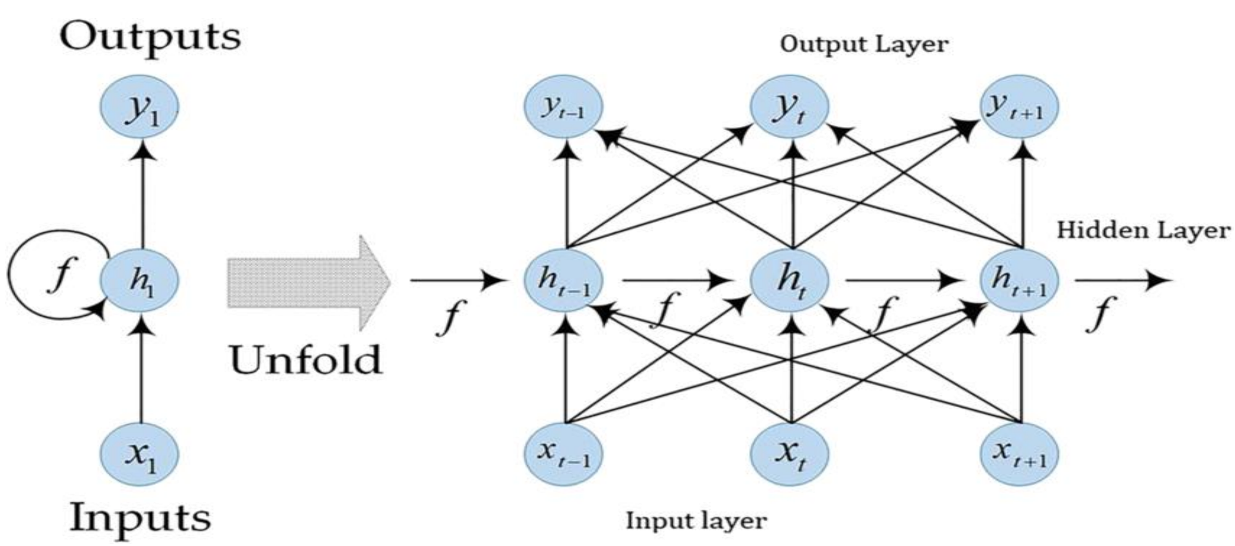


Figure 2 Recurrent Neural Network adopted

dependencies in the data because the hidden state at each time step is a function of the current input and the previous hidden state.

2.3.2.1 LSTM Model

LSTM is a special kind of recurrent neural network, which was proposed by Hoch Reiter S and Schmid Huber J. LSTM adds memory storage (cell) and Gates structures to the traditional recurrent neural network. the LSTM model demonstrated the best performance with a MAPE value of 1.36%. The cell (Ct) is used to record the state of neurons, and the gates pass through information selectively, by means of a sigmoid neural layer and a point-by-point multiplication mechanism. With three gate structures similar to filters, LSTM realizes information protection and control: forgetting gate (ft) determines the information discarded from the cell state, input gate (it) determines new information added to the cell, and output gate (ot) filters the final output information. LSTM can express the long-term dependent information in the input features through the gate structure, alleviating the problems of gradient disappearance and gradient explosion effectively.

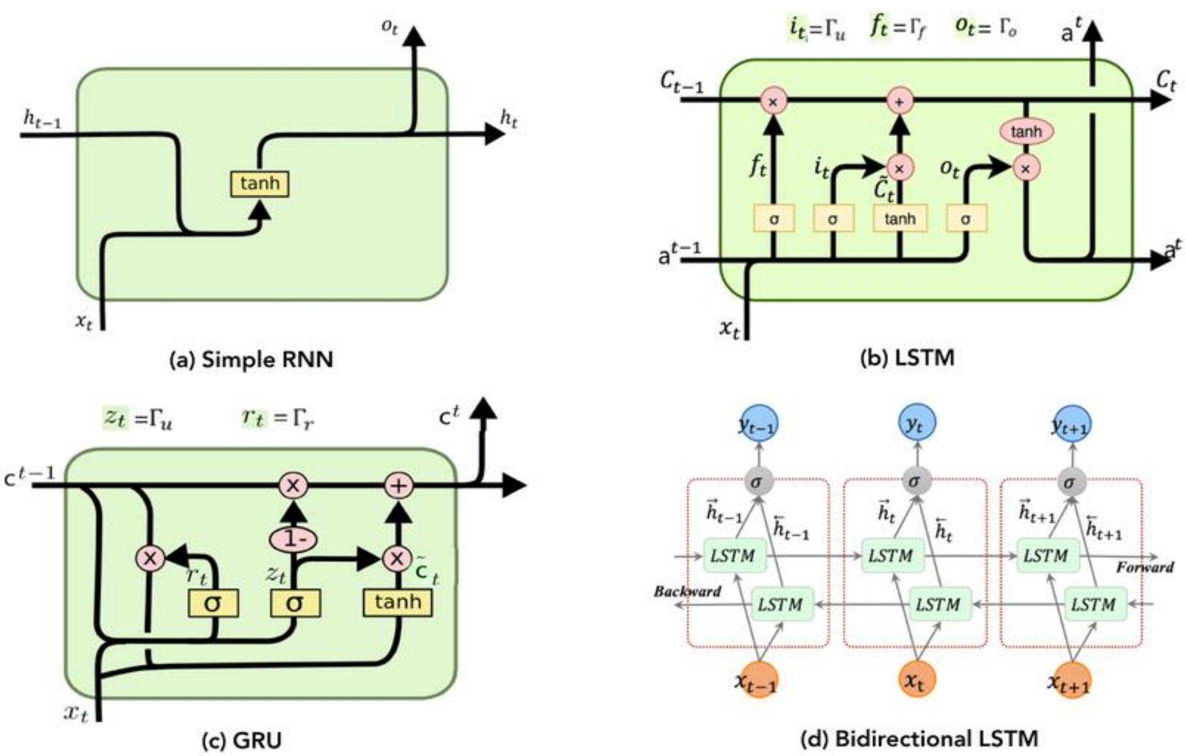


Figure 3 LSTM model Architecture adopted

In LSTM module structure showed in Figure 3, x is the input vector of LSTM model; h is the output vector of the model; f, i and o represent activation values of forgetting gate, input gate and output gate respectively. C and C' represent the cell state and its candidate values, Subscript t represents time, σ, tach represent ReLu and tach activation functions respectively and b represent weight and deviation matrix respectively. In the first step, the LSTM layer determines information that will be discarded from the previous cell state Ct−1. Therefore, the calculation of the activation value ft of the forgetting gate at time t is based on the input xt at time t, the output at time t −1 and the deviation matrix bf of the forgetting gate. The ReLu function normalizes all activation values between 0 (all forgotten) and 1 (all remembered):

$\mathrm{ft} = \sigma\,(\mathrm{wf}\,[\mathrm{ht} - 1, \mathrm{xt}\,] + \mathrm{bf})$ Equation 2.1

- F_t: forget get output it determine what information to discarded from previous cell
- σ: sigmoid activation function, squashing the value between 0 and 1.
- Wf: weight matrix of the forget get.
- $\mathrm{ht} - 1$: hidden state from previous time step
- xt: input at the current time step

In the second step, the LSTM layer decides information that will be added to the cell state Ct. This step consists of two parts: first, calculate C't, that it may be added to the cell state, Second, calculate the activation value of the input gate:

$\mathrm{C't} = \mathrm{tach}\,(\mathrm{wC}[\mathrm{ht} - 1, \mathrm{xt}\,] + \mathrm{bc})$ Equation 2.2

- $\mathrm{C't}$: candidate cell state, which is potential update for cell state
- tach : hyper parabolic hyper function
- wC: weight matrix of generating the candidate cell state
- bc: bias for candidate cell

third step is update cell state

$I_t = \sigma(\mathrm{Wi}[\mathrm{ht} - 1, \mathrm{xt}] + \mathrm{bi})$ Equation 2.3

- I_t: input get output, controlling how much the candidate state C't, should be added to cell state.
- Wi : weighted matrix for the input gate
- bi: bias for input gate

The four calculate output gate activation(O_t) and hidden state(H_t).

$\mathrm{Ot} = \sigma(\mathrm{Wo}[\mathrm{ht} - 1, \mathrm{xt}] + \mathrm{bo})$ Equation 2.4

$ht = Ot * \tanh(Ct)$ Equation 2.5

- Ot : output gate output, determining the part of the cell state to output
- Wo: weighted matrix of output gate
- bo: bias of output gate

2.3.2.2 GRU (Gated Recurrent Unit)

Recurrent networks with gated recurrent units (GRU) units are long-term 311 memory networks like LTSMs but emerged in 2014 as a simplification 312 of LTSMs due to the high computational cost of the LSTM networks. GRU is one of the most commonly used versions that researchers have converged on and found to be robust and useful for many different problems. The use of gates in RNNs has made it possible to improve capturing of very long range dependencies, making RNNs much more effective. The LSTM is more powerful and more effective since it has three gates instead of two, but GRU is a simpler model and it is computationally faster as it only has two gates, Γu update gate and Γr Relevance gate. The Γ u gate will decide whether the ct memory state is or is not updated using the ect memory state candidate. The Γr gate determines how relevant is ct−1 to compute the next candidate for ct , that is, ect. A GRU unit is defined by the following equations:

$\Gamma r = \sigma(Wr \cdot [xt, ct - 1] + br)$ **Equation 2.3.2.2.1**

$r\Gamma u = \sigma(Wu \cdot [xt, ct - 1] + bu)$ **Equation (2.3.2.2.2)**

$C\tilde{}t = \tanh(WC \cdot [xt, \Gamma r * ct - 1] + bC)$ **Equation (2.3.2.2.3)**

where Wc and bC are weight matrix and bias for candidate and Wu and Wr, and bu and br are the weights and the bias that govern the behavior of the Γu and Γr gates, respectively, and Wc and bc are the weights and bias of the ect memory cell candidate.

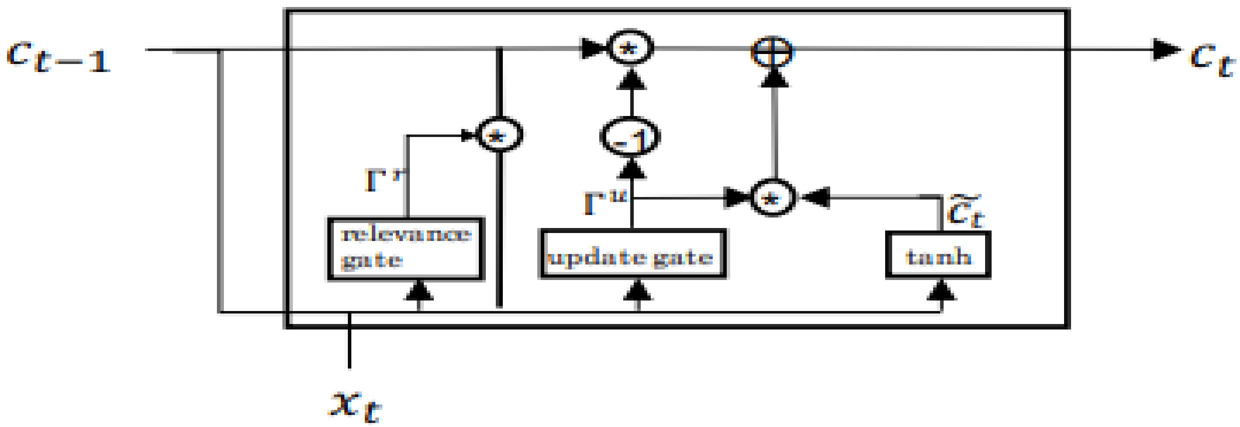


Figure 4 GRU model Architecture adopted

2.4 Related work

In real-time driver drowsiness detection using deep learning the researcher's aims to detect drowsiness by applying object detection tasks using techniques of MobileNets, a light weight convolutional neural network architecture along with the single short multibox detector (SSD) system that is on top of the mobile Net architecture. they used annotated images that includes human faces eyes with open and closed with 1234 images which is nearly 10MB with annotations. They test dataset with several light conditions, under broad range of conditions, subjections and obstruction. For dataset training they used tensor flow API with an average step speed of 0.5 s, to train their model. then, the researcher used MobileNet_SSD_V1 model. And they finally suggest that SSD_MOBILENET architecture for better suit for drowsiness detection. However, the model relies solely on facial features for drowsiness detection it was not able to detect time series behaviors that perfectly defines the driver. (Md. Tanvir Ahammed Dipu1, 2021)

Other detection method that was introduced was automated classification system for drowsiness detection using convolutional neural network and electroesphalogram. to detect the drowsiness using single-channel EEG signals with best possible accuracy without considering hand-engineered features and signal processing techniques, the proposed work performs classification directly from raw EEG signals using CNN. the EEG dataset used and EEG sleep data publicly available in national institute of health(NIH) research resource named as physionet.the sleep EEG was recorded by the Fpz-Cz and PZ-OZ electrodes. The data has been recorded for different durations for various subjects /that divided into five groups awake, on-rapid eye movement(NREM) AND Rapid Eye Movement (REM), move state and not scored state.as the drowsy stage the stage when a person transits from awake stage to sleep stage for a very short span of time ,so they consider the initial stage of sleep data, that S1 stage as the drowsy state for four work.thus,we model the data as series of alternate awake and sleep stage out of which they consider a window of single stage.however,arthifacts interfere with EEG and reduces the information of signals so to avoid this they used manually validating each epoch data which is time taking and tedious. The same types of dataset used by other researcher's. (Phanikrishna, 2020) .

Techniques of driver behavior recognition method using temporal convolutional neural for vision based driving behavior recognition has proposed using spatial stream convolutional neural network and temporal stream convolutional neural network by creating video-based dataset of driving behavior containing 1237 video. Of with average 15 seconds per video and involving 6 simulated diving behavior. They utilize spatial stream CNN to learn appearance information's such as texture, contour, interest point etc. from still images and it has identical square of VGGNet, which take 224X224X3 still frame as import and outputs the probability distribution of different driving behavior categories. The spatial stream CNN is staked by thirteen convolutional layers and three inner produced layers. They used Thirteen convolutional layers' filter images with kernel size of 3X3, stride of 1 and stacked into 5 blocks. max pooling follows after each convolutional block with kernel size of 2x2, stride of 2. fully connected layers can map the convolutional feature map to feature vector. They perform experiment on self-created simulated driving behavior dataset and proposed architecture with mid-level fusion strategies and achieves state of the art recognition performance. But, still real driving dataset is essential for practical application. Zhang et al(Yaocong Hu, 2019).

Driver behavior classification using label data collected from volunteers and features selection that aimed to predefine input values for predict activities has introduced by signal data obtained from smart phone. on the volunteer's pocket, handbag, or in hands. they suggest approaches to preparing data for classifier by collected and transform data base one windowing technical and feature extraction. they used accelerometer, gyroscope and magnetometer sensors to transform accelerometer data from smart phone coordinate system. They used two approaches to classify data as offline on line training. They tried to build up dataset from series activities while they are driving on sleep, drunk and frequency swing with high velocity. They used K means clustering algorithm to set of K linear activity into three class and the label of abnormal. The ANN algorithm is applied by some research in human activity behavior recognitions system. the neurons have received input from other neurons. The value of each inputs determined by weight associated with them. The sum of input weights computed and value output is according to transfer function. For classification they used Random Forest, KNN, naïve Bayes, SVM.they also used 10-fold cross validation for evaluating the accuracy of each classification algorithms. (Dang-Nhac Lu, 2018).

On Driver Behavior Profiling and Recognition Using Deep-Learning Methods: In Accordance with Traffic Regulations and Experts Guidelines the researchers tried to participate 30 individuals and the sample was diverse enough in its gender and age representation they have average of 22.28 years of driving experience with data acquisition system consist of various sensors including onboard diagnostics(OBDII)reader,2ultrasoundsensors, inertia measurement unit, sensor(GPS) and preprocessed and used three models DNN,RNN and CNN by comparing performance of deep learning across various time frame segments ranging from 1 to 10 s.eventhough CNN is primarily used nowadays image classification, it produced the most accurate result when compared to DNN and RNN.hyperparameters such as epochs,batchsize,learning rate, and optimizers were tested on segments of the classification of 1 to 10 s in length .a confusion matrix was deployed to test the performance of the classification algorithms. (Ward Ahmed Al-Hussein, 2022).

2.5 Summary on related work

Table 1. Related Work

No.	Authors (Year)	Deep Learning Model used	Accuracy>93%	Dataset size more than 7000	No Time-consuming validation	comparison with other deep learning methods
1	Md. Tanvir Ahammed Dipu (2021)	✓	—	—	✓	✓
2	Venkata Udaya Sameer (2021)	✓	✓	—	—	✓
3	Zhang et al. (Yaocong Hu, 2019)	✓	—	—	✓	—
4	Dang-Nhac Lu (2018)	—	—	✓	—	✓
5	Abdul (Ward Ahmed Al-Hussein, 2022)	✓	—	✓	—	✓
6	Ours	✓	✓	✓	✓	✓

CHAPTER THREE

3. MATERIAL AND METHOD

Research methodology is the technique of methodically solving research problems. It helps the researcher realize which strategies or approaches are being used in the investigation The classification process for driver behavior is explained in detail in this section. This part outlines the methods for collecting data, from the preliminary stages of data collection to data preprocessing, and provides a thorough explanation of the assessment metrics applied to the models.

3.1 Proposed Overall Architecture

This section describes the proposed overall architecture of driver behavior classification. The proposed architecture contains basic development processes; such as data acquisition, preprocessing, and model training with recurrent neural networks like custom CNN, GRU and LSTM Finally, the model's performance is evaluated using Accuracy, Recall, Precision and F1-score. The reason choosing these deep learning algorithms is because of the nature of the dataset, i.e., the dataset is in a time-series format. This data set contain information of 7178 drivers event, files with csv format with varying lengths that record the car’s position in addition deep learning algorithms had become the current state of art and has got better performance effectiveness for time-series forecasting.

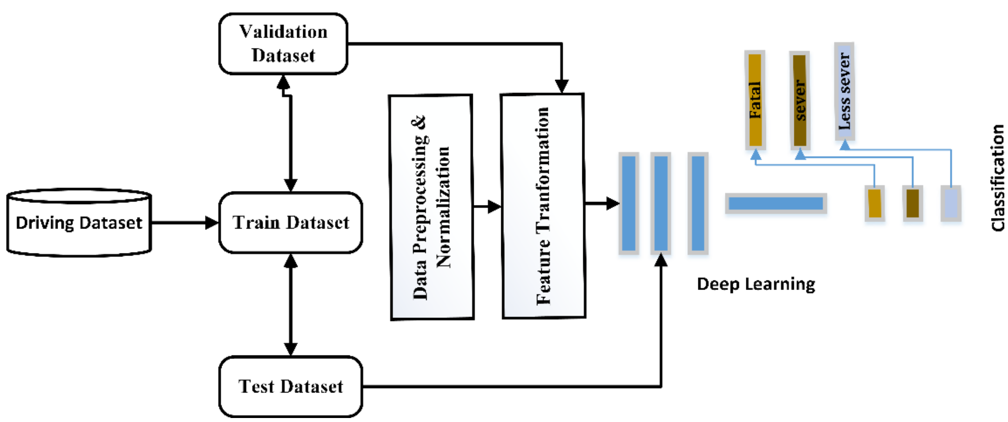


Figure 5 proposed overall structure

Table 2 Dataset Features or attribiutes describtion

NO	Attribute Name	Datatype	Description
1	Truck	Float32	Heavy duty trucks
2	Acceleration extreme	Float32	increase in speed, significantly higher than normal driving conditions
3	Acceleration medium	Float32	moderate increase in speed that is above normal acceleration but not as intense as extreme acceleration
4	Brake extreme	Float32	Indicates instances where a vehicle undergoes a rapid deceleration, much higher than typical braking events
5	Brake medium	Float32	moderate braking that is stronger than usual but not as severe as extreme braking
6	Excessive Idle	Float32	situations where the vehicle engine remains running while the vehicle is stationary for extended periods
7	Harsh Acceleration	Float32	aggressive and rapid acceleration, which can lead to increased fuel consumption
	Harsh Braking	Float32	sudden and forceful braking that can cause abrupt stops, posing risks to the driver
	Harsh Cornering	Float32	sharp and aggressive turning at high speeds, which can lead to loss of vehicle control and increased risk of accidents
	Harsh driving	Float32	A general term encompassing aggressive driving behaviors such as harsh acceleration, harsh braking
	Speeding medium	Float32	driving at speeds moderately above the posted speed limit or safe driving conditions
	Speeding mild	Float32	driving slightly above the speed limit. While not as dangerous as medium or extreme speeding
	Turn extreme	Float32	very sharp and rapid turning maneuvers that exceed typical driving conditions
	Turn medium	Float32	moderate turning that is more pronounced than usual but not as intense as extreme turns
	Speeding extreme	Float32	Indicates driving at speeds significantly above the posted speed limit
	Harsh Braking	Float32	occurs when a driver decelerates at a rapid rate, often resulting in a sudden stop.
	Speeding extreme	Float32	extreme refers to driving at speeds significantly higher than the posted speed limits or what is considered safe for the current road conditions.
	Time Stamp	Float32	date of each event takes place
	Label	Float32	Classes of drivers Fatal, Severe and Less Severe

3.2 Data Source and Collection methods

Dataset is crucial for any deep learning research specially for time series data analysis like driver behavior classification using deep learning. we have used driver violation report data's extracted from GPS that is collected from one transport company (National Transport PLC). this data has number of records that shows drivers violation and period of the violations. For our research purpose we used six-month violation data of 150 trucks with average 80 trucks movement activities per day. We have extracted each month violation report and from this data we tried to classify the drivers into three classes fatal, Sevier, and Less-Sevier by applying the following steps.

First we have extracted each month violation data from GPS system that contained each driver's violation along their activity during their daily operation. Like below table

Table 3 Row dataset collected from GPS System

Truck	Time	Violation
[redacted]067	8/5/2023 6:45	Acceleration: medium
[redacted]067	8/5/2023 14:49	Turn: medium
[redacted]710	8/11/2023 10:38	Harsh Braking
[redacted]708	8/12/2023 16:09	Excessive Idle
[redacted]708	8/13/2023 18:04	Harsh driving
[redacted]710	8/15/2023 15:45	Harsh Braking

Second we have combined the violation data extracted and we restructured the dataset using hash label encoding to have truck violation data with each count of violation occurrence from each driver to form below table of dataset.

Table 4 Restructered dataset

Driver	Date	Acceleration extreme	Acceleration medium	Brake extreme	Brake medium	Excessive Idle.	Harsh Acceleration	Harsh Braking	Harsh Cornering	Harsh driving	Speeding extreme	Speeding medium	Speeding mild	Turn extreme	Turn medium
711	2023-08-01	0	0	0	0	1	0	0	0	0	0	0	0	0	1
084	2023-08-01	0	0	0	2	0	0	0	0	0	0	0	0	0	0
560	2023-08-01	0	0	0	0	1	1	0	1	0	0	0	0	0	0
654	2023-08-02	0	0	0	0	1	0	0	0	0	0	0	0	0	0
655	2023-08-02	0	1	0	0	1	1	5	5	1	0	0	0	1	0
656	2023-08-03	0	0	0	0	0	1	12	4	0	0	0	0	0	0

Thirdly we have given weight to each violation type based on company's violation manual on evaluation metrics stated by GPS Experts and best practice of other international some company's (Oil Libya, Coca-Cola, and Dangote Cement) manuals and then we have placed values to each violation occurrence by multiplying each occurrence of violation by its given weight as below.

Table 5 Violation type with weight

Violation Type	Value
Acceleration extreme	10
Acceleration medium	5
Brake extreme	10
Brake medium	5
Excessive idle	2
Harsh Acceleration	8
Harsh Braking	8
Harsh Cornering	8
Harsh driving	10
Speeding medium	6
Speeding mild	3
Turn extreme	7
Turn medium	4
Reckless driving	10

Table 6 restructured data with weight

Driver	**Date**	**Acceleration**	**Acceleration**	**Brake extreme**	**Brake medium**	**Excessive Idle.**	**Harsh**	**Harsh Braking**	**Harsh Cornering**	**Harsh driving**	**Speeding extreme**	**Speeding medium**	**Speeding mild**	**Turn extreme**	**Turn medium**
711	8/1/2023	0	0	0	0	2	0	0	0	0	0	0	0	0	4
84	8/1/2023	0	0	0	10	0	0	0	0	0	0	0	0	0	0
560	8/1/2023	0	0	0	0	2	10	0	10	0	0	0	0	0	0
654	8/2/2023	0	0	0	0	2	0	0	0	0	0	0	0	0	0
655	8/2/2023	0	5	0	0	2	10	50	50	10	0	0	0	7	0
656	8/3/2023	0	0	0	0	0	10	120	40	0	0	0	0	0	0

Fourthly we have calculated "driver score "by calculating waiting sum of each rows violation based on predefined weights like below table and normalized score between 0 and 1 using min max scaler to overcome outliers.

Table 7 normalized dataset

Date	Driver	Acceleration	Acceleration medium	Brake extreme	Brake medium	Excessive Idle.	Harsh Acceleration	Harsh Braking	Harsh Cornering	Harsh driving	Speeding	Speeding medium	Speeding mild	Turn extreme	Turn medium	score
8/1/2023	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.02994
8/1/2023	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0.04191
8/1/2023	3	0	0	0	0	1	1	0	0.2	0	0	0	0	0	0	0.11976
8/2/2023	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
8/2/2023	5	0	1	0	0	1	1	0.416	1	1	0	0	0	1	0	0.80838
8/3/2023	6	0	0	0	0	0	1	1	0.8	0	0	0	0	0	0	1

Lastly, we have tried to classify driver based on normalized score by setting thresholds using quantiles. We have sated low threshold for quantiles less than 0.33, high threshold for quantiles greater than 0.66, and medium threshold for quantiles between 0.33 and 0.66. we have settled this threshold after we have tried to classify our data set based on average, other quantiles (0.25,0.75), by seeing how that effect the classification and by analyzing data distribution or properties of our dataset. based on this we have classified drivers into three behavioral categories of high threshold, medium threshold, and low threshold for Fatal, severe, and less-severe respectively. Then, we use augmentation technique and added noise to dataset to reduce overfitting during training Like below table.

Table 8 classified dataset

Driver	Time	acceleration extreme	Acceleration medium	Brake extreme	Brake medium	Excessive Idle	Harsh Acceleration	Harsh Braking	Harsh Cornering	Harsh driving	Speeding medium	Speeding mild	Turn extreme	Turn medium	Driver Score	label
1	8/1/2023	0.00044	0.015	0.07914	0.007695	0.0316	0.0344	0.026015	0.021271	0.03811	1.0025	0.0043	0.071664	0.076396	5.985375	Less-severe
2	8/1/2023	0.029649	0.02419	0.034376	0.017191	1.010074	0.034339	0.00423	0.05878	0.049266	0.03659	0.048407	0.00101	0.902214	5.951554	Less-severe
3	8/1/2023	0.026975	2.021	0.01176	0.104146	0.0036	0.023421	0.01053	0.029069	0.014556	1.08365	1.0867	0.978359	3.110469	37.93774	severe
4	8/1/2023	0.000358	0.015	0.02211	2.080942	0.0082	0.040677	0.032801	0.154888	0.00524	0.058323	0.02238	0.07566	0.031598	9.993917	Less-severe
5	8/1/2023	0.057613	0.07936	0.06766	0.086881	0.99952	0.03976	0.004799	0.04155	0.0096	0.06451	0.01554	0.01229	0.059581	2.005359	Less-severe
6	8/1/2023	0.02188	0.0042	0.03918	0.02394	0.035	0.939936	12.0216	3.933148	0.03181	0.05022	0.03012	0.004311	136.0232	0.060594	Fatal

3.3 Dataset

The proposed dataset was collected from trucks that have GPS device which model is called Teltonica TFT100 installed on Trucks board for controlling purpose of trucks during operation.

Figure 6 source for proposed dataset

The total dataset contains 7178 records with 21 columns. From the dataset **training file** is composed of **5743** recordings; **2539** are classified as **Fatal**, **1186** as **Sevier**, and **2018** as **Less-Severe**.

The **test dataset** is composed of a slightly lower number of instances: **1435** divided into **688** for **Less-Sevier**, **365** for **Sevier** and **382** for **Fatal**, which reflects the proportions from the training dataset but with lower values.

We have data set containing tabular dataset containing Acceleration Extreme Acceleration, medium, Brake extreme, Brake medium, Excessively, Harsh Acceleration, Harsh Braking, Harsh cornering, Harsh driving, speeding medium, Speeding, mild, turn, extreme, Turn, medium, Reckless, driving along with timestamps and class labels.

In our case, we are assuming that a driving behavior can be in one of the following classes:

- ✓ Less-Sevier – a minor violation with less significant consequences. unsafe, are less dangerous than severe or fatal incidents.
- ✓ Sevier – high risky and can Couse considerable harm, requires substantial vehicle repair.
- ✓ Fatal – These behaviors are considered the most dangerous and critical, often involving significant

violations of traffic laws or extreme recklessness.

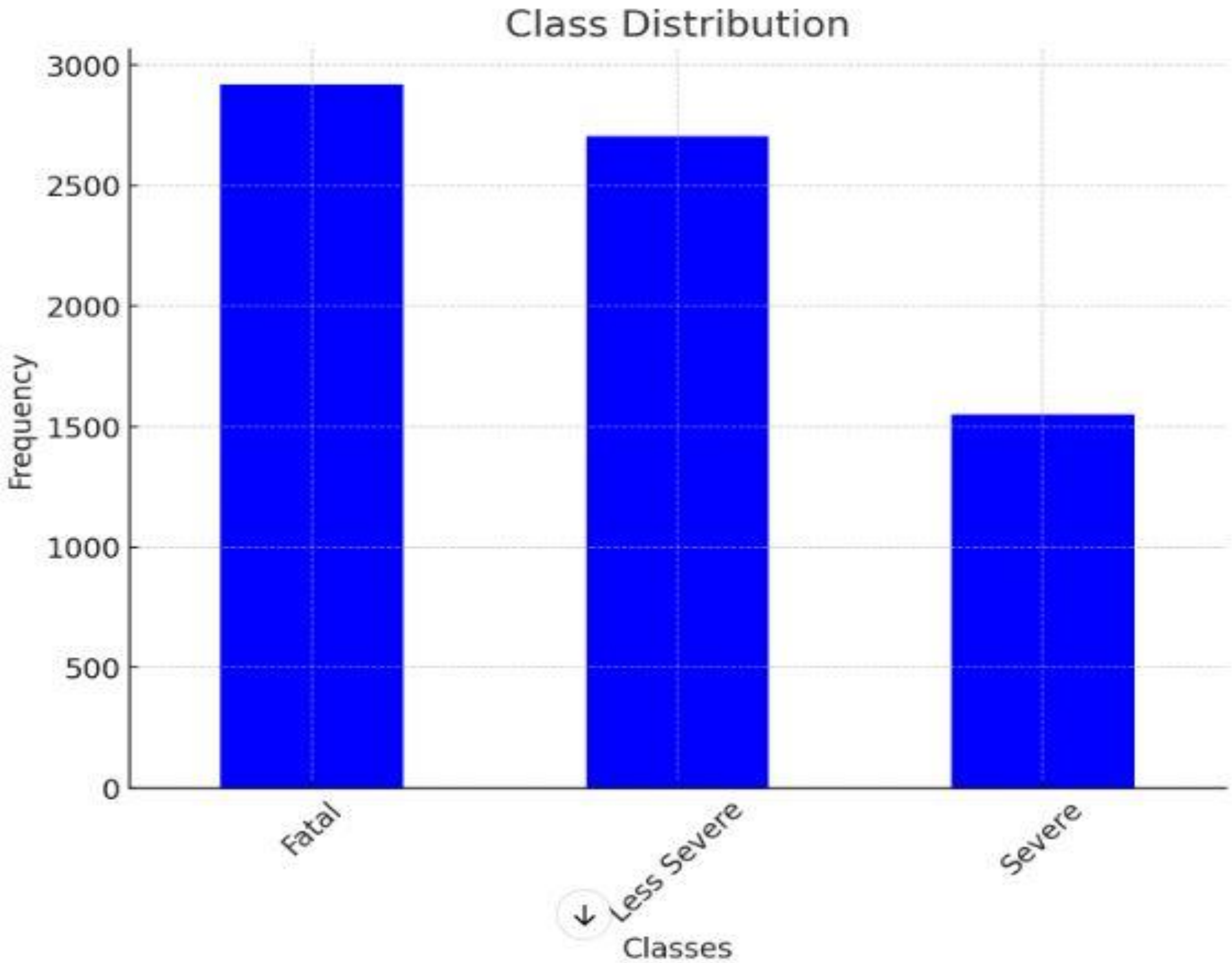


Figure 7 class distribution

3.3.1 Explanatory Dataset Analysis

The data analysis provides an initial understanding of the dataset. For data analysis purposes, we will present how data is distributed and how the GPS data differ between the driving behaviors collected. For example, a higher correlation between speeding and harsh braking could indicate fatal driving behavior. Here are some insights.

Table 9 dataset description

statistic	Acceleration	Brake	Excessive_Idl	harsh Cornering	Harsh driving	Speeding medium	Speeding mild	turn	reckless driving
Count	7178	7178	7178	7178	7178	7178	7178	7178	7178
Mean	13.86	15.192	0.179	20.615	0.794	0.374	0.428	0.208	1.234
Std Dev	74.626	79	0.743	80.641	4.162	0.417	2.171	0.993	28.252
Min	0	0	0	0	0	0	0	0	0
25th Percentile	1	1	0	0	0	0	0	0	0
Median	4	4	0	0	0	0	0	0	0
75th Percentile	10	10	0	10	0	0	0	0	0
Max	2580	2300	15	2130	933	938	65	28	2185

Data format: The real data set requires a modification in data format before it can be used directly by the machine learning model. Data formatting is the process of transforming the format of the data into one that a machine learning algorithm can use or understand. The data is saved as a Comma-Separated Value (CSV)in MS-Excel after being collected. This CSV data is transformed into spreadsheet cells and columns by the spreadsheet program for simple viewing and editing.

Normalization: The data fall between the ranges of 0 and 1 or any other range of values is required. Deep learning is sensitive to the amount of input data, particularly when utilizing the Sigmoid (default) or ReLU activation functions. The dataset can be easily normalized using the MinMaxScaler preprocessing class from the sci-kit-learn toolkit. The completion of data preprocessing in machine learning is signaled by feature scaling. It is a method for ensuring that the independent variables in a dataset remain inside a specific range. In this instance, it reduces the number of possible variables, so that you can compare them fairly. To scale data within a range, min-max is based on the minimum and maximum values of characteristics. The

MinMaxScaler is used by the researcher to normalize or scale the data for this thesis.

$$x'_i = \frac{X_i - min(X)}{max(X) - min(X)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{Equation (3. 2)}$$

where x'_i is the rescaled element, x is the feature vector, and xi is an element of feature x.

Feature extraction with Fourier analysis: By applying Fourier analysis to these driver behavior dataset, we decompose them into their frequency components. Specific frequencies in the signal might correspond to certain driver behaviors.

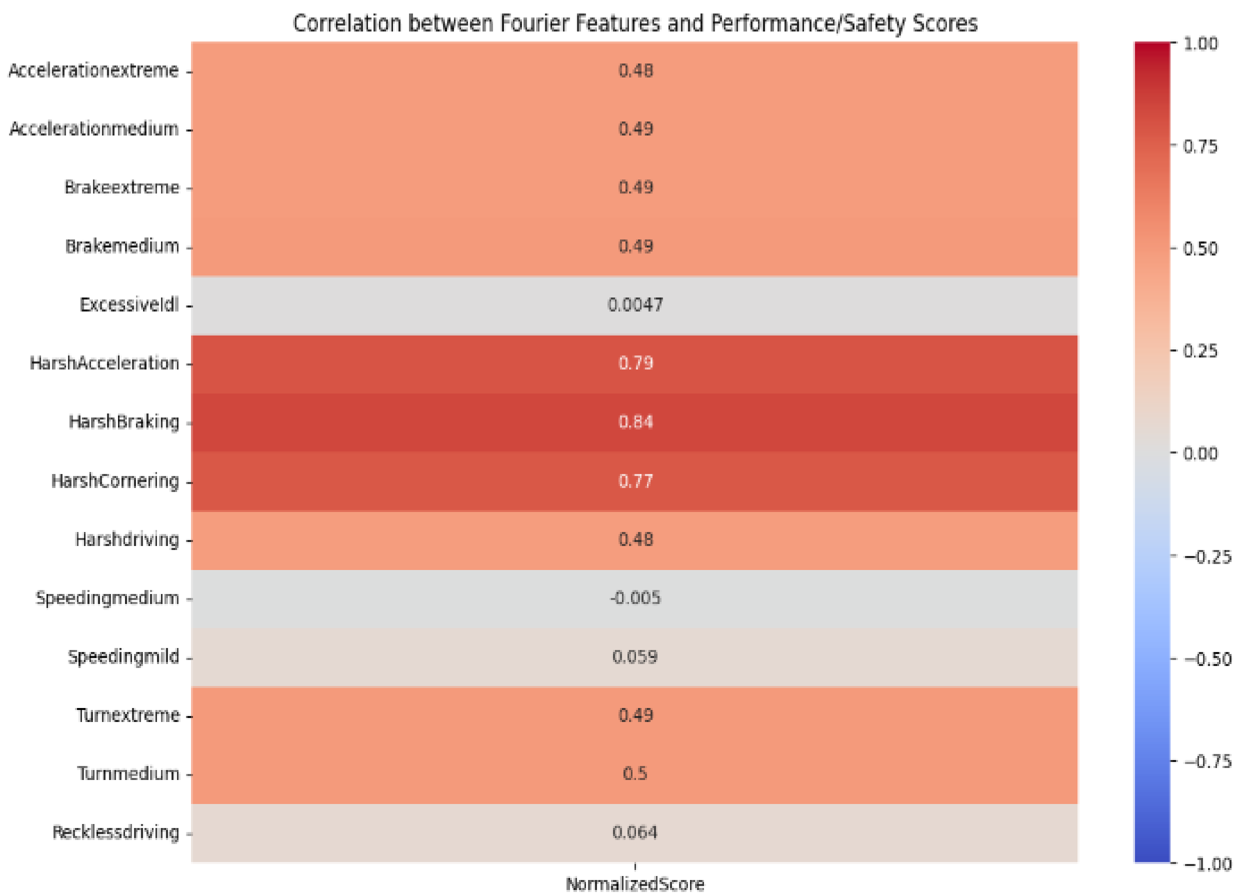


Figure 8 correlation between features and driving score

As the above visualization illustrates, we can see 3 types correlation with high positive correlation, moderate positive correlation, and low or no correlation of violation types committed with their respective scores.

The analysis reveals that there are high positive correlations between certain driving behavior and normalized driver score which suggest that these behaviors are key indicators of potentially unsaved or fatal driving. specially, harsh braking shows a very strong positive correlation (0.84) with normalized score, indicating that driver who frequently brake harshly tends to have high score, reflecting unsafe driving. similarly, harsh acceleration events have strong positive correlation (0.79) with normalized score, implying that drivers who accelerate harshly also tends to have high scores. Harsh cornering is another behavior that exhibits strong positive correlation (0.77) suggests that frequently sharp turns or aggressive cornering are associated with score.

In contrast, some behaviors exhibit low or negligible correlation with the normalized score. excessive idling has almost no correlation (0.0047),and medium and mild speeding events also show very weak or negligible correlation ranging from (-0.005 to 0.059).the findings suggest that harsh braking, harsh acceleration and harsh cornering are key indicators of higher normalized score .highlighting that intervention aimed at reducing this behavior should significantly improve driving safety and performance. The strength and direction of this relationship were measured using correlation coefficient formula which, calculate linear relationship between different types of driving behavior

Correlation: - lets measure the strength and direction of linear relationship between violation types. Using below correlation coefficient formula.

$$r = \frac{\sum(\mathrm{Xi} - \mathrm{X}(\mathrm{Yi} - \mathrm{Y})}{\sqrt{\sum(Xi - \bar{X})^2} \sum(Yt - \bar{Y})^2}$$

Where

- Xi and Yi are individual sample point
- $\bar{X}$ and $\bar{Y}$ are the mean values of the X and Y variables, respectively.

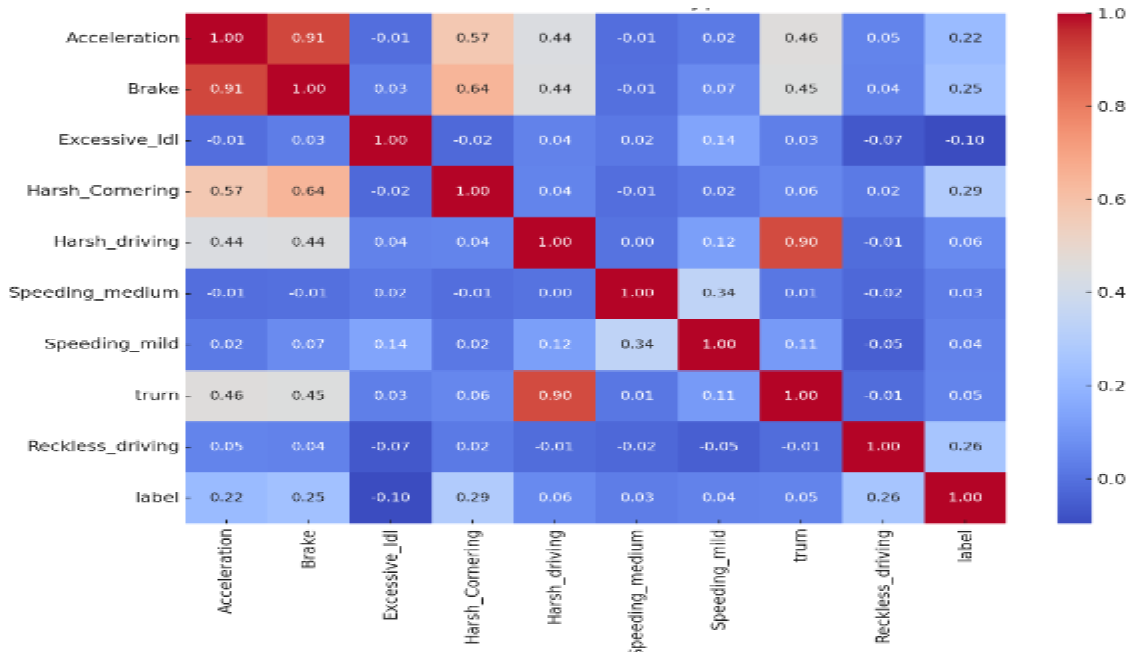


Figure 9 Correlation of Feature between violations

Above figure shows three correlation types of strong positive, moderate positive, weak and moderate correlations, and negligible correlations.

The analysis indicates several important correlations between different driving behavior. a very strong positive correlation (0.91) is observed between extreme acceleration and braking. suggesting that driver who frequently accelerate also tends to brake frequently. possibly indicating stop- and go driving pattern. additionally, there is strong correlation (0.91) between harsh cornering and harsh driving, implying that driver who often make sharp turns also likely to engage in other aggressive driving behaviors.

In terms of moderate positive correlation acceleration is moderately correlate with harsh driving (0.44) indicate that frequent acceleration are often part of boarder aggressive driving pattern. similarly, there is moderate between harsh driving and turn behavior (0.54), suggesting that aggressive driving also include sharp and frequent turns often occur alongside harsh cornering.

There is also weak moderate positive correlation such as between braking and harsh driving (0.44) indicating that frequent baking is associated with aggressive driving behavior. turning is similarly moderately correlated between harsh driving (0.45) showing, a link between frequent turning and aggressive driving patterns.

Conversely, negligible correlation is observed with behavior like excessive idling and mild speeding. excessive idling shows a very low negligible correlation with other metrics, suggesting that it is strongly related to other driving patterns. Mild speeding also has weak correlation with other metrics indicating that it is not a major factor in driving factor analysed.

3.3.2 Implication of the analysis

Driving behavior patterns: the strong correlations among acceleration, braking, and harsh driving suggest that interventions targeting one of these behaviors might influence the others.

Labels Analysis: while labels show some correlation with harsh driving and harsh cornering, additional features or metrics might be needed to fully understand the classification.

Focused intervention: efforts to improve driving safety could focus on reducing harsh driving behaviors, which are interconnected and collectively contribute to fatal driving pattern.

Therefore, the more the correlated the attributes to the target attribute, each of them is used to classify driver behavior.

3.3.3 Splitting the dataset

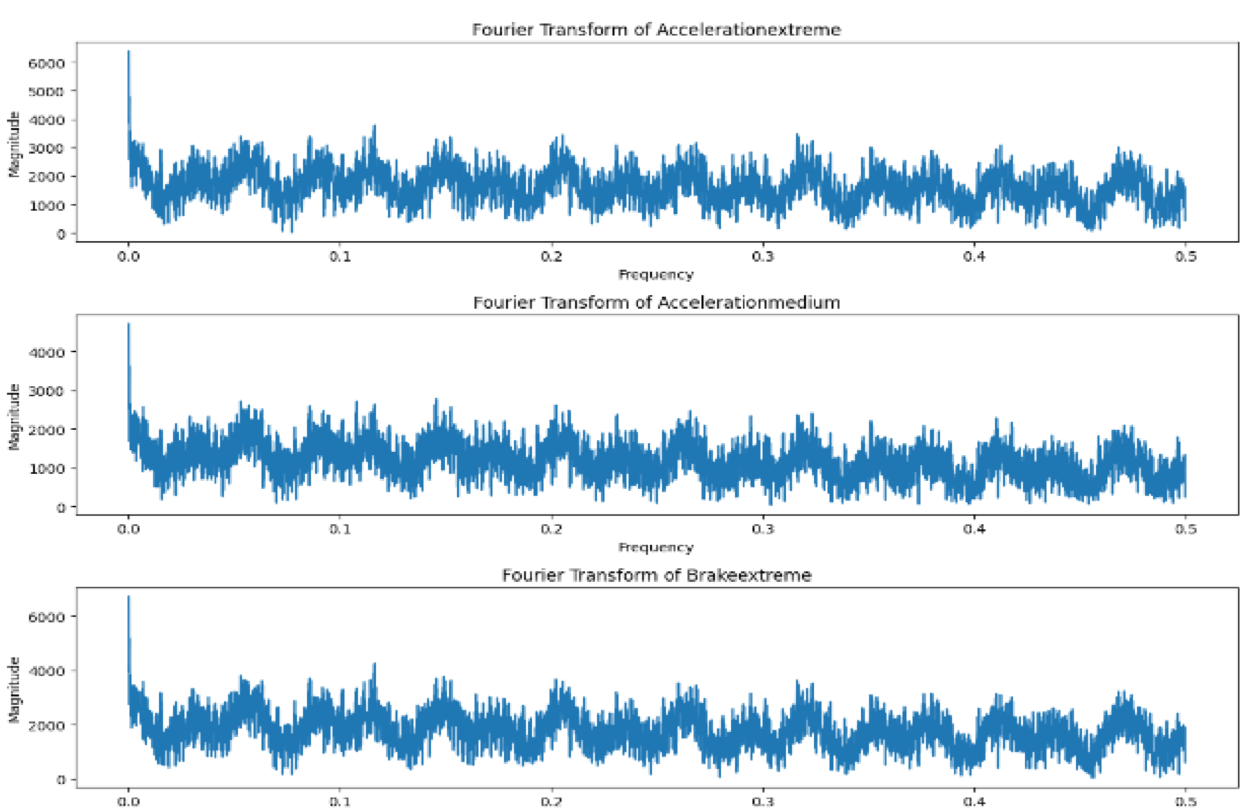
a simplest way to split the input for training and testing is select non- overlapping subsets of data while preserving the order of the data records. This approach is useful to evaluate the models on data for a certain date or within a certain time range.

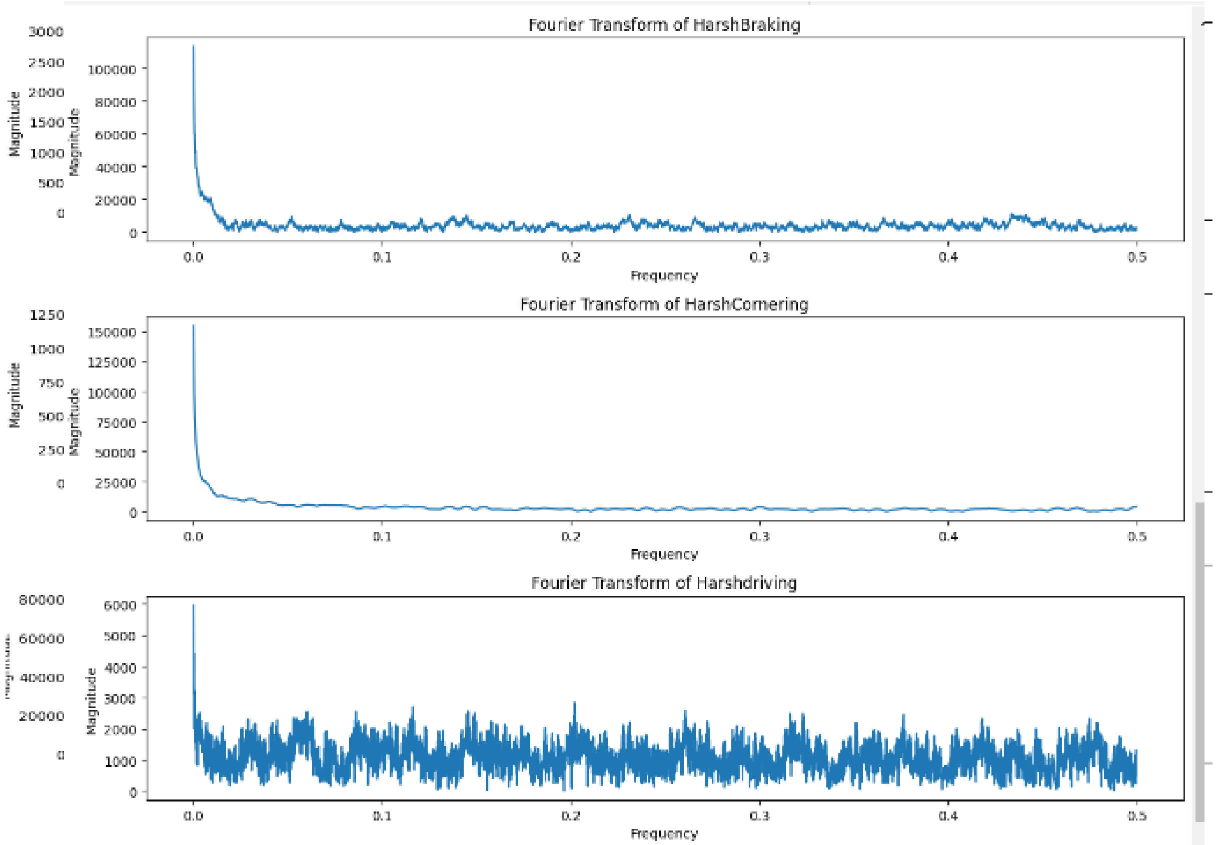
the beginning of the range of data for training and the end of the range of the data for testing might produce a more accurate estimation of the model quality than using records data drawn from the entire data range. The part of a data set used to train a model is called the training set. While a test set is used to evaluate the model's predictions. A random train/test split method was used depending on the size and shape of the problem because it had a significant computing advantage and accuracy over other approaches (ION, 2022). In this study, 80/20 ratios are used to split the dataset into train and test. Because the model has a good fit for it, this ratio value was chosen as a results of a random train/test split during model training.

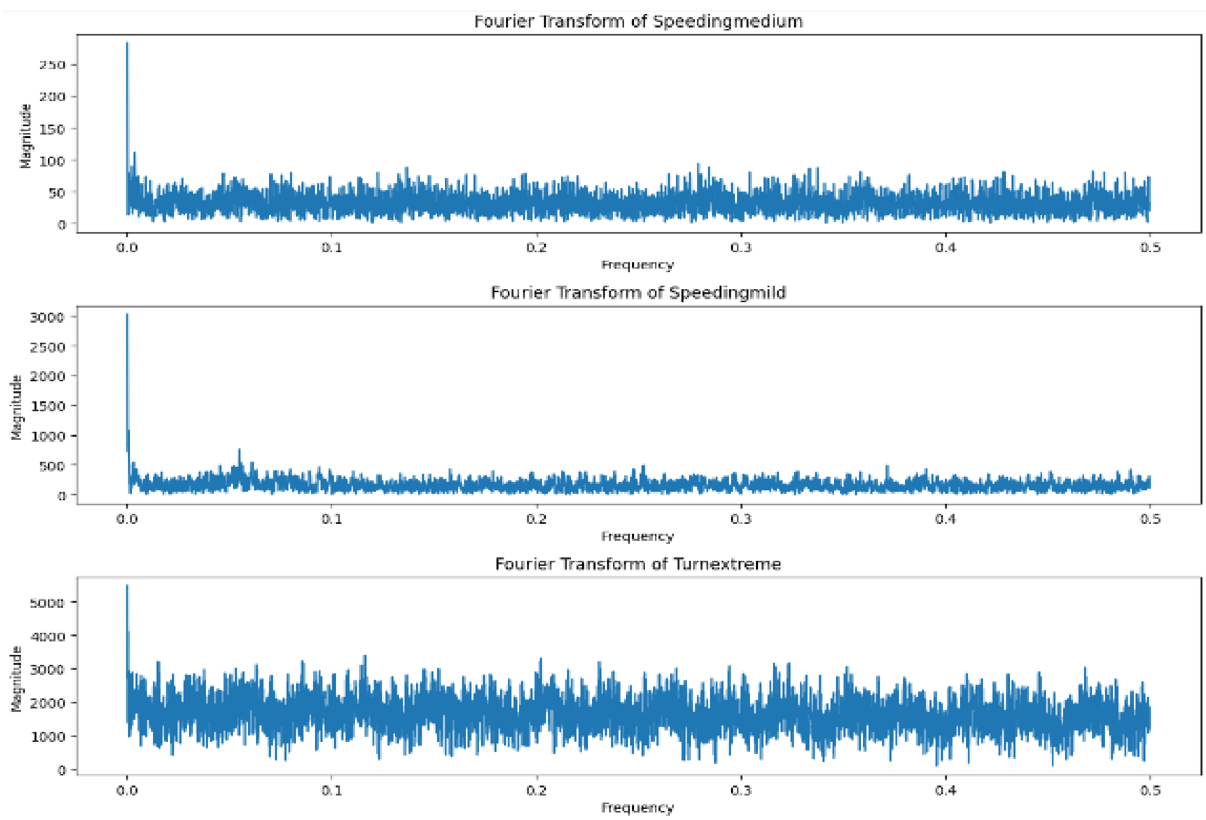
3.3.4 Data visualization

is the process of displaying data so that viewers may see its relevance. Visualization approaches could help in understanding and resolving the majority of business issues. Python has a wide range of modules that may be used to extract data's information. Users of Matplotlib can produce a variety of visualizations, including histograms, scatterplots, bar graphs, and pie charts**.** The graph below shows the researcher's detailed summary and analysis of driver behavior classification for additional information and classification. We have used Fourier transformation because of like

EEG may not always reveal important patterns clearly. Fourier transform helps to convert the time domain data into the frequency domain where different frequency components (e.g. low, medium, high frequency behaviors) can be identified.




Fourier Transform of HarshBraking
Magnitude
Frequency
Fourier Transform of HarshCornering
Magnitude
Frequency
Fourier Transform of Harshdriving
Magnitude
Frequency


Fourier Transform of Speedingmedium
Magnitude
Frequency
Fourier Transform of Speedingmild
Magnitude
Frequency
Fourier Transform of Turnextreme
Magnitude
Frequency

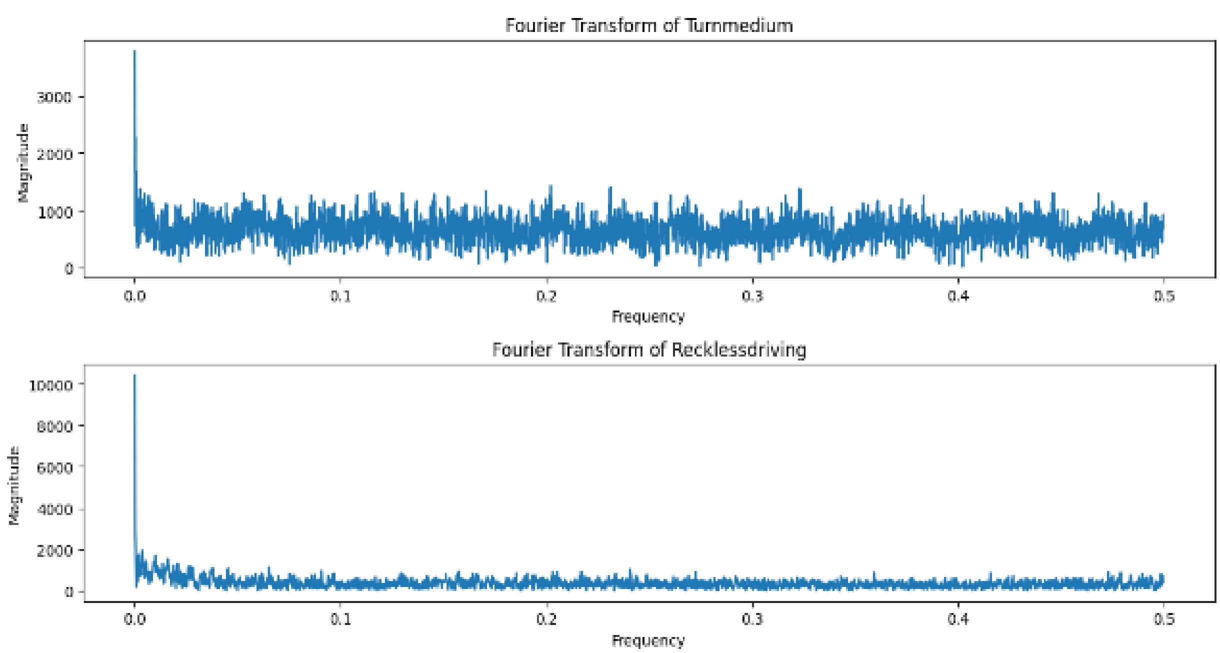


Figure 10 observing data's as signals

3.4 Evaluation methods

The following are some techniques we used for assessing deep learning-based driver behavior classification from sensor data:

Classification Metrics:

- ✓ **Accuracy**: The overall percentage of correctly classified driver behavior instances. (e.g., (# correctly classified) / (# total instances))
- ✓ **Precision:** Measures the proportion of positive predictions that were actually correct for a specific driver behavior class. (e.g., True Positives / (True Positives + False Positives))
- ✓ **Recall**: Quantifies the percentage of real positive examples that were accurately recognized for a particular driver behavior class. True Positives / (True Positives + False Negatives) are two examples.
- ✓ **F1-Score**: A single score derived from the harmonic mean of precision and recall. (For instance, 2 * (Recall * Precision) / (Recall + Precision))

3.5 Development Tools and Techniques

3.5.1 Design Tools

Design tools are used in the creation, presentation, and interpretation of design concepts. The primary tool used to create the system is called Microsoft Visio. It is lightweight, powerful, and used to produce professional-looking flowcharts, network diagrams, and other design concepts. In addition to Enterprise Architect, various web platforms are employed in this study for design purposes.

3.5.2 Hardware Tools

Bellow hard ware tools would be use for the completion of the research.

Table 10 Hardware Tools used

No.	Tools	Specification	Used for
1	CPU	Core I3	
2	HARD DRIVE(HDD)	500 GB	
3	RAM	4GB	

3.5.3 Software Tools

The following software tools would be used to implement this research.

Anaconda: - gives you access to a toolbox designed for scientific study. R or Python coding is possible after installing Anaconda. There are 1500 packages in it. For writing text, they perform a similar function to text processors like Microsoft Word, Google Docs, and Pages, but in reality, they are much more. IDEs offer a variety of helpful capabilities for writing and editing code, inspecting and visualizing data, storing variables, displaying results, and working together on projects. Anaconda's capabilities include the ability to share, collaborate on, and replicate projects as well as import data from files, and databases.

Python: - is a high-level programming language used mostly for creating websites and apps. Guido van Rossum created it for the first time in the Netherlands .Python will be used in this study for data collection, pre-processing, model creation, and performance assessment. Python has a number of advantages over other programming languages, including being free and open source, being more user-friendly, and having superior visualization tools.

Jupyter Notebook: - offers a collaborative environment that makes it possible to develop and share files that contain narrative content, stay code, and graphics. It was once known as Python Notebooks It employs modeling, simulation of numbers, statistics manipulation, and visualization. Jupiter notebook will be used in this study to develop and run code. Some of the capabilities of the notebook include the ability to share code with anyone, assist the data scientist in documenting the evolving analysis process, and display images produced by the code-running cells.

Pandas: - is a Python module for handling large amounts of data. It provides resources for data exploration, modification, cleaning, and analysis. Large volumes of data can be evaluated with the help of pandas to draw conclusions. Pandas is a tool for analyzing and manipulating data. Wes McKinney created it for the first time in the year 2008. Pandas is a good choice for tabular data having columns of different types, such as those found in an Excel or SQL table. Tools for reading and writing data or importing datasets, reshaping and pivoting of data sets, dataset merging and joining, and data structure column insertion and deletion are some of the capabilities of pandas.

NumPy: - is a scripting language for Python. Its acronym is "Numerical Python." The predecessor of NumPy, Numeric, was developed by Jim Hugunin. A new package called Numeracy was also developed with some added features. Travis Oliphant combined the features of the Numarray and Numeric packages to create the NumPy package in 2005. There are many contributors to this open-source project. A high-performance multidimensional array object and capability for manipulating these arrays are both included in this scientific Python module. Powerful N-dimensional array objects, sophisticated functions, tools for integrating other programming languages, handy for linear algebra, and capabilities for random number generation are some of the features of NumPy.

Matplotlib: - one of the most popular Python data visualization packages. A cross-platform library uses array data to create 2D charts . Although NumPy and other extension code are widely utilized by matplotlib, which is essentially built into Python, matplotlib offers quick performance even for large arrays. Matplotlib is a NumPy extension for numerical mathematics and a fantastic tool for viewing or presenting data in a graphical or pictorial

fashion. In this study, the dataset was represented graphically and graphically using matplotlib.

Keras: - is a Python-based, user-friendly, high-level neural network API that can operate on top of TensorFlow. Google advertised it as being quick and simple to use . In this study, we'll use KARAS to define and train the model with various tuned hyperparameter values, like dropout layers and activation function. The fundamental characteristics of Kera are that it supports layers like dropout, batch normalization, and pooling that are required to generate convolutional and recurrent neural networks, as well as neural network building pieces such layers, activation functions, and optimizers.

Tensor Flow: - the most well-known and widely applied deep learning framework. It was created by Google in 2015 and is supported by mobile devices, CPUs, and open-source platforms. TensorFlow has a number of advantages, including the capacity to process large amounts of data efficiently across many machines, support for both machine learning and deep learning models, and improved performance when dealing with scientific expressions with multidimensional metrics.

The following software tools would be used to implement this research.

Table 11 Software Tools

No	**Tools**	**Description**
1	Python	Programming used to implement and demonstrate this thesis requires many of the drivers used to configure and package to install some device with the computer.
2	Tensor Flow	Python deep learning library in the back end.
3	Keras	Will be used to construct deep neural network.
4	Matplotlib	Will be used for data visualization.
5	Anaconda	An application used to install the up-to-date version of python with its different modules and IDEs.
6	Jupyter Notebook	The most popular and handy IDE among AI and deep learning researchers to work with python.
7	Google Colab	Allows to write and execute python in browser, with no configuration required, Access to GPUs free of charge, Easy sharing

CHAPTER FOUR

4. PROPOSED MODEL AND ARCHITECTURE

4.1 Chapter Overview

Deep learning algorithms have many layer types that are created for various reasons using classifiers. For this work, three distinct types of traditional machine learning and deep learning models were trained, and the accuracy of their predictions is evaluated. The performance of the deep learning model is evaluated using a collection of data on driver behavior classification, and the best approach was selected using the model assessment metrics.

4.2 Problem formulation

The goal of this research is to develop a robust deep learning model to classify driver behavior using given dataset of driving metrics. The classification task involves predicting safety label of the driver based on various recorded driving behavior. This study leverages advanced deep learning techniques to capture intricate patterns in the data that traditional machine learning model might miss. The dataset consists of several driving behavior metrics, including acceleration, braking, excessive idling, harsh cornering, harsh driving, speeding, turning, and reckless driving. additionally, each record is labeled with safety classification, indicating different levels of driving safety. This research aimed to develop models that can accurately classify driver safety labels based on recorded driving behaviors. The insight gained from these study can contribute to improve driver training programs, enhancing road safety, and reducing accident rates by identifying and mitigating risky driving behavior.

4.3 Architecture Proposed Model

The main objective of this research is to develop driver classification model using deep learning models like CNN, RNN, and LSTM in order to see which deep learning model is the most suitable model for classification of driver behavior .the model tried to apply and process data by collecting six month freighting data of the drivers obtained from one transport company(National Transport PLC) which has 150 fleet size .around 14 types of violation has been selected for analysis which is sevier,less-servier,and fatal. The proposed architecture of this study serves as framework for implementing the model. the step by step process involved in achieving the desired outcome clearly outlined in the proposed architecture, providing a general representation of the system.

To start, the study involves collecting relevant data from National transport plc, which was transport company that have 150 truck and the trucks main duty was to load cement, coal, metal, and other material to the customers and factories. The data set we found was extracted from GPS monitoring system. which, serve as the dataset for model. The collected data undergoes preprocessing to its suitability for deep learning process. Preprocessing tasks include such as normalization, augmentation, and adding some noise to reduce overfitting, which help to refine the data for further analysis.

The next step involves building classification neural network model such as CNN.RNN and LSTM.these models are specifically designed to address the selected driver behavior classes by tuning the training process and employing hyperparameter tuning techniques, the models are optimized for accuracy and performance.

Once the models are build, they are evaluated using appropriate evaluation metrics to access their effectiveness in classifying driver behavior. The model with the highest accuracy and lowest loss function is selected as he final driver behavior classification model.

The proposed architecture of the predictive model is depicted in Figure below. This architecture demonstrates the flow and components of the model, providing a visual representation of its structure and functionality.

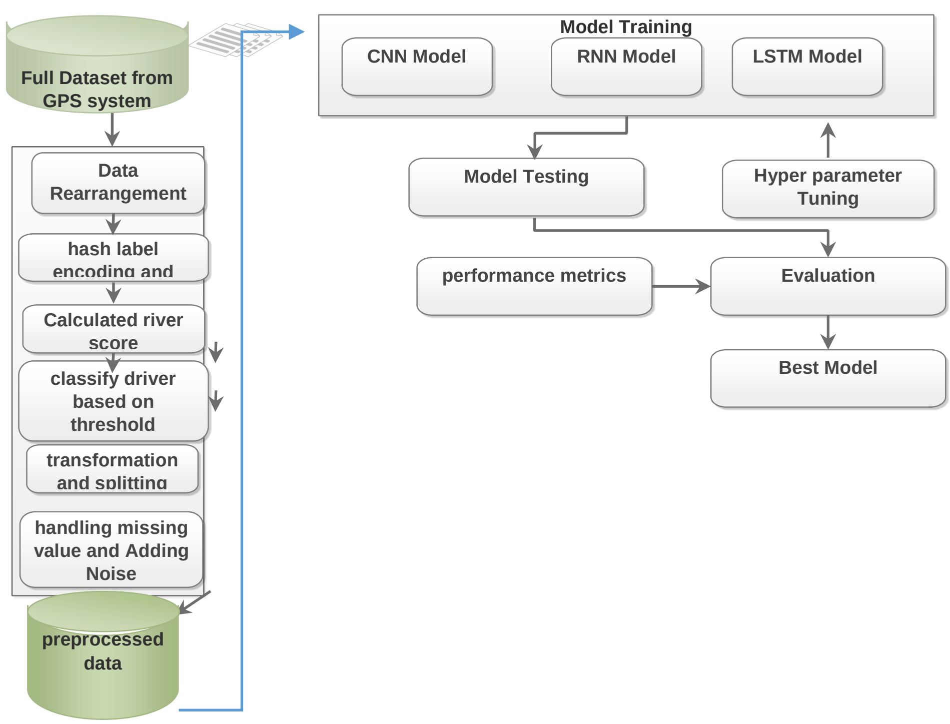


Figure 11 detailed Architecture of proposed model

4.4 Data Set for Benchmarking

The benchmark dataset was collected using a Samsung Galaxy S10 smartphone and a Dacia Sandero 1.4 MPI which had 75 horsepower. (RoCHI, 2015).

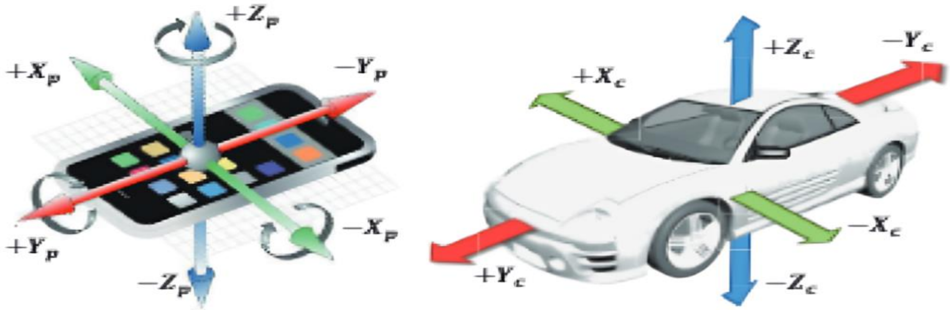


Figure 12 Coordinate systems of smart phone and vehicles:

training file is composed of 5945 recordings; 2332 are classified as slow, 1713 as aggressive, and 1900 as normal. Each set of instances is collected on the same road section. The number of instances varies between classes because selected two readings per second, and driving aggressive takes less time than driving normal and slow. We have almost the same proportion of instances for each of the two datasets.

The test dataset is composed of a slightly lower number of instances: 785 divided into 273 for slow, 297 for normal and 214 for aggressive, which reflects the proportions from the training dataset but with lower values. The route had a smaller distance and took less time for logging, influencing the number of instances collected.

As stated on the above chapter this study used a dataset which is extracted from GPS system of one Transport companies 150 trucks by taking their six month continuous movements with their violation records. Dataset is composed of 7178 recordings; 2921 are classified as Fatal, 2706 as less-severe and 1551 as severe. Each set of instances is collected on the same road section.

4.5 Data Preprocessing

The second phase of the proposed model is the preprocessing phase that occurs after the finalizing our dataset. The full dataset which are used for benchmarking is provided without any preprocessing.

The input data should be a sequence of days, since the raw data extracted does not provide enough information about driver behavior. The models that will be built aims to find which data are related to fatal driving, then, it will become a classification problem. Also we take some adding noise and normalization techniques to prevent overfitting.

We also evaluated the importance of each feature by random forest classifier to identify which feature contribute the most prediction. And we can consider removing less important features. then, after identifying key features we trained a model by introducing dropout layers to randomly drop units during training, helping the model generalize better.

Also we have implemented early stopping during the model training to stop training when the validation performance starts to deteriorate. Below is graph to show feature importance.

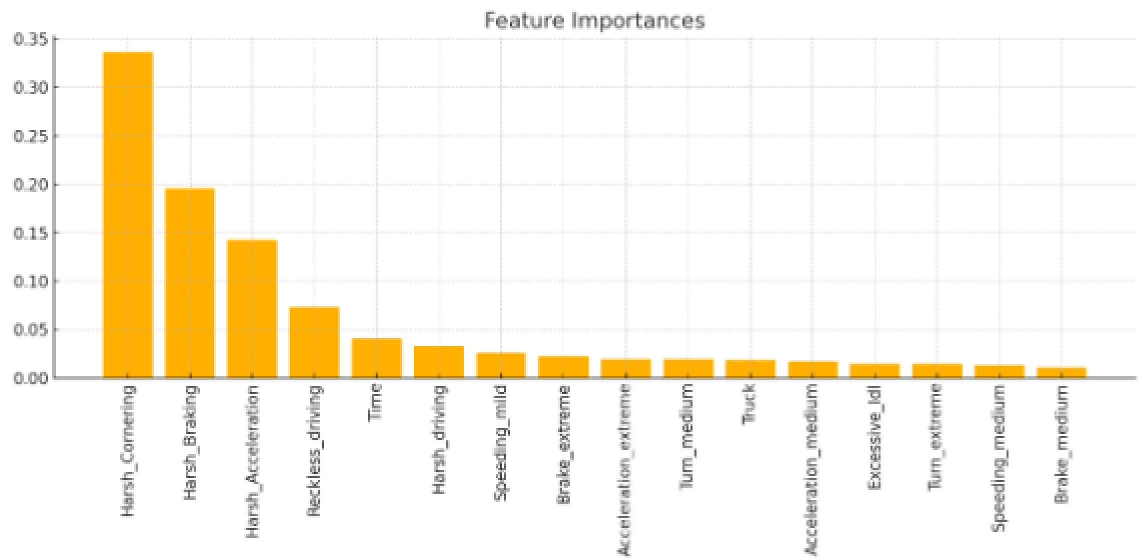


Figure 13 feature importance

4.6 Proposed Models Architectures

After careful consideration of various models for driver behavior classification in in driver behavior classification domain, we have selected LSTM as our preferred choice.

We extensively evaluated alternatives GRU, CNN, from deep learning methods but LSTM stood out due to its unique advantages in capturing temporal dependencies and handling complex patterns in driver behavior data.

4.6.1 CNN

Extracted driver behavior data from GPS system can be represented as a time series. By creating windows of this data and stacking them, we can form a 1D-like representation similar to an image. CNNs excel at processing such data.

CNN Architecture for Driver Behavior Classification:

Input Layer: This layer receives the preprocessed sensor data, reshaped into a 1D-like format (e.g., window size x number of violations).

Convolutional Layers: These layers' extract features from the data. Each layer applies multiple filters that learn to detect specific patterns within the sensor data windows.

Pooling Layers (Optional): These layers downsample the data, reducing its dimensionality while preserving important features. This helps control overfitting and computational cost.

Flatten Layer: This layer transforms the high-dimensional output of the convolutional layers into a one-dimensional vector.

Fully Connected Layers: These layers perform the final classification. They take the flattened feature vector and learn to map it to different driver behavior categories (e.g., aggressive, normal, slow).

Table 12 Convolutional layer Components for CNN Model

Layer	**Filters**	**Kernel Size**	**Activation**
Conv1D_1	64	3	ReLU
Conv1D_2	128	3	ReLU
Conv1D_3	256	3	ReLU

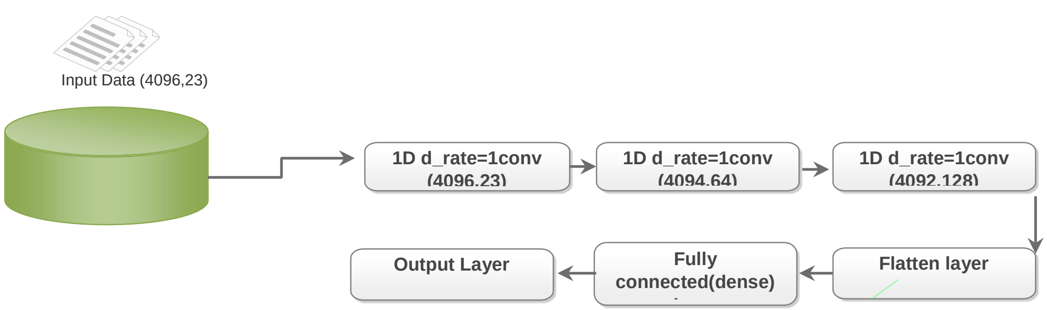


Figure 14 layers of CNN

4.6.2 LSTM Model

LSTM is a special kind of recurrent neural network, which was proposed by Hoch Reiter S and Schmid Huber J. LSTM adds memory storage (cell) and Gates structures to the traditional recurrent neural network. The cell is used to record the state of neurons, and the gates pass through information selectively, by means of a sigmoid neural layer and a point- by-point multiplication mechanism. With three gate structures similar to filters, LSTM realizes information protection and control: forgetting gate determines the information discarded from the cell state, input gate determines new information added to the cell, and output gate filters the final output information.

Table 13 layer Components for LSTM Model

Layer (type)	Output Shape	Units/Filters	Type	No of param
LSTM	(None, 128)	128	LSTM Layer	76,288
Dense	(None, 128)	128	Dense Layer	16,512
LSTM	(None, 1, 128)	128	LSTM Layer	131,584
Dense	(None, 1, 256)	256	Dense Layer	33,024
Dense	(None, 1, 128)	128	Dense Layer	32,896
Dense	(None, 1, 64)	64	Dense Layer	8,256
Dense	(None, 3)	3	Output Layer	1

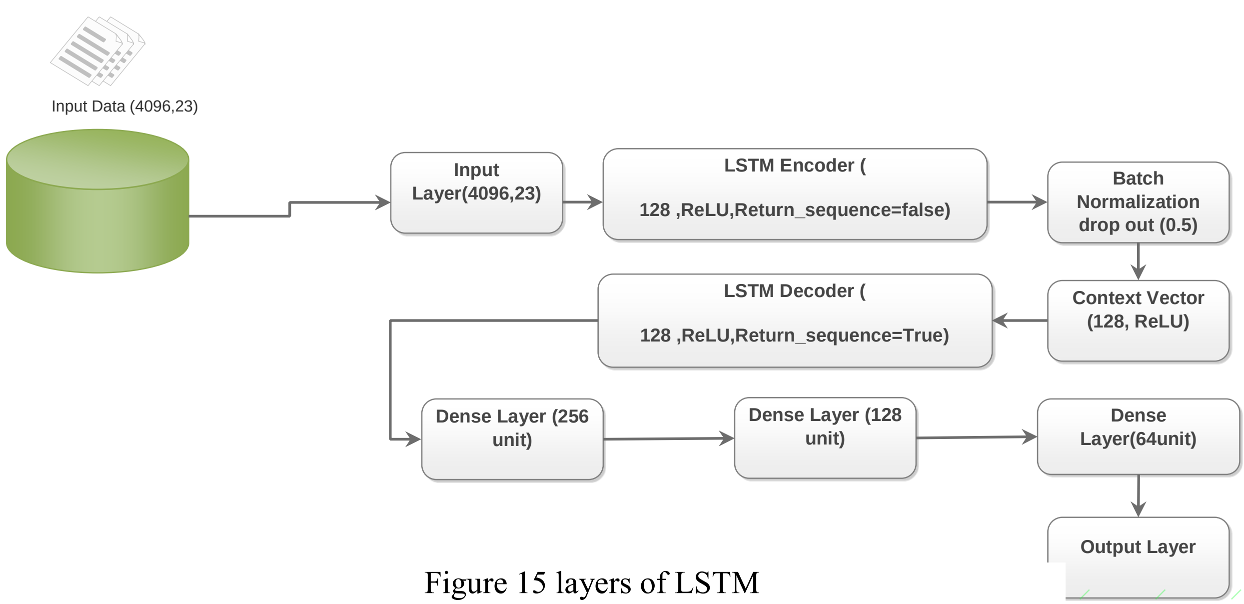


Figure 15 layers of LSTM

4.6.3 GRU Model

The inter-neuronal connection structure of RNNs with at least one cycle is known as a GRU. GRU is used to handle time-series problems and solves the disappearing and exploding problem in classical RNN while learning long-term dependencies. Kyunghyun Cho created GRU for the first time in 1997. Only two gates make up the framework of an RNN. The data flow into and out of memory is controlled by the reset and update gates, respectively. In a manner similar to the LSTM but without the requirement of a separate memory cell, GRU gate units regulate the flow of data within the unit.

Table 14 layer Components for GRU Model

Layer (type)	Output Shape	Filters/Units	Param #
GRU	(None, 20, 64)	64 units	12,864
Batch Normalization	(None, 20, 64)		256
Dropout	(None, 20, 64)		0
GRU_1	(None, 128)	128 units	74,496
BatchNormalization_1	(None, 128)		512
Dropout_1	(None, 128)		0
Flatten	(None, 128)		0
Dense	(None, 256)	256 units	33,024
BatchNormalization_2	(None, 256)		1,024
Dropout_2	(None, 256)		0
Dense_1	(None, 3)	3 units	771

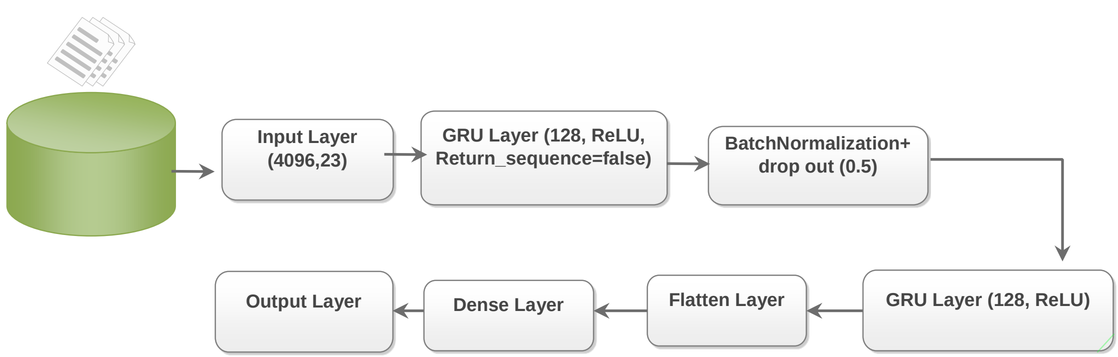


Figure 16 layers of GRU

4.7 components of proposed model

Our proposed model presents a deep learning architecture designed for time-series classification using and LSTM-based encoder-decoder framework. The model begins by introducing Gaussian noise to the input data to enhance robustness, with the noise. This preprocessed data is then split into training and test sets while maintaining class distribution using stratified sampling.

4.7.1 Input Layer

This Input layer is designed to accept multivariate time-series data, where the input shape is (time steps, features). this layer fid the data into model, allowing the subsequent layers to process and extract meaningful features over time.

4.7.2 Encoder LSTM layer

The encoder is crucial component designed to summarize the sequential information.it consist of an LSTM (Long Short Term Memory) layers with 128 units and uses the ReLU activation function. The LSTM is configured with 'return_sequences=False', meaning it only outputs the final hidden state which serve as summary of the input sequence. This summary is then passed on further processing. LSTMs are particular useful for capturing long-term dependencies in time-series data due to their ability to mitigate vanishing gradient issues.

4.7.3 Batch Normalization and Dropout Layers

After the encoder LSTM, batch normalization is applied to stabilize and accelerate the training process by normalizing activations. A dropout layer with dropout rate 0.5 is then used to prevent overfitting by randomly disabling half of the neurons during training. This layer help insure that the model generalizes better to unseen data.

4.7.4 context vector

The encoder sequence from LSTM is passed through a dense layer with 128 units and ReLU activation to create a context vector. The context vector represents the most salient features extracted from a sequence, condensing the information needed for the subsequent decoding phase. This context vector is then repeated across time steps using the 'RepeatVector' layer to make it compatible with the decoder.

4.7.5 **Decoder LSTM Layer**

The decoder consists of another LSTM layer with 128 units and ReLU activation, configured with ‘return_sequences=true’ to output a sequence of hidden states corresponding to each time step. This allows the decoder to generate outputs for each time step in the sequence. The output of the decoder LSTM undergoes batch normalization and dropout for the same reasons as in the encoder phase, ensuring stable and regularized training.

CHAPTER FIVE

5. IMPLEMENTATION OF THE PROPOSED MODEL

5.1 Chapter Overview

In this chapter, the implementation of the proposed model architecture is discussed. The working environment, LSTM (Short Long Term Memory) experiments undertaken will be discussed.

5.2 Working Environment

This section describes the environment setup of the hardware components that are used in the implementation of the modeling process with their specifications.

Laptop: utilized for design the model

- ✓ RAM: 12GB
- ✓ Desktop: utilized for pre-processing the data set
- ✓ Operating System: Windows 10
- ✓ Processor: Intel ® Core™ i3 CPU @ 2.50GHz
- ✓ System Type: 64-bit operating system, x64-based processor

5.3 Environment Setup

In order to do our study, we have used particular programming and IDEs.

JupyterLab: - is an open-source web-based interactive development environment for data science, machine learning, and scientific computing. It is a flexible and extensible platform that provides a user-friendly interface for working with Jupyter notebooks, code editors, terminals, and other interactive tools.

Colab: - is a Google cloud-based notebook that allows you to create Python code on any browser with sufficient computing power (CPU, 12GB RAM) and resources to train with the CPU and GPU.

5.3.1 Tools and Python Packages

This section describes all tools and python packages that are used in the implementation of the modeling process with their specifications. Those libraries are listed, along with their version and description.

Table 15 Packages used for the study

Libraries	Version	Description
Numpy	1.24.3	It is a powerful library in Python for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a wide range of mathematical functions to operate on these arrays efficiently.
Tensorflow	2.12.0	It is an open-source machine learning framework used for building, training, and deploying machine learning models and neural networks. It provides a comprehensive ecosystem of tools and libraries for tasks such as computer vision, natural language processing, reinforcement learning, and large-scale distributed computing.
Keras	2.12.0	It is a high-level neural networks API written in Python. It is designed for fast and easy experimentation with deep learning models.
Sklearn	1.2.2	Sklearn simplifies the development and deployment of machine learning solutions and is widely used in academia, research, and industry.
Pandas	1.5.3	It provides powerful data structures, such as DataFrame and Series, that allow for efficient handling and manipulation of structured data. Pandas offers functionalities for data cleaning, transformation, merging, slicing, and aggregation.
Seaborn	0.12.2	Seaborn simplifies the process of creating plots such as scatter plots, bar plots, box plots, and heatmaps.
Matplotlib	3.7.1	Matplotlib is a widely-used Python library for creating visualizations and plots.

5.4 Training Procedure

In the training procedure the following process has taken place in order to achieve the goal of this research.

- ✓ Data collection
- ✓ Data preprocessing
- ✓ Configuring the CPU and the dataset
- ✓ Implementing the proposed solution.

The technique is outlined in detail in each phase to provide further information on what has been done.

5.5 Dataset Description and Loading of Dataset

5.5.1 Preprocessing Implementation

This chapter compiles all of the experimental results along with a variety of descriptions. Before supplying the dataset to the prediction model, the researchers carry out feature engineering and data pretreatment operations. For the dataset, the train-test splitting ratio was established at 80/20. While the training data is used to train the model, the test set is used to evaluate the model's performance. Following data preparation, the researchers used the deep learning algorithms GRU, CNN, and LSTM to build a model that addressed the study goals and evaluated the system architecture. The following hyper parameters were used in the model's implementation and assessment: the dropout layer, activation function, optimizer, loss function, and evaluation metrics.

The researchers generally conduct three experiments to classify driver behavior and assess the effectiveness of models. Finally, using historical data that has been preprocessed, the researchers examine the effectiveness of state-of-the-art deep learning systems for classifying driver behavior.

5.6 Model Implementation for LSTM

5.6.1 Loading the Data Set

In order to train our model using (TPU) Tensor Flow Processing Unit we need to load our data set from our local working directory to Colab. Our Data Set mainly consist of 3 driver behaviors as classes as shown on figure 17.

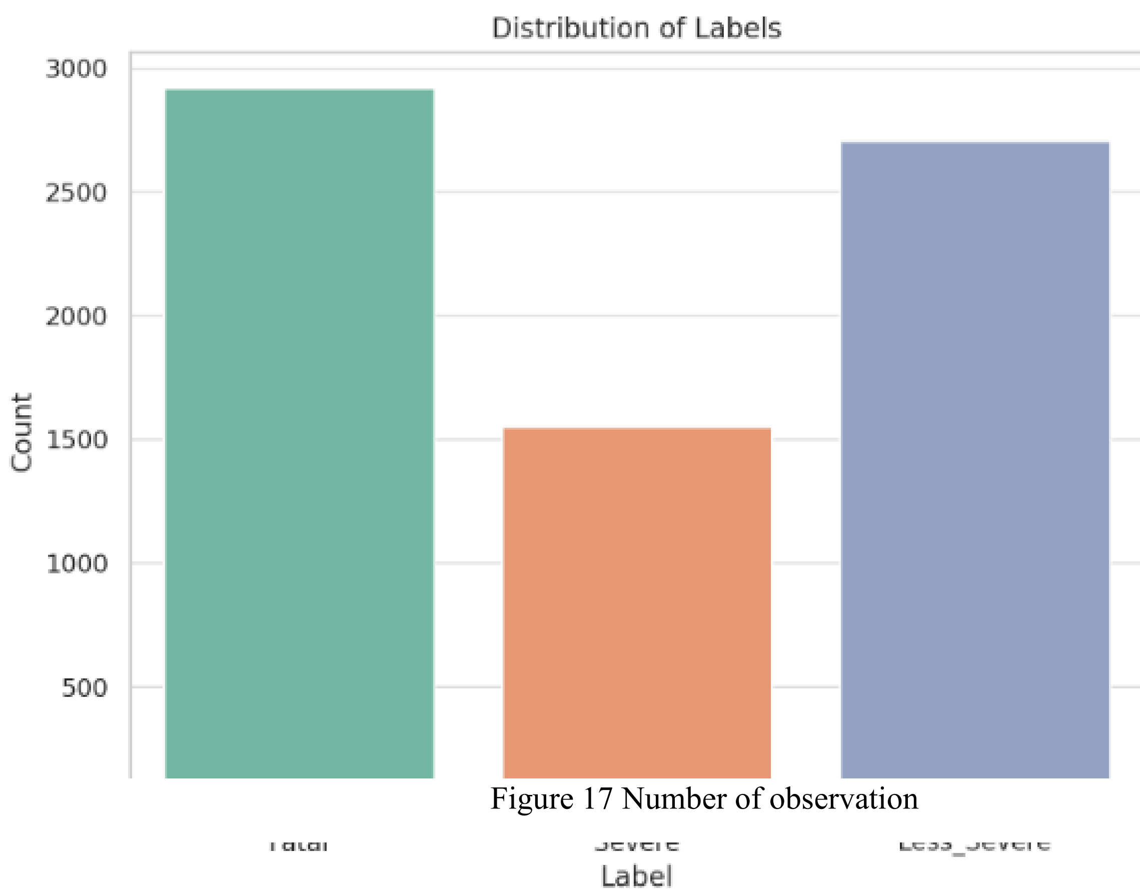


Figure 17 Number of observation

5.6.2 Hyper- parameters

Hyper parameter tuning is a crucial aspect of machine learning model development, as it directly influences the model's performance and generalization ability. Each hyperparameter plays a specific role in shaping the behavior and effectiveness of the model. The table below discusses the significance of each hyperparameter used for our models and how they may impact the model's training and performance.

The following hyper parameters were set up for this study.

Create Nodes: Using 64,32 nodes on the first and second layers of the neural network model, ReLU was utilized as the activation function, and 1 neuron was employed on the output layer. Based on trial and error during model training, this value of the hyper- parameters was chosen since the model has a good fit for it.

Evaluate Model: The regression model evaluation metrices Accuracy, Recall, and Precision were used to assess the model's performance.

Compile Network: A mean squared error loss function was set up by default and the Adam optimization technique was applied.

Training Phase: Three distinct models are proposed by the researcher. These models are the GRU, LSTM, and CNN. A dense layer, an activation layer, and a dropout layer make up its structure. The models' input is previous customer data on driver behavior, and their output is predicted driver classification based on that history.

Testing Phase: To assess how successfully the system adapts to unknown datasets, Driver classification data different from the training datasets have been provided to the learnt model. The prediction model should be trained with training data after being constructed. In order to train the suggested system, fit routines were employed.

Batch Size: gives the number of samples that must be processed before the internal model parameters can be changed. When predictions are made after iterating through one or more samples, it is comparable to a for loop. The models were evaluated using a range of batch sizes by the researcher. The optimal batch size was chosen after much trial and error in order to assess the model and make a forecast.

Epochs: The model employs a batch size of 64 and has the best fit on a value of 100 epochs. These epochs were chosen because the researcher obtained superior results using them, and after 100 epochs, there was no significance influence.

Validation Data: The data are what keep our model in training and are what are used to estimate the model. In order to choose the final models, it is utilized to estimate the model objectively. The test data and the test target are its two parameters.

Dropout: - Deep networks frequently run more slowly. Dropout is a regularization technique that addresses this issue by permitting the integration of exponentially many different neural network models in a reasonable amount of time during training. The term "dropout" refers to the removal of all incoming and outgoing connections of the model architecture. As a result, on the feed-forward pass, the links between deleted units and the activation of downstream units are temporally ignored. As a result, the backpropagation updates for the associated parameters are erased, and the neighbors cannot rely on the dropped units for parameter tuning. Avoiding overfitting is the main objective of dropout. In order to prevent the model from fitting either too well or too poorly, the researcher used 0.5 dropouts.

Early Stopping: - When training a learner iteratively, it is a sort of regularization used in machine learning to reduce overfitting. Iterative learning methods like gradient descent can be regularized by ceasing training as soon as the validation error decreases to a minimum.

Early stopping is the term used to describe this. The algorithm becomes better over time, and when the validation error reaches a certain threshold, training on the training set simply finishes early.

Activation Functions: It decides whether a particular neuron is stimulated or whether the information that the cell receives is important. The activation function (data) complexly changes the signal that was received. The input layer of a neural network, which consists of three layers, does not utilize an activation layer because it does not do any computation. Information entered through the input layer is processed in a number of ways by hidden layers, which then deliver the results to the output layer. In the hidden and output layers, activation functions are naturally used. We use a continuous activation function to solve this issue .

ReLU Function: it represents rectified linear unit and nonlinear activation function that is widely used in the neural network. The main advantage of this function is not activating all neurons simultaneously i.e., the neuron will be deactivated only when the output of linear transformation is 0. It can be defined mathematically as:

$f(x) = max(0, x)$..

During the back-propagation phase of neural network training, weights and biases are left unchanged when the gradient value is 0 .

Optimizer: The objective is to minimize the loss function, which is the difference between the expected and actual outputs. Through a technique known as gradient descent, we must run numerous iterations with various weights in order to find the ideal weight value in order to minimize the cost function. Optimizers are techniques or procedures that modify a neural network's weights and training rate in order to reduce losses.

Adam: - is a deep learning model update technique that, as opposed to the conventional stochastic gradient descent techniques, may be used to iteratively update network weights depending on training data.

Table 16 Hyperparameter for Proposed Model

Name	**Value**	**Use**
Batch Size	64	This hyperparameter determines the number of samples that are propagated through the model in each training iteration. A batch size of 64 means that the model will process 32 samples at a time before updating its parameters.
EPOCHS	100	This hyperparameter determines the number of times the entire dataset will be passed through the model during training. With 100 epochs, the model will be trained on the entire dataset 100 times.
Input Shape	(20,1)	This hyperparameter represents the shape of a block in the model. It is defined as (20, 1), implying that each block has 20 rows and 1 column.

CHAPTER SIX

6. RESULTS AND DISCUSSION

6.1 Chapter Overview

After discussing the implementation concepts and specifics of this work earlier, this chapter examines the outcomes of the suggested solution and offers a comparative explanation of the results with graphs and tables.

6.2 Result Discussion

In our research, we utilized our preprocessed dataset to assess how well our proposed models performed. We measured the effectiveness of the models by calculating precision, recall, F-1 score and accuracy scores. We examine 3 distinct deep learning models (CNN, GRU and LSTM) for our experiments and assessed their performance using standard metrics Results on Traditional Machine learning.

6.2.1 Deep Learning Model

6.2.1.1 CNN Model (Recurrent Plot Approach)

In the first experiment of our study, we employed the CNN model, a deep learning architecture commonly utilized for driver behavior classification tasks. During training, we conducted 50 epochs for each batch-size of 32, with each epoch consisting of 5743 training steps. With learning rate of 0.01 and we use sigmoid and ReLU Activation functions, With the help of the CNN approach and historical data on driver behavior, the researchers were able to acquire a training score of 80%.

Table 17 Tune hyper parameter values of CNN

Hyper parameters	Values
Train and test split	80,20
Input layers	20, 1
Hidden layer	6
Output Layer	1
Dropout	0.5
Bach size	64

Optimizer	Adam
Activation	relu
Epochs	100

CNN Model Classification Report:

	precision	recall	f1-score	support
0	0.91	0.86	0.89	585
1	0.62	0.45	0.52	310
2	0.77	0.95	0.85	541
accuracy			0.80	1436
macro avg	0.77	0.75	0.75	1436
weighted avg	0.80	0.80	0.79	1436

Figure 18 Classification report for CNN

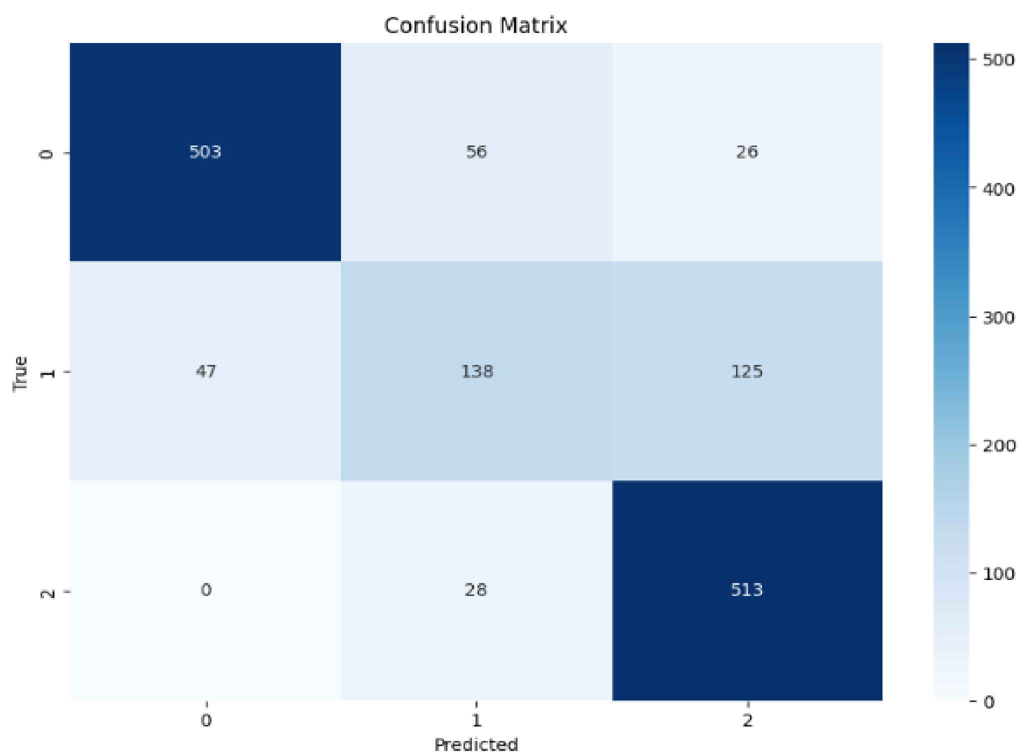


Figure 19 Confusion Matrix for CNN

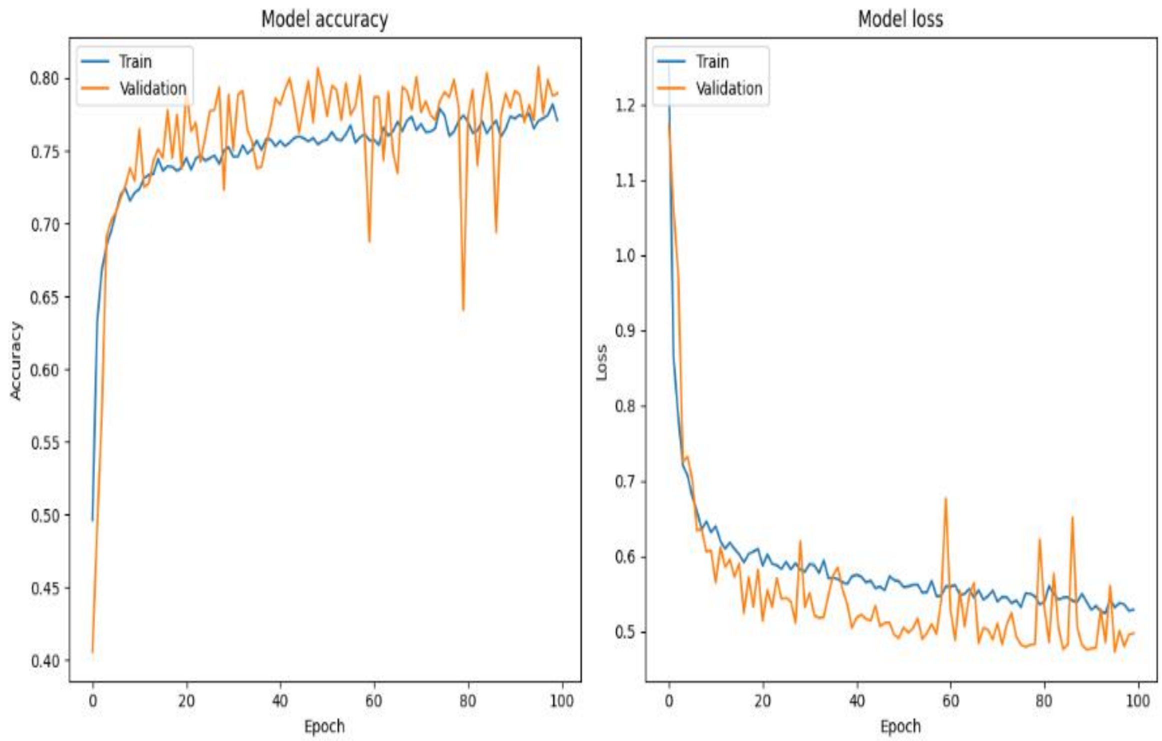


Figure 20 Accuracy vs loss curve for CNN

6.2.1.2 GRU Model

In the first experiment of our study, we employed the GRU model, a deep learning architecture commonly utilized for driver behavior classification tasks. During training, we conducted 100 epochs for each batch-size of 64, with each epoch consisting of 5743 training steps. With learning rate of 0.01 and we use sigmoid Activation functions Adam for Optimization., With the help of the GRU approach and historical data on driver behavior, the researchers were able to acquire a training score of 65%.

Table 6.4: Tune hyper parameter values of GRU

Hyper parameters	Values
Train and test split	80,20
Input layers	20,1
Hidden layer	3
Output Layer	1
Dropout	0.5
Bach size	64
Optimizer	ADAMS
Activation	relu
Epochs	100

GRU Model Classification Report:

	precision	recall	f1-score	support
0	0.76	0.72	0.74	1460
1	0.56	0.39	0.46	1461
2	0.62	0.84	0.71	1461
accuracy			0.65	4382
macro avg	0.65	0.65	0.64	4382
weighted avg	0.65	0.65	0.64	4382

Figure 21 Classification report for GRU

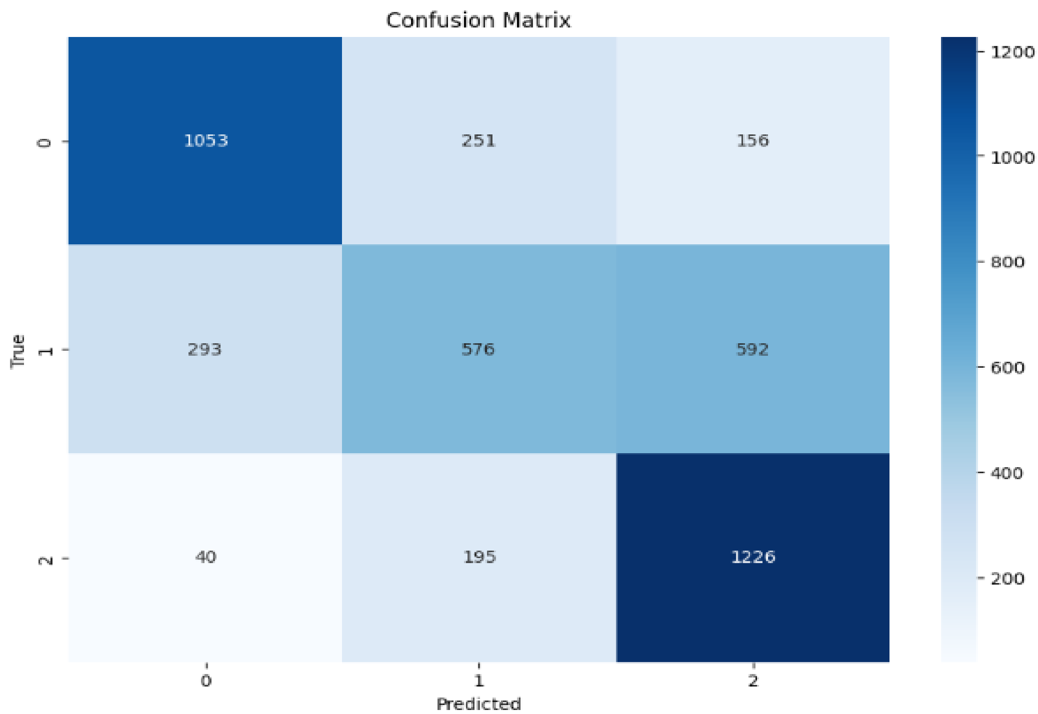


Figure 22 Confusion matrix report for GRU

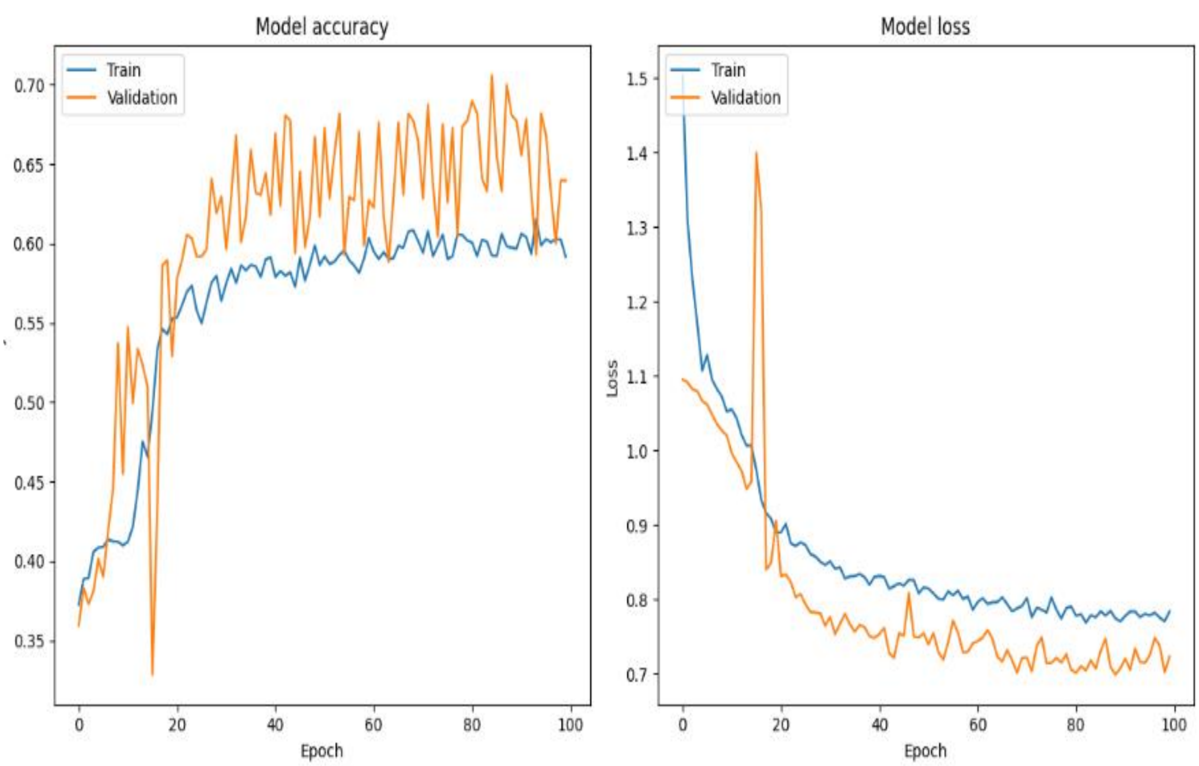


Figure 23 Accuracy vs loss curve for GRU

6.2.1.3 LSTM Model

In the Last experiment of our study, we employed the LSTM model, a deep learning architecture commonly utilized for driver behavior classification tasks. During training, we conducted 100 epochs for each batch-size of 64, with each epoch consisting of 5743 training steps. With learning rate of 0.01 and we use sigmoid Activation functions Adam for Optimization., With the help of the LSTM approach and historical data on driver behavior, the researchers were able to acquire a training score of 92%.

Table 18 Tune hyper parameter values of LSTM

Hyper parameters	Values
Train and test split	80,20
Input layers	20,1
Hidden layer	5
Output Layer	1
Dropout	0.5
Bach size	64
Optimizer	ADAMS
Activation	relu
Epochs	100

LSTM Model Classification Report:

	precision	recall	f1-score	support
0	0.98	0.96	0.97	585
1	0.83	0.77	0.80	310
2	0.90	0.94	0.92	541
accuracy			0.92	1436
macro avg	0.90	0.89	0.90	1436
weighted avg	0.92	0.92	0.91	1436

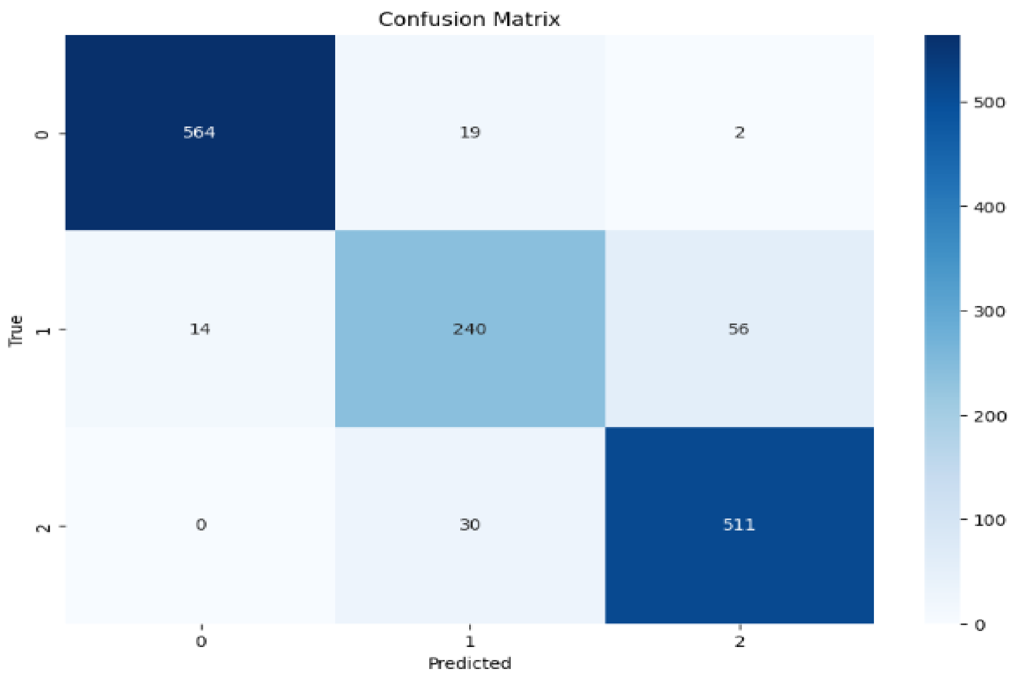


Figure 24 Confusion Matrix for LSTM

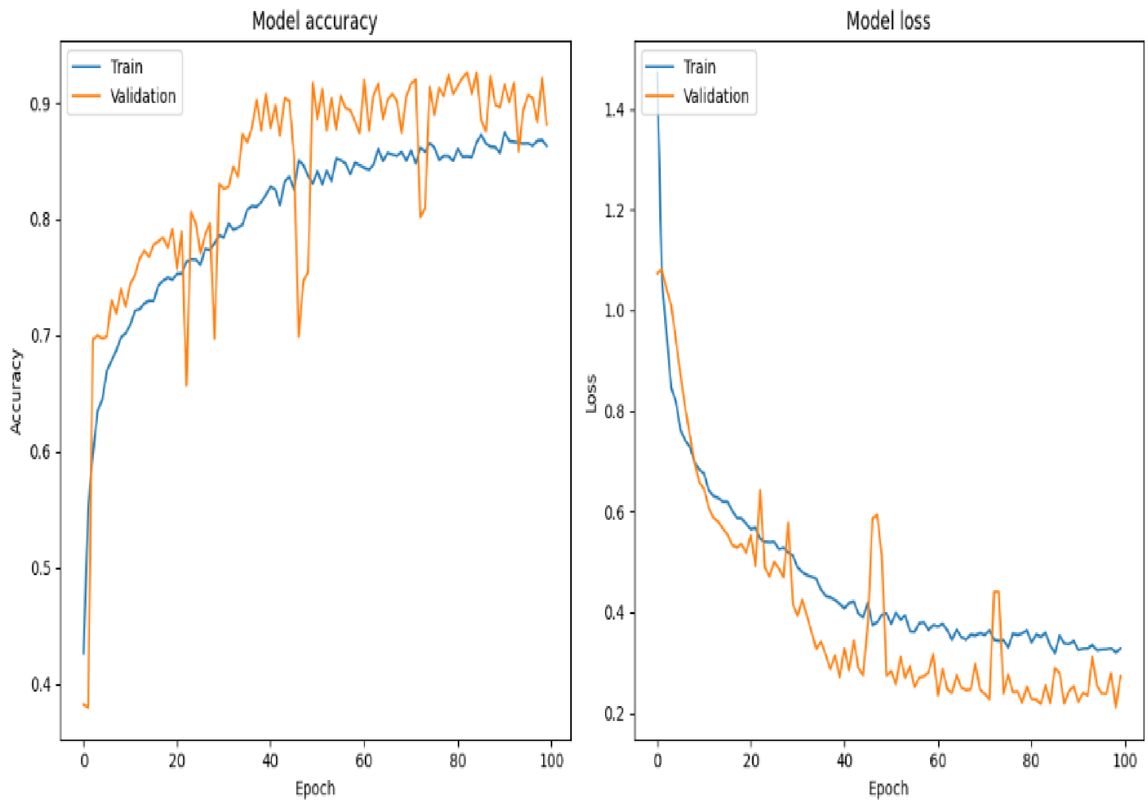


Figure 25 Accuracy vs loss curve for LSTM

Table 19 Driver behavior classification result of different deep learning models

Method	Accuracy	Precision	Recall	F1-score
CNN	80	76.6	75.3	75.3
GRU	65	76	72	74
LSTM(ours)	92	98	89	90.6

As we have seen we can say LSTM model is better performed in performance, class wise performance, and micro weighted average. The LSTM model performs than both CNN and GRU models across all metrics, showing higher precision, recall and F1-score. LSTM model also achieves the highest overall accuracy (0.92) compared to the CNN (0.80) and GRU (0.65).

Class 1, seems to be the most challenging to classify shows the lowest precision, recall, and F1-score across all models. however, the LSTM model still performs significantly better in this class compared to others. The LSTM model also has higher macro and weighted average indicating more balanced performance across all classes.

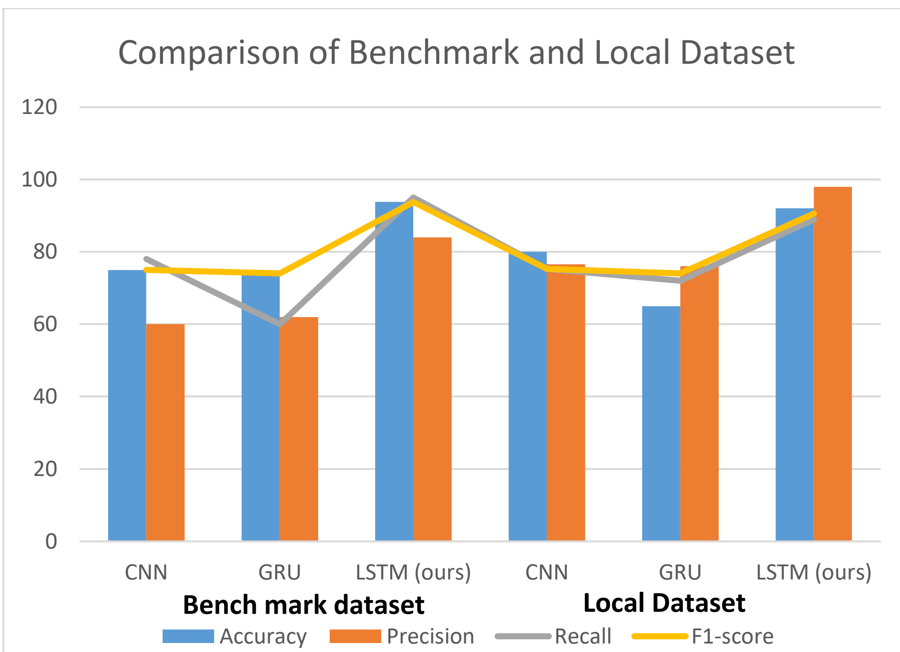
In conclusion LSTM model is most effective for the classification task, followed by CNN model, with GRU model performing the least effectively across three models.

6.3 comparison of results in local and benchmark dataset

based on below table LSTM is Consistently the best performer across both dataset, with highest accuracy, precision, recall and F-1 score. this suggest LSTMs ability to capture long term dependencies is crucial for our task (time series data driven driver behavior classification).

Table 20 comparison of results in local dataset and benchmark dataset

	Bench mark Dataset			**Local Dataset**		
Metric	**CNN**	**GRU**	**LSTM**	**CNN**	**GRU**	**LSTM**
Accuracy	75	74	93.76	80	65	92
Precision	60	62	84	76.6	76	98
Recall	78	60	95	75.3	72	89
F1-score	75	74	93.76	75.3	74	90.6



6.4 Research Question Discussion

This study answers to each of the research questions raised in the first chapter. This research provides the following answers to each question:

Answer for RQ1. What are the determining factor or attribute that influence driver behavior?

The key factors influencing driver behavior are primarily determined by several attributes, with harsh cornering being the most significant, accounting for 33.63% of the importance in predicting the target behavior. This is followed by harsh braking (19.59%) and harsh acceleration (14.23%), both of which are critical indicators of aggressive behavior patterns. Reckless driving contributes 7.35% to the overall behavior analysis, while time, or the duration of driving events, plays a smaller but still relevant role at 4.11%.in contrast, less significant factors include excessive idling (1.46%), extreme turns (1.45%), medium speeding (1.35. %), and medium braking (1.10%), which, although less impactful, still contribute to the understanding of driver behavior. Together these attributes provide comprehensive view of the elements that most strongly influence driving patterns, helping to predict and analysis driver behavior effectively.

Answer for RQ2. Which classification model is best suited to driver behavior classification?

For driver behavior classification based on sequential data GPS data, and other time-series data, proposed LSTM is generally better suited due to their ability to handle temporal dependencies effectively.

Answer for RQ3. To what extend the performance of model effective in classifying driver behavior?

this paper answered by proposed model of LSTM to what extent the model is effective in classifying driver behavior by overall accuracy of proposed model of 92%. indicates that model is generally effective in classifying driver behavior and by handling class in balance by precision, Recall and F1-score of 98%,89%, and90.6% result scored respectively provide insights into the model’s ability to correctly identify specific behaviors.

CHAPTER SEVEN

7. CONCLUSION AND FUTUER WORK

7.1 Conclusion

In conclusion, this research focused on Driver Behavior classification aimed to identify an effective model for accurately classify driver behavior. Through a series of experiments and evaluations, it has been determined that LSTM performs exceptionally well in this task.

LSTMs are particularly useful when the behavior patterns depend on long-term dependencies in the data and GRUs offer a more efficient alternative with similar performance for many applications.

The study developed a model to classify driver behavior and assessed how well the algorithms, CNN, LSTM, and GRU, performed. FOUR evaluation metrics were used to assess the model's performance.

Deep learning-based driver behavior classification has great promise for enhancing traffic safety. Deep learning models can evaluate data from in-vehicle sensors (GPS) with high accuracy. In particular, CNNs, LSTMs, These algorithms are able to categorize driver behaviors like as inattentiveness, hostility, and tiredness by successfully extracting relevant patterns from the GPS data.

7.2 Recommendations and Future Works

Based on the shortcomings of the current study, the following suggestions can be made for further investigation.

Building on the identified factors that influence driver behavior, future work can focus on several areas to enhance the understanding and prediction of driver patterns.Firstly,there is an opportunity to refine the current model by incorporating additional feature a that may capture more nuanced aspects of driver behavior, such as environmental conditions(e.g. weather, road type) and driver demographics (e.g. age, experience level).these factors could provide more holistic view and improve the accuracy of behavior predictions.

In terms of recommendations, it would be beneficial to develop real time monitoring and feedback systems that can alert drivers to potentially risk behaviors as they occur. Such systems could be integrated with in-vehicle telematics to provide immediate feedback, helping drivers correct their behavior and ultimately reduce the risk of accidents.

Finally, collaborating with policymakers and insurance companies to create incentive structures for safe driving could be explored. For instance, driver who exhibit fewer instances of harsh braking, acceleration or cornering could receive lower insurance premiums or other benefits.by integrating these approaches, future research and practical applications can lead to safer driving environments and more accurate predictions of driver behavior.

REFFERENCE

Júnior, J. F., Carvalho, E., Ferreira, B. V., De Souza, C., Suhara, Y., Pentland, A., & Pessin, G. (2017). Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLOS ONE*, *12*(4), 1–16. https://doi.org/10.1371/JOURNAL.PONE.0174959

Liu, H., Taniguchi, T., Tanaka, Y., Takenaka, K., & Bando, T. (2017). Visualization of Driving BehaviorBased on Hidden Feature Extraction by Using Deep Learning. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2477–2489. https://doi.org/10.1109/TITS.2017.2649541

Liu, Y., & Hansen, J. (2019). Analysis of Driving Performance Based on Driver Experience and Vehicle Familiarity: A UTDrive/Mobile-UTDrive App Study. *SAE International Journal of TransportationSafety*, *7*(2). https://doi.org/10.4271/09-07-02-0010

Li, Z., Zhang, K., Chen, B., Dong, Y., & Zhang, L. (2019). Driver identification in intelligent vehicle systems using machine learning algorithms. *IET Intelligent Transport Systems*, *13*(1), 40–47. https://doi.org/10.1049/IET-ITS.2017.0254

Montella, A., Pariota, L., Galante, F., Imbriani, L. L., & Mauriello, F. (2014). Prediction of drivers' speed behavior on rural motorways based on an instrumented vehicle study. *Transportation Research Record*, *2434*, 52–62. https://doi.org/10.3141/2434-07

Mohammed, A., Yazid, M. R. M., Zaidan, B. B., Zaidan, A. A., Garfan, S., Zaidan, R. A., Ameen, H. A., Kareem, Z. H., & Malik, R. Q. (2021). A Landscape of Research on Bus Driver Behavior: Taxonomy, Open Challenges, Motivations, Recommendations, Limitations, and Pathways Solution in Future. *IEEE Access*, *9*, 139896–139927. https://doi.org/10.1109/ACCESS.2021.3102222

Mukhtar, A., Xia, L., & Tang, T. B. (2015). Vehicle Detection Techniques for Collision Avoidance Systems:A Review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2318–2338. https://doi.org/10.1109/TITS.2015.2409109

Peppes, N., Alexakis, T., Adamopoulou, E., & Demestichas, K. (2021). Driving behaviour analysis using machine and deep learning methods for continuous streams of vehicular data. *Sensors*, *21*(14). https://doi.org/10.3390/s21144704

Silva, A. B., Santos, S., Vasconcelos, L., Seco, Á., & Silva, J. P. (2014). Driver behavior characterization in roundabout crossings. *Transportation Research Procedia*, *3*, 80–89. https://doi.org/10.1016/J.TRPRO.2014.10.093

Silva, I., & Naranjo, J. E. (2020). A systematic methodology to evaluate prediction models for driving style classification. *Sensors (Switzerland)*, *20*(6). https://doi.org/10.3390/S20061692

van Haperen, W., Riaz, M. S., Daniels, S., Saunier, N., Brijs, T., & Wets, G. (2019). Observing the observation of (vulnerable) road user behaviour and traffic safety: A scoping review. *Accident Analysisand Prevention*, *123*, 211–221. https://doi.org/10.1016/J.AAP.2018.11.0

Van Treese, J. W., Koeser, A. K., Fitzpatrick, G. E., Olexa, M. T., & Allen, E. J. (2017). A review of the impact of roadway vegetation on drivers' health and well-being and the risks associated with single-vehicle crashes. *Arboricultural Journal*, *39*(3), 179–193. https://doi.org/10.1080/03071375.2017.1374591

Varotto, S. F., Farah, H., Toledo, T., van Arem, B., & Hoogendoorn, S. P. (2018). Modelling decisions ofcontrol transitions and target speed regulations in full-range Adaptive Cruise Control based on RiskAllostasis Theory. *Transportation Research Part B: Methodological*, *117*, 318–341. https://doi.org/10.1016/J.TRB.2018.09.007

Wang, C., Sun, Q., Guo, Y., Fu, R., & Yuan, W. (2020). Improving the User Acceptability of Advanced Driver Assistance Systems Based on Different Driving Styles: A Case Study of Lane Change WarningSystems. *IEEE Transactions on Intelligent Transportation Systems*, *21*(10), 4196–4208. https://doi.org/10.1109/TITS.2019.2939188

Wu, B. F., Chen, Y. H., & Yeh, C. H. (2014). Driving behaviour-based event data recorder. *IET Intelligent Transport Systems*, *8*(4), 361–367. https://doi.org/10.1049/IET-ITS.2013.0009

Wu, K. F., & Jovanis, P. P. (2016). Cohort-based analysis structure for modeling driver behavior with an in-vehicle data recorder. *Transportation Research Record*, *2601*, 24–32. https://doi.org/10.3141/2601-04

Xie, G., Gao, H., Huang, B., Qian, L., & Wang, J. (2018). A driving behavior awareness model based on adynamic Bayesian network and distributed genetic algorithm. *International Journal of ComputationalIntelligence Systems*, *11*(1), 469–482. https://doi.org/10.2991/IJCIS.11.1.35

Yuksel, A. S., & Atmaca, S. (2021). Driver's black box: a system for driver risk assessment using machine learning and fuzzy logic. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, *25*(5), 482–500. https://doi.org/10.1080/15472450.2020.1852083

Yan, S., Teng, Y., Smith, J. S., & Zhang, B. (2016). Driver behavior recognition based on deep convolutional neural networks. *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016*, 636–641. https://doi.org/10.1109/FSKD.2016.7603248

Zhang, Y., Li, J., Guo, Y., Xu, C., Bao, J., & Song, Y. (2019). Vehicle Driving Behavior Recognition Basedon Multi-View Convolutional Neural Network with Joint Data Augmentation. *IEEE Transactions on Vehicular Technology*, *68*(5), 4223–4234. https://doi.org/10.1109/TVT.2019.2903110

Zhang, X., Qu, X., Tao, D., & Xue, H. (2019). The association between sensation seeking and driving outcomes: A systematic review and meta-analysis. *Accident Analysis and Prevention*, *123*, 222–234. https://doi.org/10.1016/J.AAP.2018.11.023

Zheng, Y., Chase, R. T., Elefteriadou, L., Sisiopiku, V., & Schroeder, B. (2017). Driver types and their behaviors

Within 11. ttps://doi.org/10.1080/19427867.2015.1131943.

Appendix:

NATIONAL ናሽናል
TRANSPORT ትራንስፖርት

Date:9/26/2023

To whom IT may concerns

Subject: Issuance of GPS Manual for Research Purposes

I am writing to confirm that we have processed your request for the GPS manual, which you have indicated is required for your research paper. We understand the importance of accurate and detailed information for academic and professional development purposes.

Enclosed with this letter, you will find the GPS manual. This document contains comprehensive details about the operation and functionalities of our GPS tracking systems. We hope this will be a valuable resource in aiding your research and enhancing your understanding of our technological applications.

Should you need further information or clarification on any aspects within the manual, please feel free to contact me. We are always available to assist our employees in their professional and academic endeavors.

Thank you for your commitment to enhancing your knowledge and expertise through research. We appreciate your initiative and are pleased to support your efforts.

Warm regards,

Wegayehu Aragaw
CEO
National Transport PLC
0903014422

Enclosure: GPS Manual

NATIONAL TRANSPORT PLC
ADDIS ABABA · DIRE DAWA · ADAMA

National Transport PLC

+251 115 152 902
Nationaltrasportplc@gmail.com
Addis Ababa - Diredawa - Adama

Figure 26 Collabortion Latter

National Transport PLC

GPS implementation Working Guide line

Prepared By: Wegayehu Aragaw

SEPTEMBER, 2021

Diredawa, Ethiopia

NATIONAL TRANSPORT PLC
ADDIS ABABA • DIREDAWA • ADAMA

+251 115 152 902
Nationaltrasportplc@gmail.com
Addis Ababa - Diredawa - Adama

- No driver errors or 0, no violations (harsh break, free wheel, un respect fencing routes, un respect speed limit, un necessary idling, out of green band, reckless driving etc…)
- VAT, improvement comparing same month

Based on management evaluation, feedbacks and necessary corrective actions will be taken to enhance and improve further company truck utilization.

9. Evaluating Parameters

Top and bottom drivers will be evaluated and selected based on below parameters to be motivated and incentivized depending on the findings from the GPS system.

Table 3. Fixed parameters with their respective weights

S/N	Parameters	Weight
1	Acceleration extreme	10
2	Acceleration medium	5
3	Brake extreme	10
4	Brake medium	5
5	Excessive idle	2
6	Harsh Acceleration	8
7	Harsh Braking	8
8	Harsh Cornering	8
9	Harsh driving	10
10	Speeding medium	6
11	Speeding mild	3
12	Turn extreme	7
13	Turn medium	4
14	Reckless driving	10

10. Criteria

Table 4. Selection Criteria

NATIONAL TRANSPORT PLC
ADDIS ABABA • DIRE DAWA • ADAMA
+251 115 152 902
Nationaltrasportplc@gmail.com
Addis Ababa - Diredawa - Adama
National Transport PLC

S/N	Indicator	Standard
1	Acceleration Extreme	Acceleration greater than 3.5 m/s^2
2	Acceleration Medium	Acceleration between 2.5 m/s^2 and 3.5 m/s^2
3	Brake Extreme	Deceleration greater than -4 m/s^2
4	Brake Medium	Deceleration between -3 m/s^2 and -4 m/s^2
5	Excessive Idle	Idling for more than 10 minutes
6	Harsh Acceleration	Acceleration greater than 2.5 m/s^2
7	Harsh Braking	Deceleration greater than -3 m/s^2
8	Harsh Cornering	Lateral acceleration greater than 2 m/s^2
9	Harsh Driving	Combination of harsh acceleration, braking, and cornering
10	Speeding Medium	Speeding 20-30 km/h over the speed limit
11	Speeding Mild	Speeding 10-20 km/h over the speed limit
12	Turn Extreme	Lateral acceleration greater than 3 m/s^2
13	Turn Medium	Lateral acceleration between 2 m/s^2 and 3 m/s^2
14	Reckless Driving	Multiple instances of extreme behaviors in a single trip

11. Motivation Scheme

11.1 Drivers scoring from 95-100% considered as top and subject to reward in kind.

11.2 Drivers scoring 80-94.99% considered as medium subject to written encouragement.

11.3 Drivers scoring under 80% considered as bottom subject to & capacity building training.

11. Violations that may lead to Disciplinary Actions

Below are list of driver errors /violations that lead to disciplinary actions

- Acceleration Extreme
- Acceleration Medium
- Brake Extreme
- Brake Medium
- Excessive Idle
- Harsh Acceleration
- Harsh Braking
- Harsh Cornering
- Harsh Driving
- Speeding Medium
- Speeding Mild

+251 115 152 902
Nationaltrasportplc@gmail.com
Addis Ababa - Diredawa - Adama
National Transport PLC

Figure 27 companies manual for refferences