Edge-based Algorithm Innovations for Intelligent Transportation Systems - A Safety and Efficiency Perspective

Thesis submitted in fulfilment
of the requirements for the degree of

Master of Science
in
Computer Science and Engineering
by Research

by

Goparaju Sai Usha Nagasri
2021701027
ushasai.g@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500032, INDIA
June 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled ”Edge-based Algorithm Innovations for Intelligent Transportation Systems - A Safety and Efficiency Perspective” by Usha, has been carried out under my supervision and has not submitted elsewhere for a degree.

Date

Advisor: Dr. Deepak Gangadhran

To

My Uncle

Acknowledgments

I extend my deepest gratitude to Dr. Deepak Gangadharan for his unwavering guidance and support throughout my Master's program. His belief in my abilities and his pivotal recommendation to the standing committee have deeply influenced my academic path. I am especially inspired by his calm demeanour and the reminder that "persistence is the key," which was particularly resonant when my papers faced rejection. This advice has been invaluable in overcoming my challenges. I am profoundly grateful for his mentorship and the significant role he has played in my development.

I would also like to express my profound appreciation to Dr. Bappaditya Mandal, with whom I had the privilege of collaborating. His deep insights and expert guidance have been crucial in refining my research approach and enhancing my understanding of complex concepts. The opportunity to work alongside Dr. Mandal has significantly enriched my academic experience, providing me with a broader perspective and deeper analytical skills.

Additionally, heartfelt thanks go to my collaborators Rahul, Pravalika, Bhavana, Keerthi, Shriya, and Arihanth. Each has played a vital role in various phases of our projects, bringing unique ideas and perspectives that have greatly enriched our collaborative efforts. I am also immensely grateful to my amazing lab mates Suhas, Ashish, Joseph, Sridhar, and Srikar for creating an incredibly supportive and dynamic environment. Their camaraderie, enthusiasm, and shared knowledge have not only made our lab a hub of innovation but also a place of great personal and professional growth. I must also express my deep appreciation for Ayu Parmar, who has been an incredible source of support from the first day of my time at IIIT. His encouragement during challenging times has been truly invaluable. My heartfelt thanks extend to Jayati, Spanddana, Rajshekar, Sasanka, Annesha, OM, David, Ishan as well. Their friendship and support have not only greatly enhanced my experience at IIIT but have also made those pressure-filled nights fun and bearable. Their comfort and camaraderie throughout my Master's have been a true blessing.

I owe a special thank you to my childhood friend Kavya, whose immense confidence in my potential has never wavered. This friend has stood by me through thick and thin, always sure that I would succeed and always ready to help me get there. I am also profoundly grateful to my best friends, Pranav and Lavanya, who have witnessed every facet of my personality—from the unrefined to the flawed and consistently provided encouragement. Their support has been a constant force, pushing me forward at every step of my journey.

Expressing the full extent of my gratitude towards my Mother in words is nearly impossible. As a single mother, she faced countless challenges, yet she never wavered in her support, even when I chose paths less traveled or when I took risks that seemed daunting. She has always been my biggest supporter, providing encouragement when I needed it most. Each day, I find myself inspired by her resilience and her unwavering kindness. She combines a pure heart with strong personality, qualities that make her not only my mother but my greatest role model and source of inspiration every single day. She is the epitome of strength. I strive to emulate her resilience and optimism, as she remains not just a parent but also my best friend.

I dedicate this thesis to my beloved uncle, who was a pillar of strength for our entire family and whom I sadly lost during my master's studies. His unwavering trust in my choices and his constant encouragement have shaped who I am today. He supported every endeavor I undertook and believed in me. His presence was a source of stability and inspiration, and though he is no longer with us, I carry his influence in everything I do. I am grateful for the time we shared and the lessons he taught me, and I know he would be proud of what I have accomplished. This work is a tribute to his enduring influence in my life.

Big thanks to my annoying brother, who has mastered the fine art of being annoying yet ironically remains my biggest cheerleader. It's both hilarious and heartwarming to know that the same guy who annoys me with Olympic precision would not think twice about taking on the world for me. What can I say? Sibling love has its own rules!. I would also like to express my heartfelt gratitude to my atta and dodda for their unwavering support throughout my journey. A special thanks goes to my cousins, who have consistently been there for me from childhood, providing constant support. I am particularly grateful to my cousin brother, whose guidance was pivotal during my schooling years. He taught me some of the crucial basics that significantly shaped my academic foundation. I vividly remember how, thanks to his tutoring, I jumped from rank 36 to rank 3 in my class, a milestone that greatly boosted my confidence. I would also like to pay a special tribute to my late grandfather, who we sadly lost during my master's studies. Even in his late 80s, he was passionately dedicated to acquiring knowledge, particularly in the field of Ayurveda. His commitment to learning new things at that age has deeply inspired me.

Lastly, I would like to extend my sincere gratitude to Lead Architect Anuradha Vattem. Her judgment and confidence in my abilities granted me the opportunity to work on the Smart City project, which in turn paved the way for my enrollment at IIIT. This experience has been instrumental in shaping my professional path. I would also like to thank Professor Sachin Chaudhari and Professor Aftab Hussain for their invaluable mentorship. The insights I gained from interacting with them have profoundly enriched my understanding and approach to my field of study.

Abstract

In this thesis, we tackle unique and compelling challenges within Intelligent Transportation Systems (ITS) by leveraging low power edge computing devices. alongside AI and machine learning innovations. This work delves into two important problems in ITS pertaining to efficiency of traffic flow prediction and safety in two wheeler driving.

For traffic management, we propose LSTM-based autoencoders, equipped with contextual attention mechanisms, to precisely identify and respond to anomalous traffic patterns. This approach, in analyzing vast datasets like the PeMS, not only enhances the accuracy of anomaly detection but also the efficiency of traffic flow management in urban settings.

Turning our focus to two-wheeler safety, we employ simple sensor technologies to develop models that excel in the real-time classification of driving events and fall detection. By meticulously testing various machine learning models, we've proposed time-series-based LSTM and Bi-LSTM networks for their superior accuracy in recognizing critical safety incidents. The practical deployment of these models on edge devices, such as Raspberry Pi, underscores their viability for instant safety interventions, a crucial step towards mitigating accidents before they occur.

Moreover, we developed a predictive model utilizing the Isolation Forest algorithm to anticipate fall events based on rider behavior, an innovation aiming at preemptive safety measures rather than reactive responses. This predictive capability represents a paradigm shift in how vehicular safety technologies are conceptualized and deployed, focusing on accident prevention.

Our comprehensive study illustrates the transformative potential of integrating edge computing with AI in ITS. By addressing the unique challenges of anomaly detection in both traffic management and two-wheeler safety, we contribute significantly to the advancement of intelligent transportation systems. This research not only paves the way for future innovations in vehicular safety and traffic optimization but also promises to enhance the efficiency and safety of transportation globally.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Our research embarks on a compelling journey to address the unique challenges faced by modern transportation systems, particularly focusing on traffic congestion and two-wheeler safety. These issues are not merely logistical hurdles but significant contributors to public health risks and safety concerns worldwide [1]. Statistics from various global studies paint a concerning picture, revealing staggering public health costs due to traffic-related pollution and a sharp increase in road accidents, especially involving two-wheelers[2]. These alarming trends underscore the urgent need for innovative solutions in edge-enabled artificial intelligence (AI) and machine learning. By harnessing the transformative potential of technology, our research aims to not only mitigate traffic congestion but also reduce two-wheeler accidents through real-time safety interventions. This thesis advocates the power of AI to revolutionize transportation systems, envisioning a future where roads become safer and more efficient, safeguarded by cutting-edge technology.

In our exploration, we delve into two primary domains: Anomaly Detection in traffic flow and two-wheeler safety, each posing distinct yet interconnected challenges. Traffic congestion, aggravated by urbanization and population growth, remains a pressing concern, leading to increased mortality rates linked to pollution exposure. The imperative for accurate real-time traffic flow prediction models is evident, necessitating innovative approaches to optimize traffic management and alleviate public health burdens. Simultaneously, the rise in two-wheeler accidents, particularly in developing regions, calls for tailored safety solutions to address the unique vulnerabilities of riders. Drawing insights from statistical data and national road safety reports, we endeavour to develop advanced AI-driven systems capable of real-time anomaly detection in traffic flow and fall detection/prediction, thereby enhancing road safety for two-wheelers.

By integrating edge computing with AI methodologies, our research seeks to tackle these challenges head-on, offering novel insights and solutions to enhance the intelligence of transportation systems. Through a multifaceted approach encompassing predictive modelling, anomaly detection, and proactive safety mechanisms, we aim to transform the landscape of urban mobility. Our endeavour aligns with broader public health goals, aiming to reduce pollution-related health problems, minimize transportation costs, and foster safer, more sustainable transportation networks. Through this interdisciplinary research, we aspire to pave the way for a future where transportation systems prioritize safety, efficiency, and environmental sustainability.

1.1 Motivation

1.1.1 Vulnerabilities and Risks Associated with Two-Wheelers

In a comprehensive examination of the critical issues plaguing modern transportation—traffic congestion and two-wheeler safety—our research employs a dual-faceted approach grounded in edge-enabled artificial intelligence (AI) and machine learning [3]. This study is motivated by the significant public health and safety implications of these issues, underscored by statistical analyses from various global studies. In countries like India, the narrative is equally grim, with over 461,312 road accidents recorded in a single year, marking an alarming increase and underscoring a crisis in road safety, particularly for two-wheeler users who face a disproportionately high risk of accidents and fatalities [4, 5].

These statistics serve as a wake-up call, prompting innovative solutions employing ML/DL algorithms on edge devices. The proposed development of a deep learning-based, time-series fall detection system signifies a pioneering step towards mitigating these risks. By harnessing comprehensive datasets to train the system in distinguishing between normal and fall scenarios, we aim not only to advance the precision of fall detection but also to expedite emergency responses, thereby potentially reducing the severity of outcomes following accidents.

The two-wheeler users are directly exposed and come in direct contact with the impacting vehicle or obstacle during a collision resulting in severe injuries and fatality [6]. In India, two-wheeler accidents have also been shown to have maximum case fatality in accidents [7]. Three factors impacting the cause of road traffic accidents are human, vehicle and road conditions. The human factor is the most difficult to understand, model and predict out of all these three factors. The other factors are more predictable comparatively. Also, human behavior is something that can be altered and acted upon in terms of accident prevention and mitigation. Therefore, understanding human behavior while driving becomes very crucial.

Firstly, understanding driving behavior becomes vital for safety of other commuters as it can cause high risk to them on the road. Secondly, the safety point of view of the rider, feedback and driving assistance systems are essential for improving individual driving behavior and creating awareness regarding the impacts of the way they drive. Drivers differ in the way they choose to accelerate and decelerate, the distance they keep from the leading vehicle, adherence to speed limits, and use body weight or vehicle handle movement while taking turns. Unlike four-wheelers, identifying driving patterns of two-wheelers is even more difficult due to their highly transient dynamics during operation, which is heavily dependent on the riding conditions and rider behaviors. This can help in detecting events that can potentially result in accidents. However, in order to detect an accident scenario, the first step is to recognize driving events, a combination of which could lead to accident scenarios.

1.1.2 Traffic Congestion as an ITS Challenge

The domain of intelligent traffic management and the identification of anomalies in traffic flow data stands as a pivotal concern. Such anomalies, defined as notable deviations from standard traffic patterns, often manifest as univariate time series fluctuations [8]. The criticality of accurate and prompt anomaly detection cannot be overstated; it forms the backbone of proactive incident response mechanisms that significantly enhance the efficacy of traffic management systems. This introductory discourse aims to elucidate the indispensable role of anomaly detection within the broader context of intelligent traffic management, underpinning its necessity through statistical insights and the manifold advantages it presents [9, 10]. The critical role of anomaly detection in traffic management is fundamentally linked to its capacity to enhance road safety and improve traffic flow. Anomalies, such as sudden stops or unexpected congestion, act as early indicators of possible road incidents or dangerous conditions. Employing anomaly detection proactively enables faster response to emergencies, potentially reducing the severity of accidents and saving lives [11]. Additionally, anomaly detection plays a vital role in making traffic flow smoother. It helps in identifying the unusual patterns of traffic, like unnecessary congestion or interruptions, allowing traffic control systems to make timely adjustments [9]. These adjustments might include changing traffic signals or suggesting different routes to prevent bottlenecks. Such measures are essential when considering the U.S. Department of Transportation's report, which states that traffic congestion leads to over $100 billion annually in lost time and wasted fuel for Americans [12]. This information highlights the importance of using anomaly detection not just for safety but also for efficiency on the roads.

The environmental impact of implementing advanced anomaly detection in traffic management extends far beyond improving safety and efficiency. By enabling more efficient traffic control, anomaly detection leads to smoother traffic flows and fewer instances of vehicles idling. This reduction in idle times significantly decreases vehicular emissions [9]. The Environmental Protection Agency (EPA) has highlighted that the transportation sector is the primary source of greenhouse gas emissions in the United States, responsible for almost 29% of the nation's total emissions [9]. Therefore, improvements in anomaly detection technology not only aim to decrease the delays caused by congestion but also contribute to creating a more environmentally friendly and sustainable urban ecosystem [9].

The economic advantages of utilizing efficient traffic anomaly detection systems are significant, highlighting the importance of such technologies in modern traffic management. By reducing congestion, these systems not only save time for commuters but also lead to substantial cost savings for businesses that depend on road transportation for their logistics and delivery operations. The Federal Highway Administration (FHWA) has noted that enhancements in traffic management can result in savings of billions of dollars by minimizing delays and reducing fuel consumption, thus supporting the broader economy [10]. Additionally, anomaly detection is pivotal for the development of smart cities, which aim to improve the quality of urban life and sustainability. By incorporating real-time traffic data analysis, cities can improve road safety and traffic flow while also working towards wider goals like lowering greenhouse gas emissions, improving public transit systems, and promoting economic growth.

The complexity of traffic flow patterns, influenced by several factors like road conditions, daily commuting patterns, and unforeseen incidents, necessitates robust and adaptive models for anomaly detection [13]. Traditionally, a range of methodologies have been employed to tackle this challenge, as outlined in a comprehensive survey by Braei and Wagner (2020) [14]. These methodologies span statistical models, such as ARIMA, classical machine learning techniques like K-Means Clustering, and One-Class SVM, to cutting-edge deep learning methods [15]. The latter, notably, attempts to unravel the intricate, nonlinear correlations in data to predict future traffic patterns and identify anomalies based on deviations from these predictions [11]. In the process of drafting this thesis, the author utilized ChatGPT, a large language model developed by OpenAI, for rephrasing and refining the language of the initial drafts. This tool proved invaluable in enhancing the clarity and readability of the text. [OpenAI. (2024). ChatGPT. Large language model. /g/g-B3hgivKK9-write-for-me].

1.2 Summary of Contributions

The main contributions from this thesis are presented in the chapters mentioned as follows -

- Driving Event Recognition: Development of machine learning models, including LSTM and Bi-LSTM networks, for the precise classification of various two-wheeler driving events. This contribution is pivotal for understanding rider behavior and identifying potentially hazardous maneuvers, enhancing the overall safety system for two-wheelers.

- Fall Detection System: Introduction of a time-series-based deep learning system capable of real-time fall detection. Utilizing advanced algorithms, this system accurately identifies instances of falls, significantly improving emergency response times and potentially reducing the severity of injuries.

- Fall Prediction Mechanism: Implementation of a predictive model that leverages data on rider behavior and environmental conditions to forecast potential fall events. This proactive approach aims to alert riders to imminent risks, offering a novel strategy to prevent accidents before they occur, thus advancing the safety features available to two-wheeler users.

- Development of LSTM-based autoencoders with contextual attention mechanisms for anomaly detection in traffic management, facilitating dynamic and efficient traffic control strategies.

- Edge Computing Integration: Deployment of the proposed models on edge devices, such as Raspberry Pi, showcasing the feasibility of real-time, on-device processing for traffic prediction and two-wheeler safety applications.

This research addresses traffic congestion and two-wheeler accidents, contributing to reduced pollution-related health problems and fatalities in two-wheelers. It improves traffic flow and enhances two-wheeler safety, leading to decreased transportation time, and costs and increased safety.

The thesis is organized as follows: Chapter 3 of this thesis discusses the application of time series analysis for detecting and recognizing driving events in two-wheeled vehicles, detailing the exploration of relevant datasets, the development of models, and the deployment of evaluation metrics to assess their efficacy. Following this, Chapter 4 focuses on the refinement and implementation of time series analysis techniques specifically designed to improve fall detection and prediction systems within two-wheeled transportation. This includes a thorough examination of methodologies, addressing challenges, and discussing the outcomes of these advanced analytical approaches. In Chapter 5, we introduce a novel approach for anomaly detection in traffic flow analysis using a contextual attention-based encoder-decoder network. The chapter details the architecture of this model, the training process, and the utilization of large datasets, demonstrating the model's effectiveness through various case studies that underscore its capacity to enhance traffic management systems. Chapter 6 extends the discussion to additional research that contributes to the broader goals of the thesis, including partnerships and other projects that help advance the field of intelligent transportation systems. The thesis concludes with Chapter 7, which summarizes the key findings and contributions of the research. This final chapter discusses the implications of the work for the development of intelligent transportation systems and outlines potential future research directions that could continue to advance the field, emphasizing the importance of ongoing innovation in traffic safety and management.

Chapter 2

Related Works

The Related Works section provides an overview of the existing research and methodologies relevant to the study. It begins by exploring anomaly detection in time series data, categorizing approaches into statistics-based, prediction-based, and reconstruction-based methods. Each approach is discussed, highlighting its strengths and applications in anomaly detection for various contexts. The section then delves into driver behavior studies for two-wheelers, emphasizing the importance of understanding rider patterns and mitigating potential hazards. Traditional machine learning models and deep learning techniques, particularly LSTM networks, are discussed in the context of recognizing riding patterns and enhancing safety measures.Additionally, the section addresses pre-impact fall detection, showcasing studies that aim to detect falls before they occur. Techniques such as SVMs, HMMs, and advanced machine learning algorithms are employed to develop proactive safety measures for two-wheeler riders.

2.1 Driver Behavior Studies for Two-Wheelers

In the field of two-wheeler safety, various research frameworks have been developed using traditional machine learning models to enhance the prediction and recognition of rider patterns and potential hazards ([16], [17], [18]). Previous studies have utilized data from accelerometers, gyroscopes, and GPS to recognize patterns. These systems, while foundational, often do not fully account for the dynamic kinematic state changes in moving vehicles.

Understanding driver behavior is crucial for enhancing safety measures, particularly for vulnerable road users like two-wheeler riders. Research in this domain encompasses both traditional machine learning and deep learning techniques, aiming to recognize riding patterns and mitigate potential hazards.

Initial frameworks utilize classical machine learning models and sensor data to recognize riding patterns [19]. For instance, Mitrovic proposed a system based on accelerometers, gyroscopes, and GPS data to recognize patterns using Hidden Markov Models (HMMs) [20]. However, these approaches may not fully capture the dynamic transitions inherent in moving vehicles. Deep learning techniques, particularly LSTM networks, have emerged as powerful tools for time-series classification in driver behavior studies. For example, Schalk Wilhelm Pienaar [21] proposed an LSTM-RNN Deep Neural Network

Architecture for human activity recognition, demonstrating the effectiveness of LSTM networks in processing and classifying sequential data. Advancements in collision and hazard detection complement traditional approaches, showcasing innovations in proactive safety measures [22, 23]. These studies underscore the potential of advanced machine learning techniques in enhancing safety measures for two-wheeler riders.

2.2 Pre-Impact Fall Detection

In the domain of *Pre-impact Fall* detection, several studies have laid the groundwork for human fall detection, offering valuable insights that can be adapted for two-wheeler fall scenarios. Notably, methods employing Support Vector Machines (SVM) and Hidden Markov Models (HMM) ([24],[25]) have shown promise. Furthermore, innovative approaches towards enhancing rider safety, including the development of airbag systems for two-wheelers informed by Long Short-Term Memory (LSTM) networks to accurately predict the optimal timing for deploying wearable bike airbags during accidents, have been explored [26]. This emerging research underscores the potential of leveraging advanced machine-learning techniques to improve proactive safety measures for two-wheeler riders.

Pre-impact fall detection studies offer valuable insights applicable to two-wheeler safety, aiming to detect falls before they occur. These studies employ a variety of techniques, including Support Vector Machines (SVMs), Hidden Markov Models (HMMs), and innovative approaches like airbag systems informed by LSTM networks [26]. For example, previous works have utilized SVMs and HMMs for pre-impact fall detection, showcasing promising results [24, 25]. Additionally, studies like [26] propose thresholding-based crash detection algorithms for two-wheelers.

2.3 Anomaly Detection in Time Series Data

Anomaly detection in time series data stands as a critical research domain, addressing the unique challenges posed by dynamic, sequential data. Methodologies have evolved to encompass diverse approaches, each offering insights into anomaly detection tailored for specific contexts.

2.3.1 Statistics-based Anomaly Detection

Historical approaches to anomaly detection have predominantly relied on statistical models, underpinned by the assumption that data conform to specific statistical patterns. This assumption enables the identification of anomalies when new data points significantly deviate from these established patterns [27]. Classical models in this domain include hypothesis testing [28], wavelet analysis [29], and ARIMA [30], each offering a unique perspective on anomaly detection. Notably, Yamanishi et al. [31] leveraged statistical learning theory in an online learning algorithm tailored for anomaly detection. More recent advancements, like the application of extreme value theory by Siffer et al. [32], have refined these techniques for univariate time series, though their focus has predominantly been on anomalies exceeding normal levels, leaving sub-normal anomalies less explored.

2.3.2 Prediction-based Methods

With the advent of more complex data structures and the need for dynamic analysis, prediction-based methods have gained traction in time series anomaly detection. These methods hinge on forecasting subsequent values in a series, flagging deviations from these forecasts as potential anomalies [33]. The advent of deep learning has notably enhanced the efficacy of prediction-based methods. For instance, Buda et al. [34] utilized LSTM models to achieve precise forecasting capabilities, and Hundman et al. [35] applied unsupervised LSTM models for anomaly detection in spacecraft telemetry data. Another innovative approach, DeepAnT, introduced by Munir et al. [36], effectively combined CNNs for time series prediction. However, despite their success in short-term forecasting, these models often falter in rapidly changing environments, such as those encountered in financial markets, underscoring the need for more adaptive and responsive methodologies.

2.3.3 Reconstruction-based Anomaly Detection

Reconstruction-based anomaly detection approaches offer a paradigm shift from prediction-based methods. These models encode standard sequences into latent spaces and detect anomalies through discrepancies observed during the reconstruction phase in test data. By training exclusively on normal data, reconstruction-based models enhance sensitivity and accuracy in anomaly detection, particularly in semi-supervised learning scenarios. Studies like Darban et al. [37] have demonstrated the effectiveness of this approach in capturing anomalies in time series data.

Chapter 3

Analyzing and Recognizing Driving Events in Two-Wheeled Vehicles Through Time Series Data

Classification of a motorcycle's driving events can provide deep insights to detect issues related to driver safety. Safety in two wheelers is a less studied problem, and we are attempting to address this gap by providing a learning based solution to classify driving events. Firstly, we developed a hardware system with 3-D accelerometer/gyroscope sensors that can be deployed on a motorcycle. The data obtained from these sensors is used to identify various driving events. We have investigated several machine learning (ML) models to classify the driving events. However, in this process, we identified that though the overall accuracy of these traditional ML models is decent enough, the class-wise accuracy of these models is poor. Hence, we have developed time-series-based classification algorithms using LSTM and Bi-LSTM to classify various driving events. We have also embedded an attention mechanism in the architecture of these models for enhanced feature learning, thus improving the accuracy of event recognition. The experiments conducted have demonstrated that the proposed models have surpassed the state-of-the-art models in the context of driving event recognition with reasonable class-wise accuracies. We have also deployed these models on the edge devices like Raspberrypi and successfully reproduced the prediction accuracies in the devices. The experiments demonstrated that the proposed Bi-LSTM model showed a minimum of 88% accuracy and a maximum of 99% accuracy in class-wise prediction on a 2-wheeler driving dataset.

3.1 Introduction

Transportation has become one of the basic needs of human life. Driving patterns on roads in cities of developing countries are very different from those in cities in developed countries. In India, two-wheelers are popularly used [38] for reasons such as 1) Best mobility solution for 1 or 2 people, 2) Requires less parking space, 3) Easy maneuvering through traffic, and 4) Low purchase and running cost. With the increasing urbanization, the two-wheeler users are increasing in numbers, resulting in a greater risk of accidents. The alarming increase in mortality owing to road traffic accidents has been a matter of great concern globally. A study [2] was undertaken to find the trend of two-wheeler accidents over the five years (2000-2004) with respect to age and sex of the victim, type of injury sustained, type of vehicle involved and time distribution of accidents. Every day as many as 1,40,000 people are injured on roads across the world, of which more than 3000 die and around 15,000 are disabled for life [1].

The two-wheeler users are directly exposed and come in direct contact with the impacting vehicle or obstacle during a collision resulting in severe injuries and fatality [6]. In India, two-wheeler accidents have also been shown to have maximum case fatality in accidents [7]. Three factors impacting the cause of road traffic accidents are human, vehicle and road conditions. The human factor is the most difficult to understand, model and predict out of all these three factors. The other factors are more predictable comparatively. Also, human behavior is something that can be altered and acted upon in terms of accident prevention and mitigation. Therefore, understanding human behavior while driving becomes very crucial.

3.1.1 Motivation

Driving behavior plays an important role in maintaining safety on roads [39]. It also affects traffic flow, fuel consumption, air pollution, public health, personal mental health and psychology. A quarter of serious or fatal injuries to bike riders occurred in accidents where the rider lost control with no other vehicle involvement [1].

Firstly, understanding driving behavior becomes vital for safety of other commuters as it can cause high risk to them on the road. Secondly, the safety point of view of the rider, feedback and driving assistance systems are essential for improving individual driving behavior and creating awareness regarding the impacts of the way they drive. Drivers differ in the way they choose to accelerate and decelerate, the distance they keep from the leading vehicle, adherence to speed limits, and use body weight or vehicle handle movement while taking turns. Unlike four-wheelers, identifying driving patterns of two-wheelers is even more difficult due to their highly transient dynamics during operation, which is heavily dependent on the riding conditions and rider behaviors. This can help in detecting events that can potentially result in accidents. However, in order to detect an accident scenario, the first step is to recognize driving events, a combination of which could lead to accident scenarios.

To the best of our knowledge, there has been very little work (mainly using traditional machine learning models, which is discussed in Section 4.1.3) on developing a system for automatic recognition of bike driving events. Hence, we attempt to fill this gap systematically by investigating various approaches and propose novel time-series based deep learning (DL) models for driving event recognition. Therefore our contributions in this space are as follows:

1. Due to the unavailability of two-wheeler driving data, we have developed the hardware system and deployed it on the two-wheeler.

2. We have collected the driving data using this system to understand the driver's behavior.

3. We have compared various traditional machine learning algorithms using the data acquired.
4. We propose time-series based DL models for driving event recognition and demonstrate its superiority over traditional machine learning models in terms of accuracy.
5. We deployed the time-series based DL models on Raspberry Pi platform and demonstrated their trade-offs between accuracy, memory usage and inference time.

3.2 Related work

Driver behavior is the primary cause of two-wheeler accidents. There have been works on studying driving event recognition in the case of four-wheelers using classical and machine learning approaches. In this context, there are various frameworks [16], [17], [18] that use unsupervised, semi-supervised and supervised models for the multi-class classification of driving maneuvers and also identifying the specific types of abnormal driving behaviors from sensor fusion data of four-wheelers. The few works on driving behavior studies for two-wheelers are presented next.

3.2.1 Driver behavior studies for two-wheelers

There are some frameworks developed using traditional machine learning models for two-wheelers. Mitrovic proposed a simple system based on accelerometers, gyroscopes, and GPS data to recognize patterns using HMMs [20]. Ferreira et al. [40] compared the performance of Multi-Layer Perceptron (MLP), Support Vector Machines, Random Forest, Bayesian Networks in the classification of driving maneuvers from smartphone sensors. In [19], a machine learning framework was proposed to identify the class of riding patterns using data collected from 3-D accelerometer/gyroscope sensors mounted on motorcycles. Additionally, they also proposed an approach for sensor selection to identify the significant measurements for improved riding pattern recognition. But this work does not capture the kinematic state change of moving vehicles. Hence, to capture those dynamic transitions, we have proposed a time-series based classification models for two-wheelers. In [41], the authors adopted a Machine Learning based movement identification process with an Artificial Neural Network (ANN) algorithm.

There are some studies based on deep learning as well in the context of time-series classification in general. LSTMs are proven to excel in learning, processing and classifying such types of data. Schalk Wilhelm Pienaar [21] proposed an LSTM-RNN Deep Neural Network Architecture for human activity recognition signifying the importance of usage of RNN for time-series data. A prior work [22] deals with collision and hazard detection for motorcycles. This is usually done by setting absolute thresholds on the accelerometer measurements, which is not intuitive. Hence the authors have proposed a method based on self-organized neural networks that can deal with a large number of inputs from different types of sensors. In [26], the authors have proposed an airbag system using LSTM to decide on the deployment of a wearable bike airbag in case of an accident.

The Journal is further organized into the following sections. In Section III, we explain our approach to hardware design, data collection strategy and proposed model. Section IV provides a detailed experimental procedure, data pre-processing, training, and testing of the model. Section V consists of the conclusion and future scope.

3.3 Methodology

In this work, we have implemented the end-to-end framework for recognizing various driving events. The overall problem can be addressed in four phases, as shown in the Fig. 3.2. Phase 1 (shown as Phase-1) comprises of design and development of the hardware followed by data collection. In the 2nd phase (shown as Phase-2), the dataset is labelled and pre-processing is performed before feeding it as input to the model. In this work, the proposed time-series based classification models have been developed and trained in Phase-3. In the 4th phase (shown as Phase-4), the models are optimized and deployed on edge devices like Raspberry Pi. The deployed models have been analyzed w.r.t. memory usage and inference time. The different phases presented in Fig. 3.2 have been discussed further in subsequent sections.

The proposed system consists of a microcontroller, accelerometer, gyroscope, GPS module and an SD card module. The sensors are interfaced using an Arduino Nano 33 IoT controller board as shown in Fig. 3.1. A brief explanation of the functions of each component of the hardware is presented below.

3.3.1 Microcontroller

Nano 33 IoT is a miniature sized module containing an Arm Cortex M0+ based ATSAMD21 processor and a u-blox NINA-W102 WiFi+BlueTooth module. It has 256 kB SRAM and 48 MHz clock speed, with a flash memory of 1 Mbites, which makes it ideal for running machine learning algorithms.

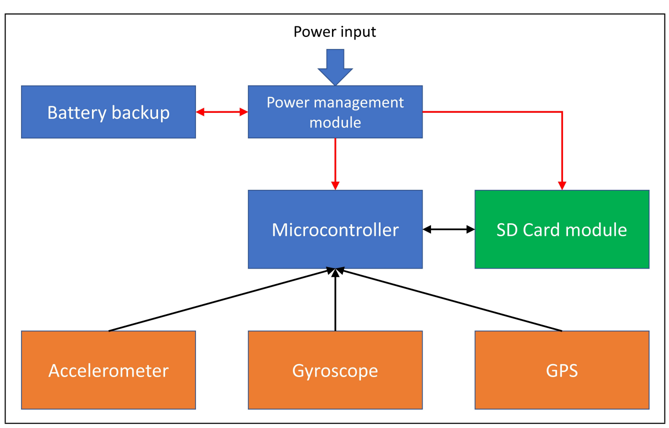


Figure 3.1: Block diagram of the fabricated hardware system

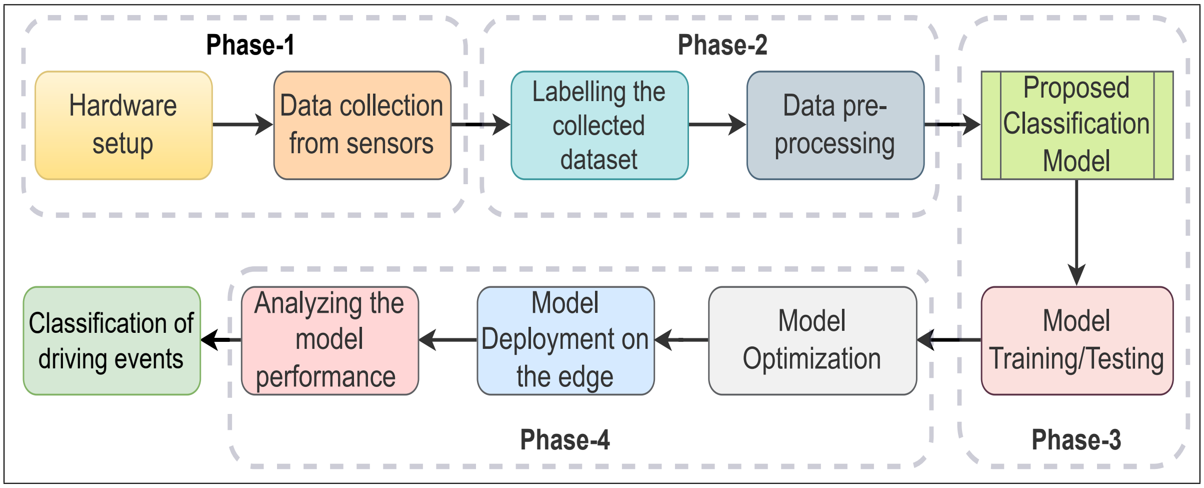


Figure 3.2: Overall block diagram of the proposed Driving Event Recognition.

3.3.2 Inertial measurement unit (IMU)

The system included an LSM6DS3 IMU sensor that provides three-axis linear acceleration and angular velocity information. The acceleration is reported in terms of gs (multiples of acceleration due to gravity), whereas angular velocity is reported in terms of degrees per second (dps). The IMU used in this work can provide a full-scale range of measurement of $\pm 16\ g$ linear acceleration and ± 2000 dps angular velocity.

3.3.3 GPS module

The NEO-M8 module utilizes concurrent reception of up to three GNSS systems (GPS/Galileo together with BeiDou or GLONASS), and recognizes multiple constellations simultaneously to provide outstanding positioning accuracy in scenarios where urban canyon or weak signals are involved. The position information is obtained in the form of an NMEA sentence is transmitted to the processor using UART. The information obtained from the sensor also includes the global timestamp and surface velocity of the module.

3.3.4 SD card module

The module (Micro-SD Card Adapter) is a Micro SD card reader/writer module that communicates with the microcontroller using an SPI interface. The module is used to write information collected from all the sensors into a microSD card for analysis later.

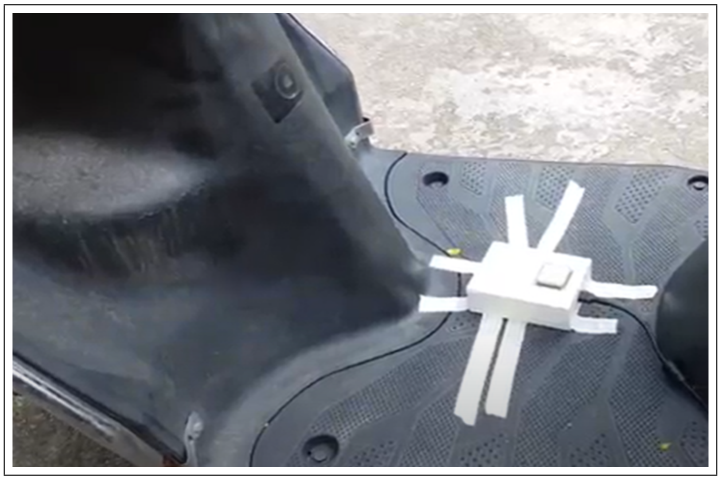


Figure 3.3: Placement of the hardware system on the motorcycle.

3.4 Data collection

The data is collected for various driving events from the hardware system deployed on the two-wheeler. The system is placed near the foot rest area of the two-wheeler as shown in Fig. 3.3. The data from the accelerometer and gyroscope sensors are continuously logged in the SD card. Simultaneously, the entire ride is recorded with the help of a camera. Later, by matching the timestamps between the sensor data and the video, the dataset is labeled manually. The dataset consists of 7 features, namely Ax, Ay, Az (accelerometer data on the x, y and z-axis), Gx, Gy, Gz (Gyroscope data on the x, y and z-axis) and speed. The sampling rate of the sensors is 104 Hz. We have performed five driving events, such as Left Turn (LT), Right Turn (RT), Straight Line (SL), Speed Bump (SB), and Stop (ST). The two-wheeler used for collecting the data is Honda Activa. The route selected for data selection covers all the events and is relatively safe to collect the data.

3.5 Classification using traditional machine learning models

As a preliminary part of this work, we implemented various ML models to understand the capabilities of each of these models with respect to classifying various driving events. Hence, we implemented various ML algorithms that have been used in prior works, such as K-Nearest Neighbor (KNN), Support Vector classifier (SVC), Decision Trees (DT), Random Forest (RF) and Naive Bayes (NB), which gave accuracies of 88.46%, 41.02%, 87.17%, 89.74% and 84.61%, respectively.

Among the ML models, it is observed that the RF model has the highest accuracy. Though the overall accuracy of these models are decent, the class-wise accuracy (accuracy to classify a specific driving event) is pretty low, especially in the case of events like ‘Left Turn’ and ‘Bump’. The main reason is that the vehicle is in motion due to which there will be some changes in kinematic states such as acceleration, deceleration, angular velocity, etc., of the vehicle while performing certain events like turns, braking, etc. ML models are not capable of capturing these transitions. On the other hand, neural network models can be trained to capture these transitions. Hence, we propose time-series based classification models to classify various driving events.

3.6 Proposed DL based time-series classification Models

Time series classification data differs from a typical classification problem since the attributes have an ordered sequence. In this work, firstly, we test the efficiency of the time series-based DL models on driving event recognition, which has not been done earlier for two-wheelers. The rationale behind using time series-based DL models is that the driving events have a dependency on the immediate historical data pattern. Therefore, we implement LSTM and Bi-LSTM, two widely popular time series-based DL models.

3.6.1 LSTM

In addition to the rationale mentioned earlier, the reason for choosing LSTM [42] to perform driving event classification is its ability to learn from the raw time series data directly thereby eliminating the need to manually engineer input features. LSTM is an efficient recurrent neural network that can hold information from the time series data for a longer duration of time. It can be used to model sequential data and is hence used to learn complex human behavior while riding two-wheelers.

3.6.2 Bi-LSTM

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. The input sequence given to the network consists of the seven features (Ax, Ay, Az, Gx, Gy, Gz and speed) of the dataset. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTM's on the input sequence [43]. The first LSTM traverses on the input sequence in the given order, whereas the second one on the reversed copy of the input sequence. This can provide additional context of the driving event to the network and result in faster and even fuller learning on the problem.

3.7 Attention mechanism for the proposed models

The attention mechanism [44] emerged innately from problems that deal with time-varying data (sequences). The main objective of the attention mechanism is to filter the critical representations out for the purpose of recognition. An attention mechanism is used to redistribute the weights of representations. It can highlight the vital information from the contextual information by setting different weights. Our attention function is straightforward; it takes the dot product of weights and inputs followed by the addition of bias terms. After that, we add a *tanh* followed by a *softmax* layer. In time-series problems, all elements of the sequence generally contribute equally to the result, but in reality, this may not be the case. For example, a sudden change in acceleration along one direction could be a better indicator of a particular driving event. Hence it is critical to capture those particular features contributing to recognizing a particular event. Hence, we have enhanced the LSTM and Bi-LSTM models by paying attention to specific features that have more impact in recognizing a particular event by embedding an attention layer.

3.7.1 LSTM with attention mechanism

LSTM cells can't understand long terms dependencies from arbitrary lengths. Therefore, their performance degrades as the sequence length increases. As the name suggests, attention furnishes a mechanism where output can *attend to* a particular input time step for an input sequence of arbitrary length. Hence, an attention layer is embedded on the LSTM layer. The simple LSTM model cannot capture these critical features.

3.7.2 Bi-LSTM with attention mechanism

An attention mechanism is utilized to focus on the information fed out from the hidden layers of Bi-LSTM. The simple Bi-LSTM structure allows the networks to have both backward and forward information about the sequence at every time step. In this work, we have added an attention layer over the Bi-LSTM layer for enhanced feature extraction. The output of the attention layer is given to the dense layers.

3.8 Experimental Results and Discussions

3.8.1 Experimental setup

The aim of this work is to develop a system and efficient DL models to execute within the system. which can perform better driving event classification compared to existing models. The training has been performed on a workstation with an Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz and one NVIDIA GeForce GTX 1650 TI Graphics Card. We have used Jupyter notebook to perform the experiments. The framework used in our work to build various models is TensorFlow-Keras. In order to evaluate the models, we have used the 'accuracy' which is the most commonly used evaluation metric as denoted in Eq. (3.1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

where true positives (TP) and true negatives (TN) denote the correct classifications of positive and negative examples, respectively. False positives (FP) represent the incorrect classification of negative

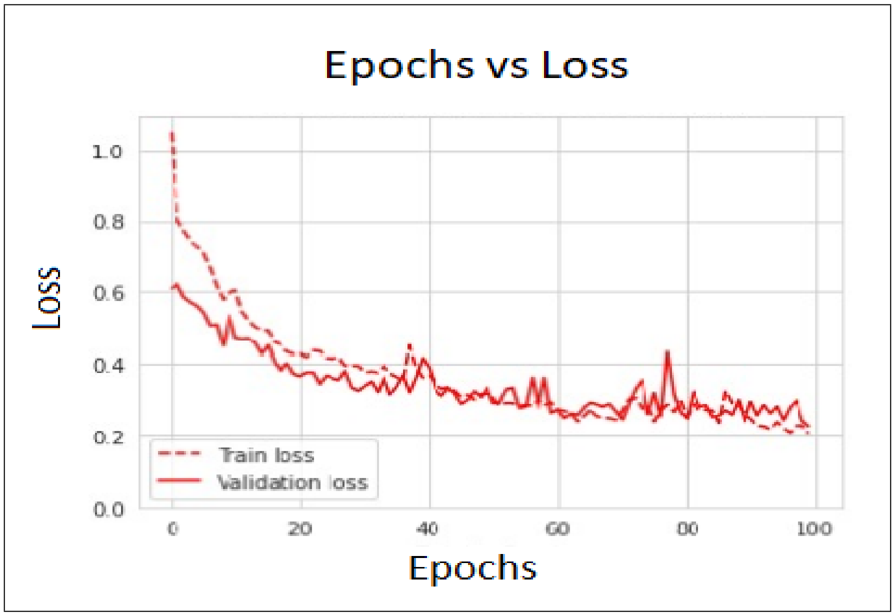


Figure 3.4: Epochs vs Loss

examples into the positive class, and false negatives (FN) are positive examples incorrectly classified into the negative class.

3.8.2 Data pre-processing

A pre-processing step is essential to replace the missing values to ensure the continuity of data and synchronization with the video. The database contains a total of 2,22,857 data points which contain 1,42,138 SL instances, 9729 RT instances, 5937 LT instances, 13163 SB instances and 7316 ST instances. The dataset is normalized to minimize redundancy and improve the integrity of database. The dataset is divided into training and test set consisting of 80% and 20% of the original data respectively.

The variation of accuracy and loss over the number of epochs for the training and validation dataset for the LSTM model is demonstrated by Fig. 3.4. Initially, the validation accuracy increases, then slows down. After 80 epochs, the accuracy and loss values become stable. At 100 epochs, the model has converged.

3.8.3 Window size vs Accuracy

Window size is one of the crucial parameters that impact the accuracy of the models. The variation of window size affects the training process, which results in the variation of the model accuracy. For the data set we have collected, the events were better detected with some window sizes as the appropriate amount of information for event detection was present in that window. After performing several

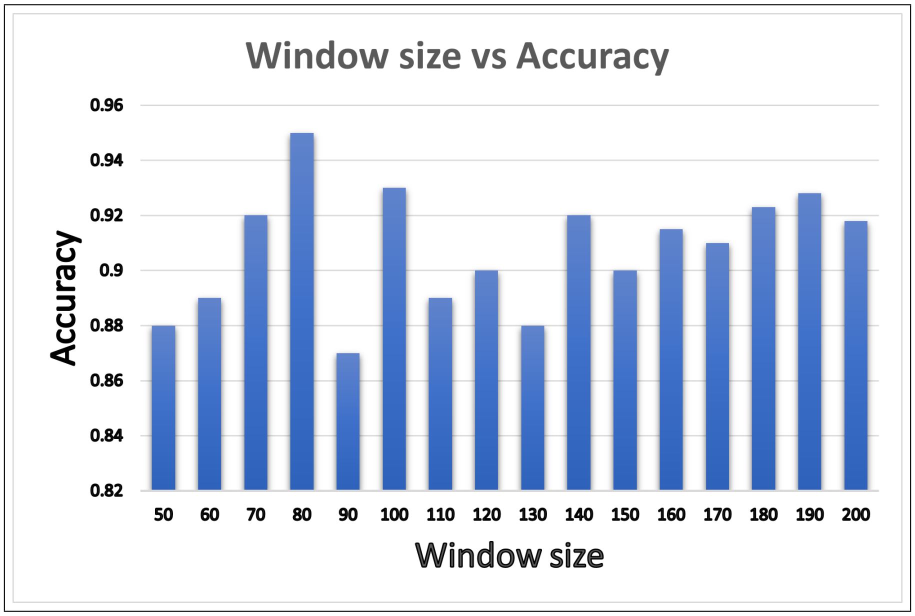


Figure 3.5: Window Size Vs Accuracy

experiments with various window sizes, we have identified that the best window size for detecting all the events in this dataset is 80 for LSTM model as shown in Fig. 3.5.

3.8.4 Deployment and Testing on Raspberry Pi

In this work, we demonstrate the implementation of deep neural network models such as LSTM and Bi-LSTM and evaluate their efficiency on a Raspberry Pi platform. The Raspberry Pi is a popular single-board computer that has been widely used for numerous applications, including computer science education, robotics, sensor networks and internet-of-things. Raspberry Pi 4 Model B+ with 1.4GHz Cortex A53 with 4GB RAM can run Linux-based OS and support several built-in libraries for Python, C, C++, etc. While LSTM and Bi-LSTM are quite popular models for generic time-series classification problems, there is little prior work about deploying these models on Raspberry Pis, primarily due to the limited processing power of older models of the Raspberry Pi.

Hence, in this work, we have implemented the above proposed time-series classification models to classify various events on Raspberry pi. The model has been trained on a different (more powerful) machine prior to off-loading the model onto the Raspberry Pi because, in the learning phase, a large amount of data is used to calculate the weights and biases of the network, thus requiring high computational resources. Once the learning phase has been completed and the network has been trained, the model can be used for classification with lower computation requirement. After the weights have been calculated, they are stored in the program memory and the proposed models can be executed on a device with a low capacity in terms of RAM.

Table 3.1: Comparision of traditional ML models and proposed models.

Model	Overall Accuracy (%)	Straight (SL)	Left Turn (LT)	Right Turn (RT)	Bump (SB)	Stop (ST)
KNN	88.46	0.95	0.21	0.62	0.41	1.00
SVC	41.02	1.00	0.56	0.59	0.42	0.78
Decision Trees	87.17	0.53	0.59	0.67	0.47	1.00
Random Forest	89.74	0.58	0.50	0.67	0.48	1.00
Naive Bayes	84.61	0.58	0.46	0.67	0.11	1.00
LSTM	**95.72**	**0.99**	**0.90**	**0.93**	**0.74**	**1.0**
LSTM with attention	**96.58**	**0.97**	**0.86**	**0.94**	**0.88**	**1.0**
Bi-LSTM	**97.12**	**0.97**	**0.86**	**0.97**	**0.88**	**1.0**
Bi-LSTM with attention	**98.92**	**0.98**	**0.96**	**1.0**	**0.91**	**1.0**

The overall and class-wise accuracies of the deployed proposed models on Raspberry Pi have been tabulated in Table 3.1. From the table, we can observe that the Bi-LSTM with attention mechanism has the highest accuracy as well as equally distributed class-wise accuracy without having any bias towards a particular class. The LSTM and Bi-LSTM models with attention mechanism have achieved better overall and class-wise accuracies, implicitly showing the importance of attention mechanism and time series-based classification.

3.8.5 Performance of Proposed models on Rapspberry Pi

3.8.5.1 Optimization of the models

Before deploying on the Raspberry Pi, the models have been quantized w.r.t. model size. In this work, we have used the TensorFlow Lite framework. Quantization works by reducing the precision of the numbers used to represent a model's parameters, which by default are 32-bit floating-point numbers. This results in smaller model size and faster computation. We have used the dynamic range quantization technique, in which weights are converted to 8-bit precision values. Dynamic range quantization achieves a significant reduction in the model size. There is a significant reduction in model size in exchange for minimal impact to accuracy. We achieved a more than four times reduction in model size with optimization to 8 bits, which ensured a feasible model capable of fast inference on edge devices.

3.8.5.2 Trade-off between accuracy, model size and inference time

After performing the quantization of the models, we have compared the accuracy, memory used, and inference time for the proposed models. From Table 3.2, we can observe that the model size after optimization has been significantly reduced. Similarly, there is a notable decrease in the inference time, making the deployed models faster. Among the optimized models, Bi-LSTM has the highest accuracy while LSTM has the most diminutive model size and inference time. From the table, we can also observe that the %reduction in accuracy, model size and inference time in the case of Bi-LSTM are 0.37%, 43.44% and 54.27% respectively. In the case of Bi-LSTM with attention the % reduction in

Table 3.2: Comparision of accuracy, memory usage and inference time for the proposed models before and after optimization.

Model	Accuracy		Memory required		Prediction time	
	Unoptimized (%)	Optimized (%)	Unoptimized (MB)	Optimized (MB)	Unoptimized (sec)	Optimized (sec)
LSTM	95.72	95.19	572.09	434.96	2.39	0.93
LSTM with attention	96.58	96.05	614.07	583.66	3.22	1.68
Bi-LSTM	97.12	96.76	793.66	448.83	5.03	2.3
Bi-LSTM with attention	98.92	97.84	829.01	648.84	5.57	3.36

model size and inference time is 21.73% and 39.67%, respectively, at the cost of 1.09% reduction in accuracy. Accuracy and inference time trade-offs differ from application to application. In the context of driving event recognition, if the edge deployed two-wheeler has no power supply constraints, Bi-LSTM with attention can be used though it consumes more RAM, but provides the best possible accuracy values for all the classes. However, in two-wheelers where power supply is a critical resource (e.g., electric vehicles), models such as LSTM with the smaller sizes are preferred with lesser classification accuracy. These resource-efficient models can also be deployed on low-power controllers like ESP32.

3.9 Conclusion and Future scope

In this work, we have developed a hardware system that can be deployed on a two-wheeler to collect the dataset with various parameters like acceleration, angular velocity, and speed. We have applied various machine learning models to the collected dataset to classify certain driving events. However, we identified that though the overall accuracy of these models is decent, the class-wise accuracies are not up to the mark. Hence, we proposed time-series based DL models which mitigate the problems persisting in the machine learning models. Then, we enhanced the DL models by adding attention layer for superior feature extraction. The experimental study shows the importance of adapting time-series based classification models in the context of driving event recognition. We have also deployed these models on Raspberry Pi to check the performance on an edge device. We have also performed quantization to these models and successfully reduced the model size and inference time. Hence, the proposed models outperform the other models not only w.r.t. overall accuracy but also w.r.t. class-wise accuracies, model size and inference time. This work is just the beginning of a more comprehensive vision. In future work, we plan to classify several other critical driving events. We will carry forward the work of driving event classification and utilize it to facilitate greater safety of two-wheelers.

Chapter 4

Time-Series Analysis for Enhanced Fall Detection and Prediction Systems in Two-Wheeled Transportation

4.1 Time-Series based Fall Detection in Two-Wheelers

Driving event recognition plays a crucial role in understanding and enhancing road safety. This research focuses on developing efficient time-series based models for Fall detection in two-wheelers. Traditional machine learning models proved inadequate in accurately classifying Fall scenarios due to their inability to capture temporal transitions in kinematic states. To address this limitation, time-series based Deep Learning (DL) models are proposed, utilizing Long Short-Term Memory (LSTM) networks. These networks enable direct learning from raw time series data, eliminating the need for manual feature engineering. Additionally, Bi-LSTMs were employed to capture contextual information from both past and future timesteps, further improving the model's understanding of driving events. The architecture was enhanced with an attention mechanism to boost accuracy. Experimental results showcased that the proposed Bi-LSTM model achieved an overall accuracy of 97%, with a specific accuracy of approximately 92% in detecting Fall scenarios. This research contributes to the development of an accurate Time-series based system for Fall detection, facilitating improved road safety in the context of two-wheelers.

4.1.1 Introduction

Transportation plays an indispensable role in our lives, and as advancements continue to shape this sector, certain challenges have emerged. One significant issue faced by developing countries is the higher proportion of two-wheeler accidents. Motorcycles and scooters are widely used as primary modes of transportation in these regions, but unfortunately, they pose a greater risk to riders due to their inherent vulnerability. Unlike enclosed vehicles, two-wheelers lack structural protection, leaving riders significantly more exposed to injuries in the event of a Fall. This vulnerability is reflected in the statistics, as highlighted in a recent report published by the Ministry of Road Transport and Highways, India. According to the report [4], more than a third (37%) of road accident fatalities in 2019 involved two-wheeler riders. The problem extends beyond developing countries, as motorcycle and moped fatalities account for a considerable proportion (17.7%) of the total number of road accident fatalities in Europe [5]. Comparatively, the likelihood of a motorcycle rider dying in a Fall is 26 times higher than that of a passenger car occupant, considering the distance traveled. These distressing figures clearly indicate that riders are among the most vulnerable road users [7].

Every day as many as 1,40,000 people are injured on roads across the world, of which more than 3000 die and around 15,000 are disabled for life [1]. The implications of such accidents are not limited to the individual riders alone; they also pose a significant risk to the general population. When an accident occurs, the response time to provide medical assistance and minimize harm is crucial. Unfortunately, the inherent risks associated with two-wheelers, coupled with the lack of structural protection, often result in severe injuries or fatalities. This increased reaction time, compounded by the vulnerability of riders, further contributes to the alarming number of deaths on the roads.

4.1.2 Motivation

The frequency of two-wheeler Fall poses a serious threat to road safety, needing a thorough understanding of the underlying causes. Numerous factors, including rider behavior, vehicle attributes, road conditions, weather, and traffic circumstances, have an impact on these collisions [45]. The intricate interactions between human behavior, infrastructure, and environmental elements that cause these incidents can be better understood by thoroughly examining these factors. However, regardless of the specific causes behind the occurrence of Falls, the early detection and timely notification of accidents hold immense potential for saving lives. Therefore, implementing a Fall detection system in two-wheelers is of great importance as a safety precaution.

To the best of our knowledge, there has been very little work on identifying Fall scenarios, specifically in two-wheelers utilizing deep learning techniques. To address this gap, we propose the development of a Time-series based Fall detection system for two-wheelers as an extension of our previous work [3]. By training the system on a comprehensive dataset of two-wheeler Fall scenarios, it will learn to recognize and differentiate between normal riding behavior and instances of Fall. This system will leverage time-series based DL algorithms to detect and classify falls accurately, enabling prompt communication with nearby hospitals or emergency services.

The research holds significant potential to revolutionize two-wheeler safety and emergency response systems. Additionally, with the increasing usage of Electric Vehicles (EVs), the fall detection system can play a crucial role in improving the safety and reliability of electric two-wheelers, thereby promoting their adoption in sustainable transportation. Leveraging deep learning capabilities, the proposed fall detection system offers a proactive approach to mitigate risks for two-wheeler riders. Hence, our contributions in this field can be summarized as follows:

1. Due to the unavailability of two-wheeler Fall data, we have used a simulator to generate various Fall scenarios and collect the data.

2. We have compared various traditional machine learning algorithms using the data acquired.
3. We propose time-series-based DL models for Fall detection and demonstrate their superiority over traditional machine-learning models in terms of accuracy.

4.1.3 Related work

Driver behavior is the primary cause of two-wheeler accidents. There have been works on studying driving event recognition in the case of four-wheelers using classical and machine learning approaches. In this context, there are various frameworks [16], [17], [18] that use unsupervised, semi-supervised and supervised models for the multi-class classification of driving maneuvers and also identify the specific types of abnormal driving behaviors from sensor fusion data of four-wheelers. A few works on driving behavior studies for two-wheelers are presented next.

4.1.3.1 Driver behavior studies for two-wheelers

There are some frameworks developed using traditional machine learning models for two-wheelers. Mitrovic proposed a simple system based on accelerometers, gyroscopes, and GPS data to recognize patterns using HMMs [20]. In [19], a machine learning framework was proposed to identify the class of riding patterns using data collected from 3-D accelerometer/gyroscope sensors mounted on motorcycles. Additionally, they also proposed an approach for sensor selection to identify the significant measurements for improved riding pattern recognition. But this work does not capture the kinematic state change of moving vehicles. Hence, to capture those dynamic transitions, we have proposed time-series-based classification models for two-wheelers. In [41], the authors adopted a Machine Learning based movement identification process with an Artificial Neural Network (ANN) algorithm.

There are some studies based on deep learning as well in the context of time-series classification in general. LSTMs are proven to excel in learning, processing and classifying such types of data. Schalk Wilhelm Pienaar [21] proposed an LSTM-RNN Deep Neural Network Architecture for human activity recognition signifying the importance of the usage of RNN for time-series data. A prior work [22] deals with collision and hazard detection for motorcycles. This is usually done by setting absolute thresholds on the accelerometer measurements, which is not intuitive. In [23], they have used to GMMs and KNN to identify fall and near fall scenarios. In [26], the authors have proposed an airbag system using LSTM to decide on the deployment of a wearable bike airbag in case of an accident.

4.1.4 Proposed Methodology

The focus of this study is to develop a Fall detection system using time-series based deep learning techniques. Our prior work [3], which involved the development of time-series based models for the analysis and classification of different driving events. In this current study, we extend our research to address the critical scenario of Fall detection. By leveraging deep learning techniques and analyzing

Figure 4.1: Snapshots of various scenarios simulated in Bikesim during simulations.

time-series data, we intend to create a robust system capable of accurately detecting and identifying Falls. This work represents an important step forward in enhancing the understanding and response to Fall events, thereby saving precious lives.

4.1.4.1 Data Collection

Given the unavailability of a real-world Fall scenario dataset and the challenges associated with collecting real-time data due to safety risks, we employed a simulator called BikeSim [46] to generate diverse Fall scenarios that closely resemble real-world situations. BikeSim is a highly regarded tool for simulating the performance of two and three-wheeled vehicles, offering high accuracy, detail, and efficiency. With over two decades of real-world validation, BikeSim has become the industry standard for analyzing motorcycle dynamics. Therefore, we utilized this simulator in our research to create a range of Fall scenarios.

Our simulations consists of various scenarios commonly encountered during motorcycle rides, including left and right turns, traversing speed bumps, riding straight, swaying and coming to a stop. In the case of Fall scenarios, we specifically simulated situations that are prone to lead to Fall. For instance,

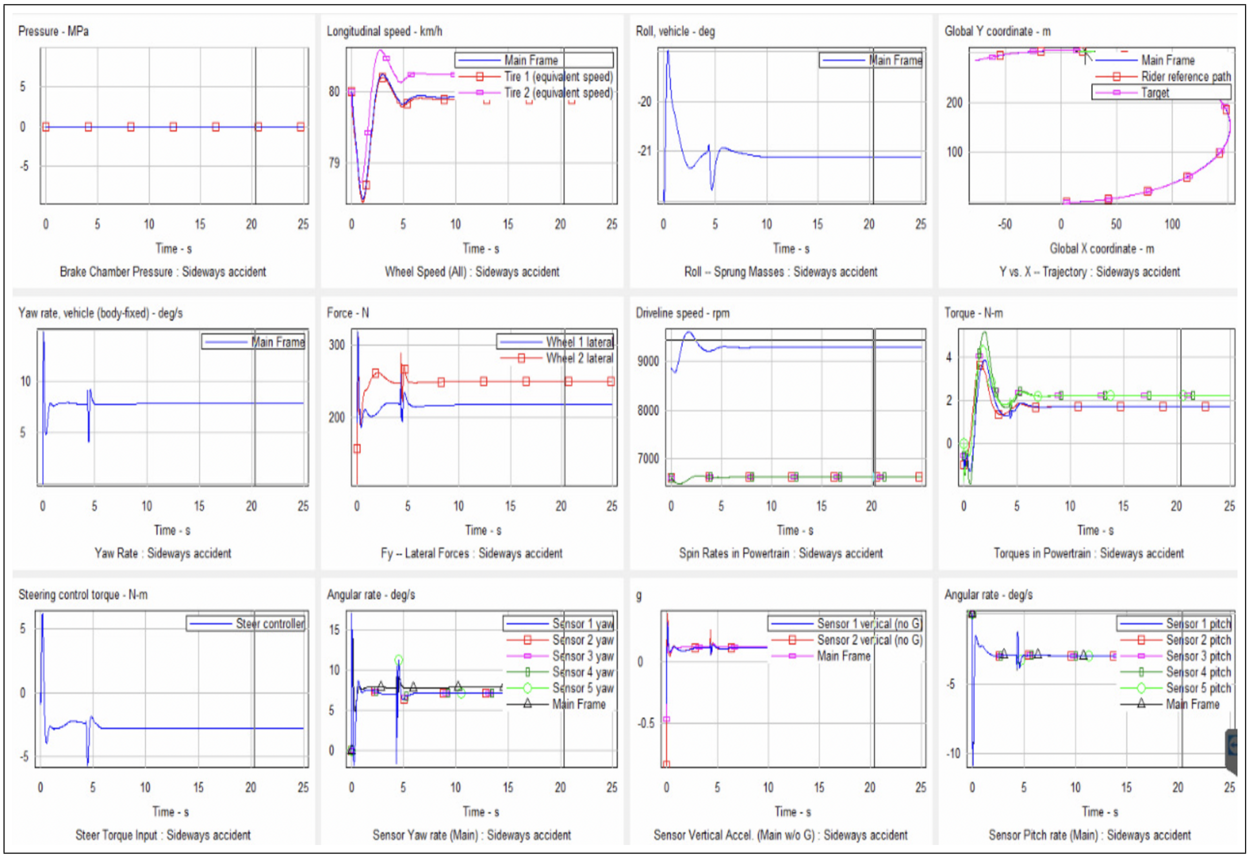


Figure 4.2: Graphs depicting the variations of different parameters during a simulation in Bikesim

taking steep turns at high speeds can result in Fall. To ensure a comprehensive understanding of Fall dynamics, we generated Fall scenarios with varying intensities, such as rolling over and falling.

By utilizing the BikeSim simulator, we were able to accurately replicate real-world riding conditions and generate Fall scenarios that closely resemble actual events as shown in Fig. 4.1. This approach allowed us to study and analyze the dynamics and patterns associated with different Fall scenarios, providing valuable insights into the factors contributing to Falls and the potential consequences for riders. The simulated Fall provides a controlled environment for investigating Fall detection methodologies and developing effective algorithms that can be used in detecting Falls and ultimately enhance motorcycle safety.

The dataset collected from BikeSim consists of several parameters, including A_x, A_y, A_z(acceleration in the x, y, and z directions), G_x, G_y, G_z (angular velocity around the x, y, and z axes). During our preliminary data analysis, we observed significant variations in these parameters over time specifically in the context of Fall scenarios.

In the case of Fall scenarios, the acceleration parameters (A_x, A_y, A_z) exhibited notable fluctuations that deviated from typical riding patterns. These fluctuations can indicate sudden changes in the vehicle's motion, such as sharp deceleration or unusual lateral movements, which are indicative of a Fall event. Similarly, the angular velocity parameters (G_x, G_y, G_z) captured the rotational movements of the vehicle. The Fig. 4.2 depicts the variations of various physical parameters such as longitudinal speed, angular velocities (yaw, pitch, roll), Force, vertical acceleration, torque, etc. All these parameters have been captured during the simulation, but only Acceleration and angular velocity values in the x, y, and z directions have been used for training the models. In Fall scenarios, these parameters demonstrated irregular patterns, deviating from the expected smooth and controlled movements observed during regular riding.

4.1.4.2 Classification using traditional machine learning models

We initially employed traditional machine learning models, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests (RF) for classification. While these models achieved high overall accuracy, their performance in classifying Fall scenarios was notably poor. The discrepancy arises from the dynamic nature of the vehicle, where kinematic states like acceleration, deceleration, and angular velocity undergo significant changes during driving events such as turns, Fall, and braking. Traditional machine learning models struggle to effectively capture these transitional patterns.

In contrast, neural network models exhibit the ability to learn complex temporal relationships, making them well-suited for capturing the dynamic changes in kinematic states during driving events. By training these models on the time-series data collected from the vehicle, they can effectively detect and classify different driving events, including Fall scenarios. The inclusion of temporal information enables the models to capture nuanced variations in the data, enhancing their accuracy in identifying Fall events.

4.1.4.3 Proposed DL-based time-series classification Models

To address the above limitation, we propose time-series based classification models that are capable of capturing and understanding the temporal transitions in kinematic states. By leveraging neural network models, we can train classifiers that have the capacity to capture and learn these intricate transitions. Time-series-based models offer the advantage of considering the sequential nature of the data, enabling them to recognize patterns and dependencies over time that eventually lead to Fall.

1. LSTM In addition to the rationale mentioned earlier, the reason for choosing LSTM [42] to perform driving event classification is its ability to learn from the raw time series data directly thereby eliminating the need to manually engineer input features. LSTM is an efficient recurrent neural network that can hold information from the time series data for a longer duration of time. It can be used to model sequential data and is hence used to learn complex human behavior while riding two-wheelers.

2. Bi-LSTM Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. The input sequence given to the network con-

sists of the six features (A_x, A_y, A_z, G_x, G_y, G_z) of the dataset. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTM's on the input sequence [43]. The first LSTM traverses on the input sequence in the given order, whereas the second one on the reversed copy of the input sequence. This can provide additional context of the driving event to the network and result in faster and even fuller learning on the problem.

4.1.4.4 Attention mechanism

The attention mechanism [44] emerged innately from problems that deal with time-varying data (sequences). The main objective of the attention mechanism is to filter the critical representations out for the purpose of recognition. An attention mechanism is used to redistribute the weights of representations. It can highlight the vital information from the contextual information by setting different weights. Our attention function is straightforward; it takes the dot product of weights and inputs followed by adding bias terms. After that, we add a *tanh* followed by a *softmax* layer. In time-series problems, all sequence elements generally contribute equally to the result, but this may not be the case. For example, a sudden change in acceleration along one direction could better indicate a particular driving event. Hence, capturing those features contributing to recognizing a particular event is critical. Hence, we have enhanced the LSTM and Bi-LSTM models by focusing on specific features that have more impact in recognizing a particular event by embedding an attention layer.

1. LSTM with attention mechanism LSTM cells can't understand long terms dependencies from arbitrary lengths. Therefore, their performance degrades as the sequence length increases. As the name suggests, attention furnishes a mechanism where output can *attend to* a particular input time step for an input sequence of arbitrary length. Hence, an attention layer is embedded on the LSTM layer. The simple LSTM model cannot capture these critical features.

2. Bi-LSTM with attention mechanism

   An attention mechanism focuses on the information fed out from the hidden layers of Bi-LSTM. The simple Bi-LSTM structure allows the networks to have both backward and forward information about the sequence at every time step. In this work, we have added an attention layer over the Bi-LSTM layer for enhanced feature extraction. The output of the attention layer is given to the dense layers.

4.1.5 Experimental results and discussions

4.1.5.1 Experimental setup

The aim of this work is to develop a system and efficient DL models that outperform existing approaches in driving event classification. The training was conducted on a MacBook Air M1, which

Model	Overall accuracy	Normal	Fall
SVM	0.904	1.00	0.243
RF	0.934	0.93	0.765
LSTM	0.941	0.973	0.807
LSTM-attn	0.969	0.981	0.846
Bi-LSTM	0.968	0.943	0.884
Bi-LSTM-attn	**0.976**	**0.962**	**0.923**

Table 4.1: Comparison of accuracies of the proposed models.

features an Apple M1 chip with an 8-core CPU and 8-core GPU. We have used Jupyter notebook to perform the experiments. The framework used in our work to build various models is TensorFlow-Keras. In order to evaluate the models, we have used the 'accuracy', the most commonly used evaluation metric as denoted in Eq. (4.1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

where true positives (TP) and true negatives (TN) denote the correct classifications of positive and negative examples, respectively. False positives (FP) represent the incorrect classification of negative examples into the positive class, and false negatives (FN) are positive examples incorrectly classified into the negative class.

4.1.5.2 Data pre-processing

A pre-processing step is essential to replace the missing values to ensure the continuity of data and synchronization with the video. The database comprises approximately 25,000 data points, consisting of various driving events such as left turns, right turns, straight rides, and stops categorized as 'Normal'. Additionally, it consists of critical scenarios like 'Fall'. The dataset is divided into training and test set consisting of 80% and 20% of the original data, respectively.

4.1.5.3 Results

The obtained results are presented in Table I, showcasing the overall and class-wise accuracies of the proposed models. The table reveals that the Bi-LSTM model with an attention mechanism exhibits the highest accuracy, particularly in Fall detection. Both the LSTM and Bi-LSTM models with attention mechanisms demonstrate higher overall and class-wise accuracies compared to other models. This highlights the importance of attention mechanisms and their ability to capture relevant patterns and features within the temporal data. Although the Bi-LSTM model has slightly lower overall accuracy than the LSTM model with attention, it exhibits superior performance in detecting Fall scenarios. This indicates its sensitivity towards Fall-specific patterns. On the other hand, the LSTM model with attention, while achieving decent overall accuracy, does not perform as well in detecting Fall scenarios. This indicates limitations in capturing the distinctive features or patterns associated with Falls.

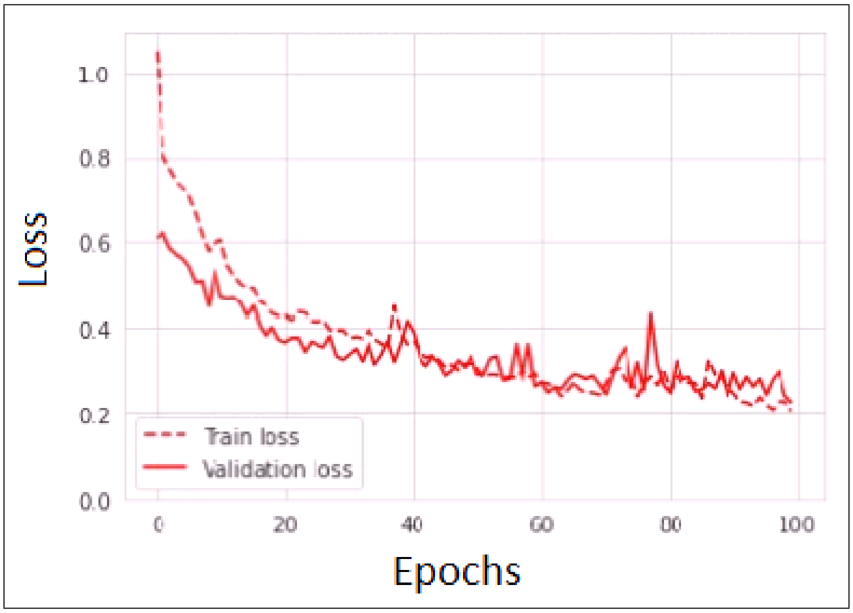


Figure 4.3: Epochs vs Loss

The variation of accuracy and loss over the number of epochs for the training and validation dataset for the LSTM model is demonstrated by Fig. 4.3. Initially, the validation accuracy increases then slows down. After 80 epochs, the accuracy and loss values become stable. At 100 epochs, the model has converged.

4.1.6 Concluding Remarks and Future scope

In this work, we have addressed the challenge of critical driving event classification, with a specific focus on Fall detection. We have simulated various critical scenarios using a simulator. The proposed time-series-based models exhibited superior accuracy in detecting fall scenarios compared to traditional machine learning models, highlighting the significance of considering temporal factors in classification. The proposed models, particularly the Bi-LSTM with attention mechanism, demonstrated superior performance in detecting Fall scenarios, highlighting the importance of attention mechanisms and time-series-based classification. Implementing this fall detection system can potentially reduce response time for medical help, ultimately decreasing fatalities. There are several avenues for further exploration and enhancement of this work. The performance of the proposed models can be evaluated on larger and more diverse datasets, including real-world driving data, to validate their effectiveness in practical scenarios. In addition to the proposed fall detection system, there is potential for further advancements in developing models for fall prediction to predict potential fall events before they occur and alarm the rider.

4.2 Enhanced Two-Wheeler Safety leveraging Isolation Forest for Pre-Impact Fall Prediction

In two-wheeler safety, the prior works have predominantly focused on post-accident detection systems. However, the importance of prediction systems need not be overstated. This work delves into the development of a proactive predictive model specifically tailored to predict falls by Leveraging Isolation Forest, emphasizing the significance of rider behaviour as the primary input. By aggregating data points leading up to falls, the model offers a holistic understanding of fall events, enabling real-time interventions. With the ultimate goal of preventing accidents rather than merely reacting to them, this research embodies the data-driven approach to enhance two-wheeler safety. This work also investigates how variations in lead time influence the accuracy of fall prediction. This examination sheds light on the trade-off between lead time and prediction accuracy. Leveraging a dataset generated through robust simulation, this work demonstrates the superiority of the proposed fall prediction model over the conventional machine learning models in terms of prediction accuracy.

4.2.1 Introduction

Transportation is a vital aspect of modern life, playing a critical role in both personal mobility and the broader economic landscape. As technological advancements continue to influence this sector, new challenges arise, particularly in the context of road safety in developing countries. These regions are witnessing a concerning surge in accidents involving two-wheelers, such as motorcycles and scooters, which are often the primary means of transportation due to their affordability and maneuverability in dense urban environments. The increased risk associated with two-wheelers stems from their fundamental design. This lack of a protective barrier means that riders are more susceptible to direct impact during accidents. The inherent instability of two-wheelers also contributes to this risk. Unlike four-wheeled vehicles, which remain stable at rest, two-wheelers require constant balance from the rider, increasing the potential for falls and collisions, particularly for less experienced riders or in poor road conditions.

This increased vulnerability is clearly reflected in recent accident statistics, which indicate a disproportionately high rate of accidents and fatalities involving two-wheelers in these regions. The recent national road safety [4] report released by the Ministry of Road Transport & Highways, Government of India, casts a stark light on the escalating crisis of road accidents and fatalities in India, presenting an acute public health concern that necessitates immediate attention. Even as the COVID-19 pandemic seemingly reduced traffic volumes, the data from 2022 presents a contrary narrative—an increase in road accidents by 12% to a total of 461,312 incidents, and a concomitant rise in fatalities by 11%, resulting in 168,491 lives lost [47]. This phenomenon is most acute among two-wheeler users, who represent a staggering majority of these fatalities. In a single year, the mortality rate for these individuals surged from 69,385 to 74,897.

The WHO South East Asian Region, for example, experiences 43% of all road traffic deaths involving powered two- and three-wheelers [1]. In countries like Thailand and Cambodia, motorcycle deaths accounted for 73% and 74% of all road fatalities in 2016. Additionally, young adults aged 15–34 years constitute over 60% of all powered two-wheeler related deaths in these countries. Such figures not only reflect the inherent risks associated with two-wheeler transportation but also highlight the critical need for robust research into preventative strategies to curb this growing trend.

4.2.2 Motivation

Accidents involving two-wheelers are a significant safety concern that arises from various factors, including rider error and challenging road conditions [45]. In safety technology, there is a noteworthy distinction between fall detection and fall prediction. Fall detection is about identifying an accident after it has happened, which is useful for emergency services. However, fall prediction is about understanding and identifying the risk before an accident happens, which is far more valuable for preventing harm.

In the current work, fall prediction focuses primarily on rider behaviour. By understanding how riders behave and the mistakes they commonly make, technology can help warn them or take steps to prevent a potential fall. This approach is proactive, aiming to stop accidents before they occur rather than just responding to them afterwards. It's like having a co-pilot who constantly watches for signs of trouble and can help steer the rider away from danger. For two-wheeler riders, this could mean the difference between a safe journey and a serious accident. That's why this research zeroes in on fall prediction—it's about keeping riders safe by addressing the root causes of falls, specifically their behaviour on the road."

To the best of the authors' knowledge, there has been limited research on fall prediction for two-wheelers. Recognizing this, the current research endeavor is committed to advancing the understanding and development of fall prediction technologies for two-wheelers. The main contributions of this work are as follows:

1. In the absence of available data on two-wheeler falls, a simulator was employed to generate a variety of fall scenarios and collect data.

2. Isolation Forest has been proposed for *Pre-impact Fall* prediction.

3. The study presents insights about relationship between lead time extension and predictive accuracy, offering insightful revelations for the development of more effective predictive systems.

4.2.3 Related Works

In the field of two-wheeler safety, various research frameworks have been developed using traditional machine learning models to enhance the prediction and recognition of rider patterns and potential hazards ([16], [17], [18]). Previous studies have utilized data from accelerometers, gyroscopes, and GPS to recognize patterns, employing techniques such as Hidden Markov Models (HMMs) to analyze the data collected [20]. These systems, while foundational, often do not fully account for the dynamic kinematic state changes in moving vehicles.

Building on this, subsequent research has sought to refine sensor data usage and sensor selection, aiming to capture more nuanced riding patterns through data collected from 3-D accelerometer/gyroscope sensors mounted on motorcycles [19]. However, there remains a gap in capturing the full spectrum of vehicular dynamics. To bridge this gap, the introduction of time-series-based classification models provides a more sophisticated approach, allowing for the capture of dynamic transitions more accurately. Beyond traditional machine learning, deep learning has also made significant strides in the realm of time-series classification. Long Short-Term Memory (LSTM) networks [21], a type of Recurrent Neural Network (RNN), have been shown to be particularly effective in processing and classifying sequential data, as evidenced by their successful application in human activity recognition.

In the field of motorcycle safety, innovative approaches for collision and hazard detection have been developed, such as those discussed in [22], which primarily rely on accelerometer data with set thresholds, though this method may not fully capture the intricacies of real-world scenarios. Advancing fall and near-fall detection, studies like [23] have applied Gaussian Mixture Models (GMMs) and k-Nearest Neighbors (KNN) to refine accuracy. Specifically, for two-wheeler fall detection, [48] unveils a cost-effective framework using a microcontroller-based Accident Detection Unit (ADU) with GPS and GSM for prompt emergency alerts in India. Complementing this, [49] details a system that detects vehicle tilt and monitors the rider's heartbeat to quickly communicate accidents to medical centers via a smartphone app, ensuring precise location sharing for faster medical response. Further enriching the domain, [50] introduces a sophisticated crash detection algorithm for two-wheelers, employing GNSS for speed monitoring, along with acceleration and roll angle analysis, offering a nuanced crash identification method that sharply contrasts with those designed for four-wheeled vehicles. In our previous works ([3],[51]) we have used temporal models like LSTMs and Bi-LSTMs with attention mechanisms for detecting various driving events as well as the Fall scenarios. Together, these studies represent significant strides in enhancing two-wheeler safety through technological innovation.

In the domain of *Pre-impact Fall* detection, several studies have laid the groundwork for human fall detection, offering valuable insights that can be adapted for two-wheeler fall scenarios. Notably, methods employing Support Vector Machines (SVM) and Hidden Markov Models (HMM) ([24],[25]) have shown promise. Furthermore, innovative approaches towards enhancing rider safety, including the development of airbag systems for two-wheelers informed by Long Short-Term Memory (LSTM) networks to accurately predict the optimal timing for deploying wearable bike airbags during accidents, have been explored [26]. This emerging research underscores the potential of leveraging advanced machine-learning techniques to improve proactive safety measures for two-wheeler riders.

4.2.4 Methodology

In two-wheeler transportation safety, the focus has traditionally been on systems designed to detect various driving events and fall scenarios after they have occurred, which is explored in our previous works ([3],[51]). While such detection mechanisms are vital for facilitating immediate post-accident interventions, the importance of predictive systems, which aim to preempt accidents by identifying potential risks before they materialize, cannot be overstated. This research takes a proactive stance towards enhancing two-wheeler safety by concentrating on the development of a predictive model. This model uniquely focuses on analyzing the rider's behavior to predict falls, ignoring other factors like the environment or the bike itself. It aims to deeply understand how the actions of a rider can indicate a possible fall.

4.2.4.1 Data collection

Due to the inherent challenges and ethical considerations involved in collecting real-world fall data from two-wheelers, this research has adopted an alternative approach by utilizing the BikeSim simulator. BikeSim [46] is recognized as a robust and industry-standard tool for simulating two-wheeler dynamics and scenarios. Its credibility stems from its widespread acceptance and use within the automotive industry to develop, test, and validate vehicle designs and safety systems.

The use of the BikeSim simulator to meticulously replicate a wide array of fall and non-fall scenarios marks a significant advancement in the context of fall prediction for two-wheelers. By simulating conditions ranging from abrupt braking and navigating steep turns to routine maneuvers such as straight riding and overcoming bumps, the dataset generated provides a comprehensive overview of the varied situations a rider may face. This extensive range of simulated scenarios enriches the predictive model with a better understanding of rider behaviour and the dynamics that lead to falls, allowing for a more accurate distinction between safe and potentially dangerous riding patterns. Furthermore, the inclusion of both fall-prone and normal riding scenarios in our dataset ensures that the predictive model is well-equipped to identify subtle indicators of risk amidst everyday riding activities. This capability is crucial for developing a predictive system that is both sensitive to imminent fall scenarios and specific in its alerts, thereby offering riders timely warnings and the opportunity to avert potential accidents. By laying a robust foundation for fall prediction through detailed simulations, this approach enhances the accuracy of fall prediction models.

The data obtained from BikeSim consists of various parameters, such as Ax, Ay, Az (measuring acceleration across three axes) and Gx, Gy, Gz (gauging angular velocity around the same axes). Initial analysis of this data highlighted distinct patterns, particularly within scenarios leading to a fall. For falls, there were significant changes in acceleration (Ax, Ay, Az) beyond what is typically seen in standard riding behaviour. These changes suggest abrupt alterations in the bike's speed or direction, signalling a potential fall event. Likewise, variations in angular velocity (Gx, Gy, Gz) were observed, reflecting the bike's rotational dynamics during a fall, and offering insights into the complex patterns of such incidents.

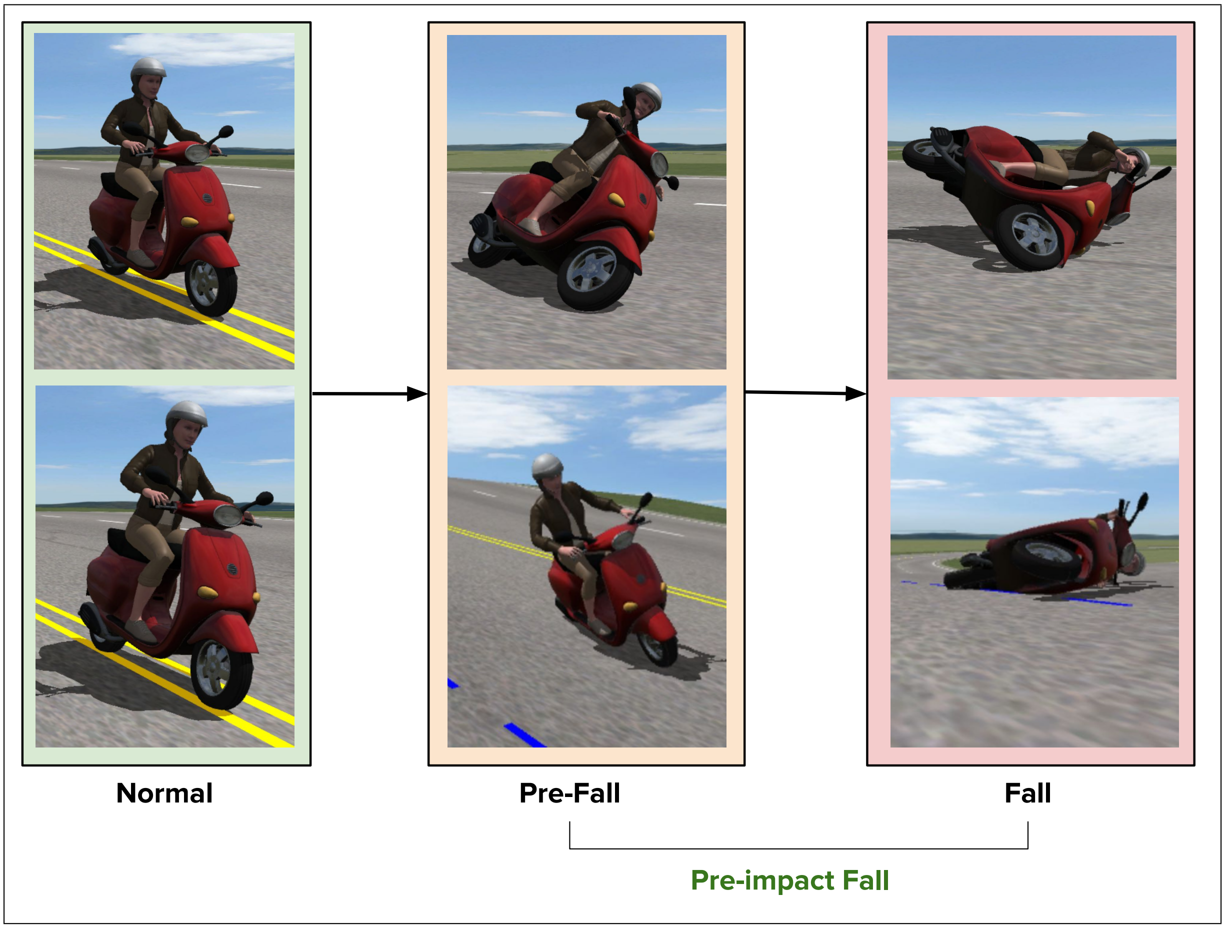


Figure 4.4: Stages of a Two-Wheeler Fall Event.

4.2.4.2 Data Pre-processing and Aggregation

The simulated data, generated with a sampling frequency of 40 Hz, offers a high-resolution view of the rider's actions and the bike's response leading up to the fall. This high sampling rate is essential for capturing the rapid dynamics of two-wheeler falls, where fractions of a second can hold critical information about the factors contributing to an accident. The dataset was meticulously labelled by hand to ensure accuracy in identifying fall incidents. A critical step in our preprocessing involved pinpointing the exact moment a fall begins within the simulated data. This precise identification allowed us to retrospectively examine the sequence of events leading up to the fall, capturing the critical moments just before the incident occurred.

To create a comprehensive view of each fall event, we combined the data points immediately preceding the fall with the instances of the fall itself. This aggregation process was guided by the intention to capture not just the fall, but the conditions and actions leading up to it, providing a rich dataset for analysis. The combined sequences, encompassing both the precursors and the actual fall events, have been collectively labelled as *Pre-impact Fall* as shown in Fig. 4.4. The aggregation of these sequences into the *Pre-impact Fall* category is influenced by the concept of *Lead time*. Lead time can be defined

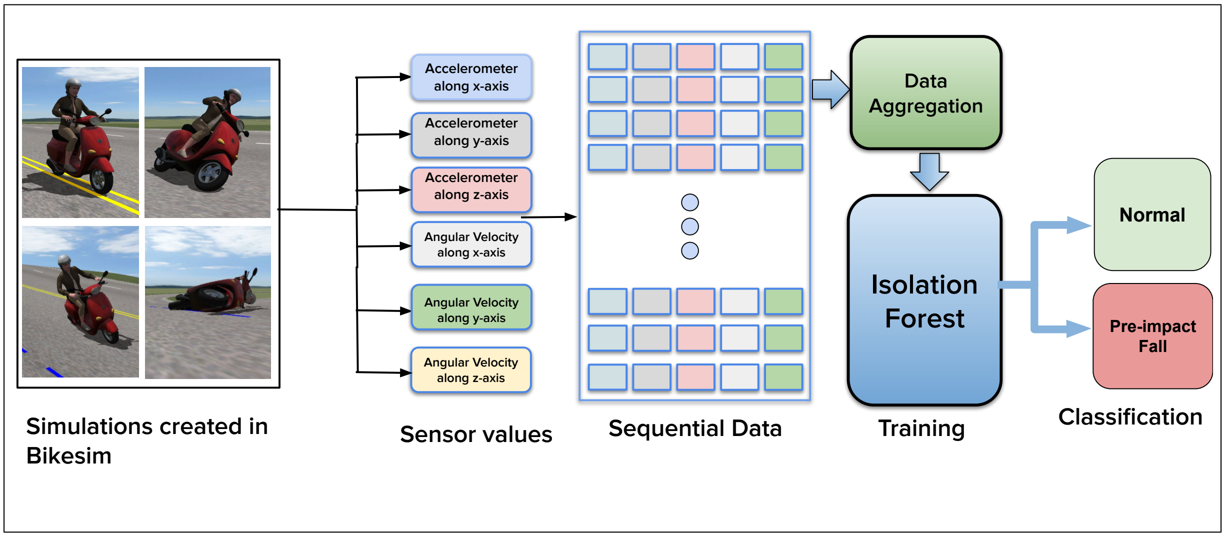


Figure 4.5: Workflow of Fall Prediction Framework for Two-Wheelers.

by the time interval between when the fall was detected and the fall impact, and accounts for the time for protective measures to be activated to protect the fall victims from fall impacts. This label signifies that the sequence of data points provides insight into the progression towards a fall, offering a detailed narrative of the events that culminate in such incidents. The overall flow of the propsed framework is shown in Fig. 4.5. This methodical approach of strategic aggregation of data sequences sets a strong foundation for the subsequent analysis. It ensures that our models are trained on data that accurately reflects the complexity and rapid evolution of scenarios leading to two-wheeler falls, enhancing the potential for effective fall prediction.

4.2.4.3 Proposed Isolation forest model for Fall Predicition

In this work, we propose the utilization of the Isolation Forest [52] technique as a novel model for the prediction of fall incidents in two-wheelers. This method stands out for its anomaly detection capabilities, making it highly suitable for identifying potential precursors to falls within vehicular driving data. The Isolation Forest technique distinguishes itself through its unique approach to detecting outliers, employing binary trees to isolate anomalies rather than profiling normal instances. The algorithmic flow is shown in Algorithm 1. In this model, data is randomly sampled and examined in a tree structure, with analysis based on randomly chosen features. Data points that descend deeper into the tree are deemed less anomalous due to the higher number of splits needed to isolate them. In contrast, data points in shorter branches are marked as anomalies, as they were easier to isolate with fewer splits.

The Isolation Forest algorithm, adapted for fall prediction in two-wheelers, begins by constructing an ensemble of isolation trees from the sensor data, which includes accelerometer and gyroscope readings denoted by $\{ax, ay, az, gx, gy, gz\}$. Each tree in the ensemble isolates data points by randomly selecting a feature and a split value within that feature's range, aiming to segregate each point into its own leaf

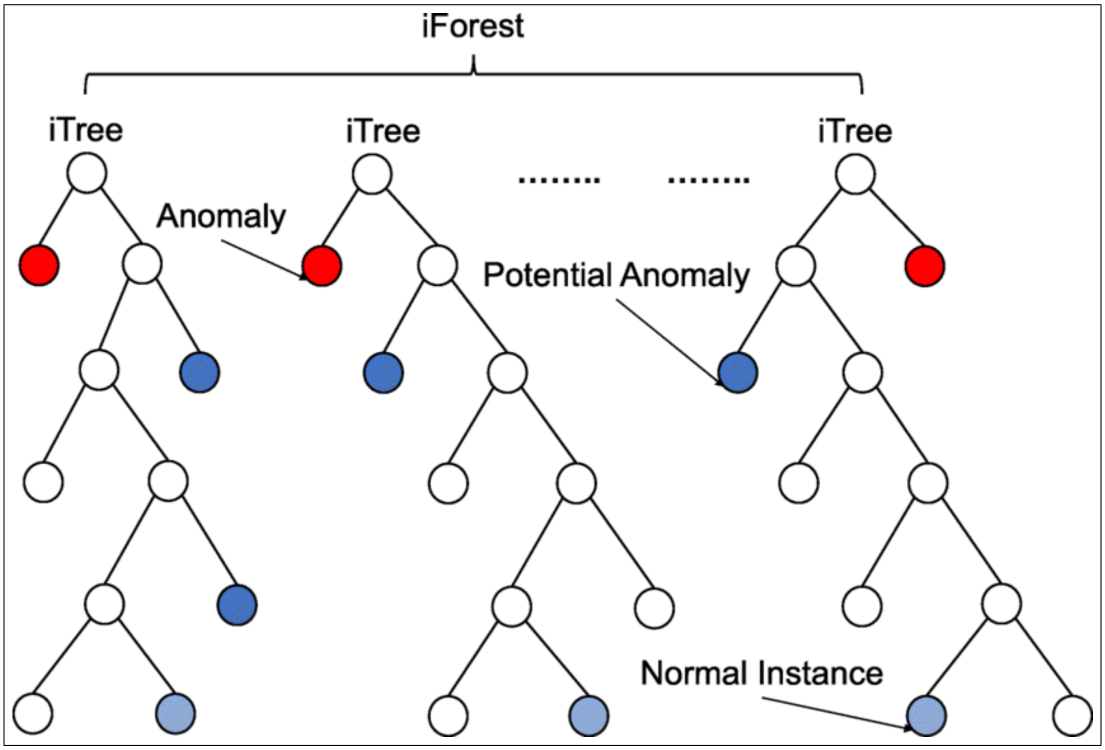


Figure 4.6: Isolation Forest Visualisation.

node. This iterative process either isolates all points or reaches a maximum tree depth as shown in Fig. 4.6. In this representation, red nodes signify anomalies—data points that are isolated with fewer splits, indicative of their divergence from typical patterns. Blue nodes represent potential anomalies, which require more splits than red nodes but fewer than typical instances to isolate. The remaining nodes, depicted in white, correspond to normal instances that conform to the expected data distribution and are isolated after several splits. This visualization captures the essence of the Isolation Forest's approach to anomaly detection, highlighting its ability to efficiently separate unusual data points from the norm.

The underlying hypothesis of the Isolation Forest is that anomalies can be isolated more easily than normal observations, necessitating fewer splits. This is quantified through the path length from the root to the leaf, with shorter paths indicating a higher likelihood of an anomaly. The anomaly score for a data point is calculated by averaging its path lengths across all trees in the ensemble, where shorter average paths signify greater anomaly probability. The mathematical representation of the anomaly score $A(s)$ for a sample s is given by:

$$\text{Anomaly Score} = 2^{-\frac{E(h(x))}{c(n)}} \tag{4.2}$$

Here, $E(h(x))$ represents the average path length of the point x across the trees, $c(n)$ is the normalization factor based on the sample size n, and an anomaly score approaching 1 suggests a high likelihood of the point being an anomaly. This model excels at identifying both the immediate moments

Algorithm 1 Isolation Forest for Fall Prediction in Two-Wheelers

```
Initialize ensemble of iTrees E.
for each T_i in E do
    Select random sub-sample S ⊂ D, D: {ax, ay, az, gx, gy, gz}.
    Assign S to root of T_i.
    while sequences in S not isolated & max depth not reached do
        Select random feature F ∈ {ax, ay, az, gx, gy, gz}.
        Choose θ ~ Uniform(min(F), max(F)).
        Split S into S_left, S_right by θ.
        Apply recursively on S_left, S_right.
    end while
end for
Training complete when all T_i ∈ E constructed.
For scoring, pass sequence through all T_i ∈ E.
Anomaly score A(s) = 1/|E| Σ_{T_i ∈ E} depth(s, T_i).
Score interpretation: A(s) ↗ ⇒ higher fall risk.
```

of a fall and the precursor instances leading up to it, highlighting deviations from normal riding patterns and enabling prompt emergency interventions.

4.2.4.4 Machine Learning Models for Fall Prediction

In the context of Fall prediction, a suite of machine learning classifiers offers diverse approaches for accurate and efficient analysis. The Random Forest Classifier, with its ensemble of decision trees, provides robust predictions even in imbalanced datasets, making it highly suitable for complex fall pattern detection. Gradient Boosting and AdaBoost classifiers refine this approach by iteratively improving on mistakes and focusing on difficult cases, enhancing model sensitivity to subtle indicators of falls. The Support Vector Classifier (SVC) excels in high-dimensional spaces, ideal for binary fall/no-fall classifications, while the K-Nearest Neighbors (KNN) relies on the similarity of data points for prediction, embodying simplicity and effectiveness. Decision Tree Classifier's clear, interpretable structure allows for straightforward identification of fall risk factors, rounding out a comprehensive set of tools for fall prediction in two-wheelers.

4.2.5 Experimental Results and Discussion

In this section, we present the experimental results obtained from our comprehensive evaluation of the machine learning models for fall prediction discussed in the previous section. The experimental phase of our study serves as a critical component, shedding light on the performance and effectiveness of the selected models in real-world fall prediction scenarios. Our aim is to provide a rigorous assessment of each model's capabilities in identifying potential falls before they occur.

Table 4.2: Comparision of Performance of various models.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.729	0.827	0.723	0.760
Gradient Boosting	0.864	0.860	0.863	0.862
AdaBoost	0.853	0.848	0.858	0.852
SVC	0.882	0.886	0.882	0.843
KNN	0.738	0.808	0.733	0.763
Decision Tree	0.729	0.823	0.720	0.757
Isolation Forest	**0.916**	**0.935**	**0.910**	**0.920**

4.2.5.1 Experimental Setup

The aim of this work is to develop an approach to predict the fall scenarios. The training was conducted on a MacBook Air M1, which features an Apple M1 chip with an 8-core CPU and 8-core GPU. We have used Jupyter Notebook to perform the experiments. The framework used in our work to build various models is TensorFlow-Keras. In order to evaluate the models, we have used the 'accuracy', the most commonly used evaluation metric as denoted in Equation (1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

Where true positives (TP) and true negatives (TN) denote the correct classifications of positive and negative examples, respectively. False positives (FP) represent the incorrect classification of negative examples into the positive class, and false negatives (FN) are positive examples incorrectly classified into the negative class.

A crucial preprocessing step involves handling missing values to maintain data continuity and synchronize it with the video. The dataset comprises approximately 20,000 data points, encompassing a variety of driving events, including left turns, right turns, straight rides, and stops categorized as 'Normal'. Furthermore, it includes critical scenarios like 'Pre-impact Fall'. To facilitate model training and evaluation, the dataset is partitioned into a training set, which constitutes 80% of the original data, and a test set, which constitutes the remaining 20%. This division ensures robust model assessment and validation.

4.2.5.2 Results and Discussion

1. Performance of various models: In our study, we rigorously evaluated various machine learning models for their efficacy in predicting pre-impact fall events compared to normal activities. The results from Table 4.2 demonstrate that the Isolation Forest significantly outshines its counterparts, achieving an accuracy of 95.5%, precision of 94.9%, recall of 95.5%, and an F1-score of 94.7%. It's important to note that these results are based on a lead time of 30, equivalent to 750 milliseconds, providing a substantial window for potentially life-saving preemptive actions.

Table 4.3: Confusion Matrix of Isolation Forest.

		Predicted Classes	
		Normal	Pre-impact Fall
Actual	Normal	0.94	0.06
	Pre-impact Fall	0.09	0.91

Although the subsecond granularity may raise issues regarding its effectiveness for manual interventions, given the rapid pace at which a rider might lose control, it's crucial to emphasize the utility of automated safety features. Specifically, this narrow window allows for the activation of automatic safety systems, such as the inflation of airbags, underscoring the significant benefits of Pre-impact Fall detection in enhancing rider protection. These metrics not only highlight the model's exceptional capability in accurately discerning pre-impact fall scenarios from normal activities but also underscore its efficiency in reducing false positives, which is vital for deploying fall prediction systems in real-world settings.

The Isolation Forest model is adept at handling imbalanced data, a common characteristic in fall prediction datasets where pre-impact fall events are substantially less frequent than normal activities. The individual class performance of the Isolation Forest model can be observed from the confusion matrix shown in Table 4.3. The algorithm's design, centered around isolating anomalies, allows it to excel in identifying the patterns that distinguish pre-impact falls, thereby ensuring high recall rates. This means the model is exceptionally reliable in capturing almost all true positive fall events, minimizing the risk of overlooking potential fall situations which is critical for the practical application of fall detection technologies. The outlier fraction setting in the Isolation Forest model greatly affects its performance. The outlier fraction defines the proportion of data points the model considers as outliers or potential fall events. When this parameter is set low, the model is very precise, meaning it can correctly identify most falls but may miss some (high precision, low recall). As the parameter increases, the model starts to catch more falls (higher recall), but it also starts to incorrectly label some normal activities as falls (lower precision). For fall detection, it's crucial not to miss any actual falls (true positives) while keeping the number of false alarms (false positives) low. Therefore, selecting the appropriate outlier fraction is essential for balancing precision and recall, ensuring the model is both reliable and trustworthy in its predictions.

The graphs shown in Fig. 4.7 illustrate the efficacy of the Isolation Forest algorithm in identifying a fall event within a sequence of gyroscope sensor data. The horizontal axis represents the time series index, and the vertical axis quantifies the sensor readings. We have considered a sub-sample sequence from the dataset in which fall has been recorded from 14571 to 14605 index. Throughout

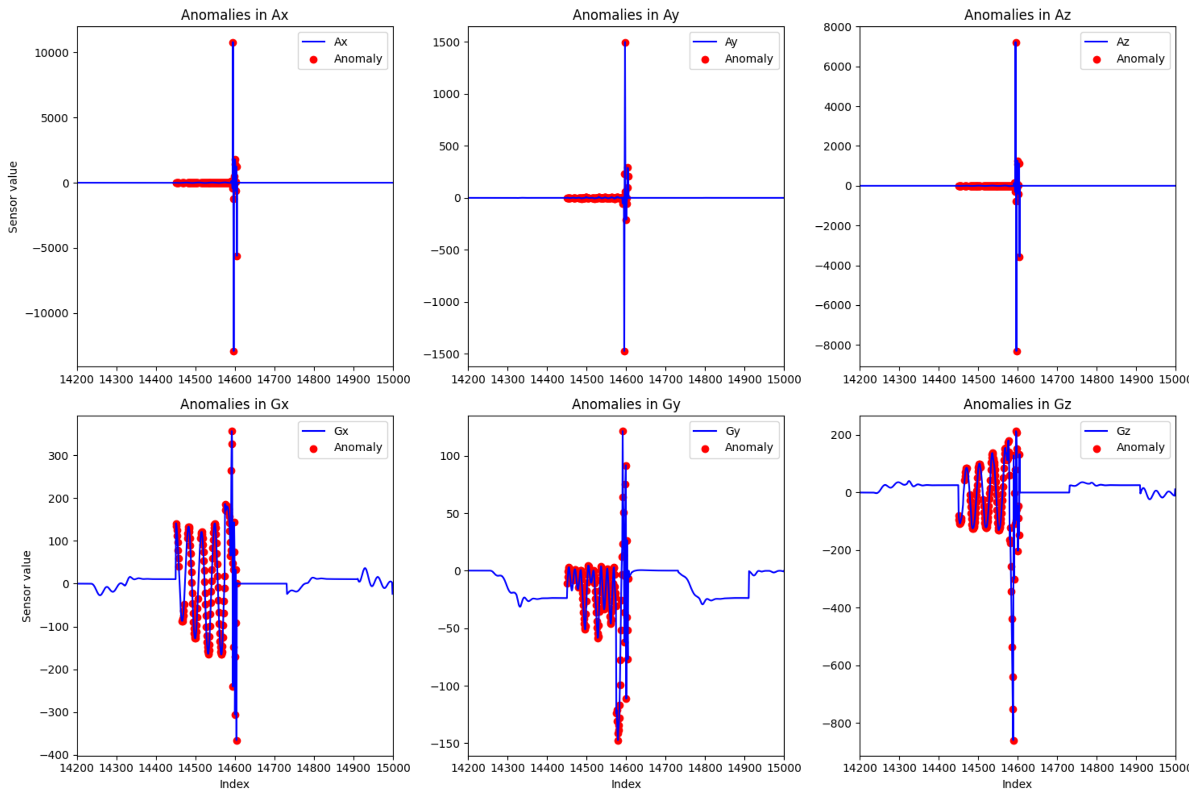


Figure 4.7: Depiction of Pre-impact Fall within a dataset where a fall event has been recorded.

this interval, there is a noticeable perturbation in sensor readings, diverging markedly from typical patterns. These Pre-impact Fall instances are highlighted in red on the graphs. Notably, the Isolation Forest model successfully pinpoints not just the fall event but also the precursor instances that lead up to it (Pre-impact Fall). The sine-like patterns in the sensor data from a two-wheeler, observed before a fall incident, could be attributed to the high sampling rate that captures more detailed cyclical movements. These precursors are particularly valuable, as they may encompass initial indicators of balance loss or instability. The model's ability to capture such precursors underscores its utility for preemptive measures in fall detection systems, enabling interventions before a fall occurs.

Furthermore, the robustness of the Isolation Forest model against noise and outliers significantly enhances its precision and F1-score, showcasing its ability to maintain a balanced performance in accurately detecting pre-impact falls. Compared to other evaluated models like Gradient Boosting, AdaBoost, SVC, KNN, and Decision Tree, none exhibited the comprehensive effectiveness of the Isolation Forest in the context of fall prediction. In our study, we explored deep learning models like LSTM and Bi-LSTM, but faced overfitting with our current dataset. This suggests that while these temporal models show promise, a more diverse dataset is needed to fully harness their potential and enhance accuracy by capturing temporal dynamics more effectively. This analysis

Table 4.4: Performance of Isolation Forest Model for Various Lead Times.

Lead Time	Accuracy	Precision	Recall	F1-Score
10	0.920	0.960	0.920	0.935
20	0.916	0.948	0.916	0.929
30	0.910	0.935	0.910	0.920
40	0.904	0.922	0.904	0.912
50	0.898	0.910	0.898	0.903
60	0.892	0.898	0.892	0.894
70	0.885	0.885	0.885	0.885
80	0.877	0.873	0.877	0.875

solidifies the Isolation Forest model's superiority in predicting *Pre-impact Fall* events, making it an ideal candidate for integration into systems designed to safeguard individuals by proactively identifying and mitigating fall risks.

2. Determining Optimal Lead Time: Optimal lead time is crucial in fall detection systems for two-wheelers because it strikes a balance between early warning and prediction accuracy. An ideal lead time ensures that the system provides timely alerts that are both reliable and actionable, allowing for interventions that could prevent the fall or mitigate its impact. This concept is particularly important because too short a lead time might not offer enough advance notice to take preventive action, while too long a lead time could lead to a high rate of false alarms or irrelevant alerts due to the dynamic nature of riding scenarios.

   In Table 4.4, as the lead time increases from 10 to 80 units equivalent to 250ms to 2000ms, a notable trend emerges in the precision and recall values. Both precision and recall exhibit a gradual reduction as we move further away from the fall event, providing empirical support for the intuitive notion that patterns related to *Pre-impact Falls* become clearer and more distinguishable as we approach the actual event. This reduction in precision and recall with increasing lead time is consistent with the expectation that the closer we are to the fall incident, the more distinct and recognizable the patterns associated with impending falls become. This pattern holds true for all the proposed machine-learning models.

   From the Table 4.4 we can also observe that when predicting *Pre-impact Falls*, a shorter lead time leads to higher precision and recall, underscoring the importance of timely detection and intervention to prevent fall-related injuries. It's important to note that having higher lead times provides the leverage to avoid falls and can act as feedback to the rider, enabling them to take timely action. These findings emphasize the significance of considering lead time as a critical factor when developing predictive models for fall detection and highlight the trade-off between the timeliness of detection and the accuracy of predictions in such scenarios.

4.2.6 Conclusion and future scope

In this study, we have taken the initial steps towards addressing the complex challenge of *Pre-impact Fall* prediction for two-wheeler safety. Through a comparative analysis of various machine learning models, the Isolation Forest has distinguished itself with superior performance in detecting *Pre-impact Fall*. A pivotal aspect of our work has been the critical examination of the relationship between lead time extension and predictive accuracy, offering insightful revelations for the development of more effective predictive systems. This research sets a promising direction for future exploration to catalyze further innovations, advancing the development of safety technologies that can preemptively alert riders and mitigate the risks of accidents. Future advancements in *Pre-impact Fall* Fall impact prediction hold immense potential, starting with the expansion of fall data diversity to train more sophisticated temporal models. The aim is to achieve precise fall predictions at extended lead times and real-time application efficiency, significantly improving two-wheeler rider safety through timely preventative measures.

Chapter 5

Advancements in Anomaly Detection for Traffic Flow Analysis

5.1 Introduction

In the realm of intelligent traffic management, the identification of anomalies in traffic flow data stands as a pivotal concern. Such anomalies, defined as notable deviations from standard traffic patterns, often manifest as univariate time series fluctuations. The criticality of accurate and prompt anomaly detection cannot be overstated; it forms the backbone of proactive incident response mechanisms that significantly enhance the efficacy of traffic management systems. This introductory discourse aims to elucidate the indispensable role of anomaly detection within the broader context of intelligent traffic management, underpinning its necessity through statistical insights and the manifold advantages it presents. The critical role of anomaly detection in traffic management is fundamentally linked to its capacity to enhance road safety and improve traffic flow. Anomalies, such as sudden stops or unexpected congestion, act as early indicators of possible road incidents or dangerous conditions. According to the World Health Organization, road traffic accidents led to around 1.35 million deaths worldwide in 2016 [8]. Employing anomaly detection proactively enables faster response to emergencies, potentially reducing the severity of accidents and saving lives. Additionally, anomaly detection plays a vital role in making traffic flow smoother. It helps in identifying the unusual patterns of traffic, like unnecessary congestion or interruptions, allowing traffic control systems to make timely adjustments. These adjustments might include changing traffic signals or suggesting different routes to prevent bottlenecks. Such measures are essential when considering the U.S. Department of Transportation's report [12], which states that traffic congestion leads to over $100 billion annually in lost time and wasted fuel for Americans. This information highlights the importance of using anomaly detection not just for safety, but also for efficiency on the roads.

The environmental impact of implementing advanced anomaly detection in traffic management extends far beyond improving safety and efficiency. By enabling more efficient traffic control, anomaly detection leads to smoother traffic flows and fewer instances of vehicles idling. This reduction in idle times significantly decreases vehicular emissions. The Environmental Protection Agency (EPA) has highlighted that the transportation sector is the primary source of greenhouse gas emissions in the

United States, responsible for almost 29% of the nation's total emissions [9]. Therefore, improvements in anomaly detection technology not only aim to decrease the delays caused by congestion but also contribute to creating a more environmentally friendly and sustainable urban ecosystem.

The economic advantages of utilizing efficient traffic anomaly detection systems are significant, highlighting the importance of such technologies in modern traffic management. By reducing congestion, these systems not only save time for commuters but also lead to substantial cost savings for businesses that depend on road transportation for their logistics and delivery operations. The Federal Highway Administration (FHWA) [10] has noted that enhancements in traffic management can result in savings of billions of dollars by minimizing delays and reducing fuel consumption, thus supporting the broader economy. Additionally, anomaly detection is pivotal for the development of smart cities, which aim to improve the quality of urban life and sustainability. By incorporating real-time traffic data analysis, cities can improve road safety and traffic flow, while also working towards wider goals like lowering greenhouse gas emissions, improving public transit systems, and promoting economic growth.

The complexity of traffic flow patterns, influenced by several factors like road conditions, daily commuting patterns, and unforeseen incidents, necessitates robust and adaptive models for anomaly detection. Traditionally, a range of methodologies have been employed to tackle this challenge, as outlined in a comprehensive survey by Braei and Wagner (2020). These methodologies span across statistical models, such as ARIMA (Zhang et al., 2005) [13], classical machine learning techniques like K-Means Clustering (Nairac et al., 1999; Rebbapragada et al., 2009) [14] and One-Class SVM (Eskin et al., 2002) [15], to cutting-edge deep learning methods. The latter, notably, attempts to unravel the intricate, nonlinear correlations in data to predict future traffic patterns and identify anomalies based on deviations from these predictions (Buda et al., 2018; Pang et al., 2021) [11], [53].

In addressing the challenges of anomaly detection in traffic flow data, it is evident that a lot of improvement can be made on this front. These challenges can be as follows:

5.1.0.1 Threshold Dependency

A significant limitation in current anomaly detection approaches is their reliance on *thresholds* to identify anomalies. These thresholds are often either preset or dynamically adjusted. The efficacy of such methods hinges heavily on the accurate setting of these thresholds, a complex task due to variability in traffic conditions and temporal factors. Furthermore, thresholds are inherently location-dependent, as traffic patterns and data scales vary from one road to another. This necessitates the establishment of unique threshold settings for different locations, adding layers of complexity to the anomaly detection process.

5.1.0.2 Lack of Labeled Data for Supervised Learning

Another constraint lies in the supervised nature of many anomaly detection systems. These models require well-labelled datasets for training, which classify data points as normal or anomalous. In real-world scenarios, like ours, such labelled datasets are often unavailable. Manually labelling large datasets to create ground truth for anomalies in traffic data is not only labour-intensive but also impractical, given the volume of data typically involved.

5.1.0.3 Detection of Anomaly Sequences

Our focus is on identifying sequences of anomalous data points within traffic flow time series, as opposed to isolated instances of anomalies. The challenge here is that the length of these anomaly sequences is not fixed and can vary significantly. Most current approaches tailored to sequence-based anomaly detection operate on the assumption of a predefined sequence length. This assumption limits their effectiveness, as it lacks the flexibility needed to accommodate the varying lengths of anomalous sequences that we aim to detect in traffic flow data. The severity of the challenge lies in differentiating between detecting boundaries and identifying segments within the data.

In light of these challenges, our research introduces a novel approach to anomaly detection in traffic flow data, specifically tailored to address the unique characteristics. By leveraging advanced deep learning techniques, we aim to develop a model that not only overcomes the limitations of threshold dependency and label scarcity but also exhibits the flexibility required to detect variable-length anomaly sequences. This work presents the methodology, results, and implications of our findings, contributing to a significant advancement in anomaly detection in the field of intelligent traffic management.

5.1.1 Contributions

1. We synthesized a traffic flow dataset with real-life-like anomalies, offering a practical tool for advancing anomaly detection research. This dataset replicates complex traffic irregularities, supporting the development and testing of anomaly detection algorithms.

2. We proposed an encoder-decoder model enhanced with contextual attention, significantly boosting the detection of complex traffic anomalies. This approach marks a considerable advancement by enabling more nuanced identification of traffic flow disruptions.

3. Leveraging the PeMS dataset, we created a synthetic baseline enriched with realistic anomalies and integrated it with California Highway Patrol (CHP) incident reports. This novel dataset merges actual traffic incidents with traffic flow data, facilitating comprehensive anomaly detection research on a widely recognized traffic dataset.

5.2 Related Works

Anomaly detection in time series data has been extensively researched, with methodologies evolving to address the unique characteristics of different data types and application domains [54, 55]. This section delves into the existing literature on time series anomaly detection, categorizing the methodologies into four distinct approaches: statistics-based, prediction-based, similarity-based, and reconstruction-based methods, with a particular emphasis on their application to traffic flow data.

5.2.1 Statistics-based Anomaly Detection

Historical approaches to anomaly detection have predominantly relied on statistical models, underpinned by the assumption that data conform to specific statistical patterns. This assumption enables the identification of anomalies when new data points significantly deviate from these established patterns [27]. Classical models in this domain include hypothesis testing [28], wavelet analysis [29], and ARIMA [30], each offering a unique perspective on anomaly detection. Notably, Yamanishi et al. [31] leveraged statistical learning theory in an online learning algorithm tailored for anomaly detection. More recent advancements, like the application of extreme value theory by Siffer et al. [32], have refined these techniques for univariate time series, though their focus has predominantly been on anomalies exceeding normal levels, leaving sub-normal anomalies less explored.

5.2.2 Prediction-based Methods

With the advent of more complex data structures and the need for dynamic analysis, prediction-based methods have gained traction in time series anomaly detection. These methods hinge on forecasting subsequent values in a series, flagging deviations from these forecasts as potential anomalies [33]. The advent of deep learning has notably enhanced the efficacy of prediction-based methods. For instance, Buda et al. [34] utilized LSTM models to achieve precise forecasting capabilities, and Hundman et al. [35] applied unsupervised LSTM models for anomaly detection in spacecraft telemetry data. Another innovative approach, DeepAnT, introduced by Munir et al. [36], effectively combined CNNs for time series prediction. However, despite their success in short-term forecasting, these models often falter in rapidly changing environments, such as those encountered in financial markets, underscoring the need for more adaptive and responsive methodologies.

5.2.3 Reconstruction-based Anomaly Detection

Addressing the shortcomings of forecasting-based models, the field has seen a paradigm shift towards reconstruction-based models. These models are particularly adept at handling the challenges posed by rapidly changing time series data. By encoding standard sequences into latent spaces, they enable the detection of anomalies through discrepancies observed during the reconstruction phase in the test data [37]. This approach has proven to be highly effective, especially in semi-supervised learning scenarios where the model is trained exclusively on normal data. This training focus inherently enhances the model's ability to identify deviations, thereby improving the sensitivity and accuracy of anomaly detection.

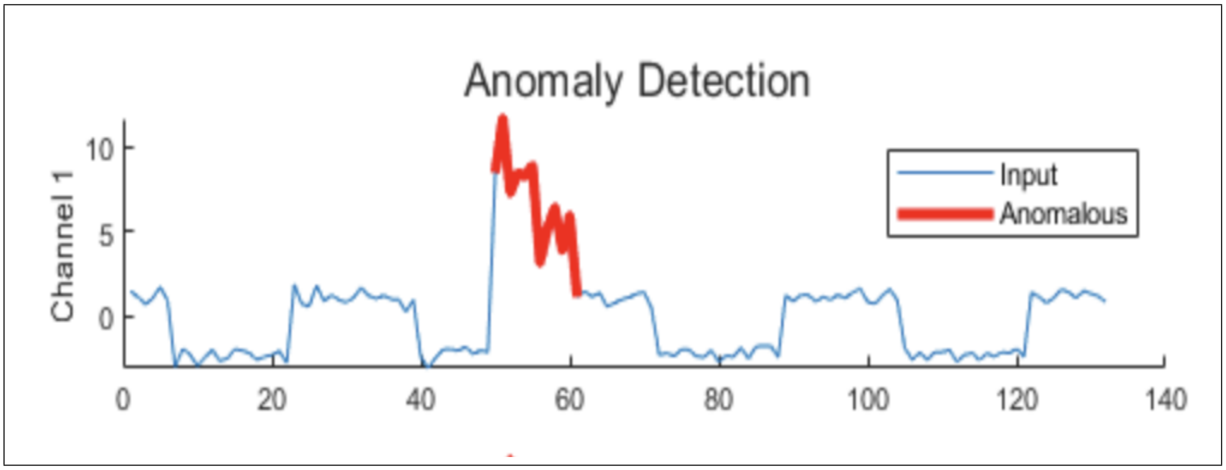


Figure 5.1: Illustration of sequence anomalies in time series flow data, highlighting disruptions in regular patterns over time.

In conclusion, each of these methodological approaches offers unique insights and capabilities in the realm of anomaly detection for time series data. While statistical methods lay a robust foundational framework, prediction-based models introduce advanced forecasting capabilities through deep learning. However, the intrinsic dynamism and complexity of traffic flow time series data call for more adaptable and responsive approaches, such as those offered by reconstruction-based models. Our work seeks to synthesize the strengths of these varied methodologies to develop a comprehensive anomaly detection framework, specifically tailored for traffic flow data analysis.

5.3 Contextual Attention Information based Encoder-Decoder Network

In this research work, we have proposed a reconstruction model using an encoder-decoder architecture, specifically tailored for anomaly detection in traffic flow time series. Our approach involves the LSTM Autoencoder framework, known for its efficacy in processing sequential data. Through the encoder, input sequences are condensed into a compact, lower-dimensional representation, enabling a focused analysis of critical data characteristics. The decoder component then reconstructs the original sequence from this compressed form, allowing us to measure reconstruction loss as a key indicator of anomalies.

5.3.1 Anomaly in Time Series

The concept of an anomaly in time series data pertains to any significant deviation from the typical distribution of the dataset. This deviation can manifest as an aberrant single observation, known as a point anomaly, or as a sequence of unusual observations, termed a subsequence anomaly as shown in Fig. 5.1. Such anomalies are characteristically infrequent, suggesting that the bulk of the dataset adheres

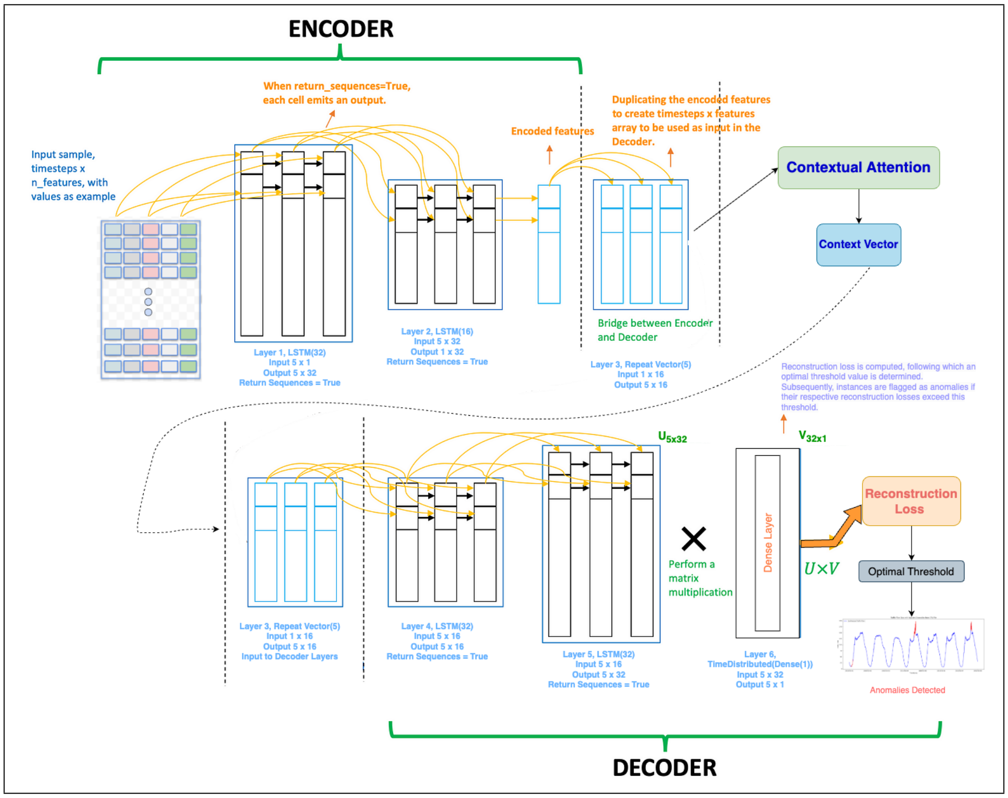


Figure 5.2: Flow diagram of the traffic flow anomaly detection process using an encoder-decoder architecture.

to a normal distribution. However, real-world data, including time series, often encompass substantial amounts of noise, which may be tangential to the core analytical focus.

5.3.2 Proposed Encoder and Decoder Architecture

The overall flow of the proposed architecture is shown in Fig. 5.2. Anomaly detection in traffic flow time series is a critical task for maintaining efficient traffic management systems. We approach this problem using an LSTM Autoencoder – a specific type of Encoder-Decoder LSTM architecture that excels in handling sequence data. The LSTM Autoencoder is composed of two main components: the encoder, which compresses the input sequence into a fixed-length representation, and the decoder, which attempts to reconstruct the input sequence from this compressed form. Fig. 5.3 provides a detailed representation of the encoder-decoder architecture that is depicted in a more general form in Figure 2. The encoder processes the sequence of traffic flow data, $X = \{x_1, x_2, \ldots, x_n\}$, and transforms it into a lower-dimensional representation, $Z = \{z_1, z_2, \ldots, z_m\}$, where $m < n$. The decoder then takes this representation and attempts to reconstruct the original sequence, $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$. The model's performance is quantitatively assessed by measuring the reconstruction loss, which quantifies

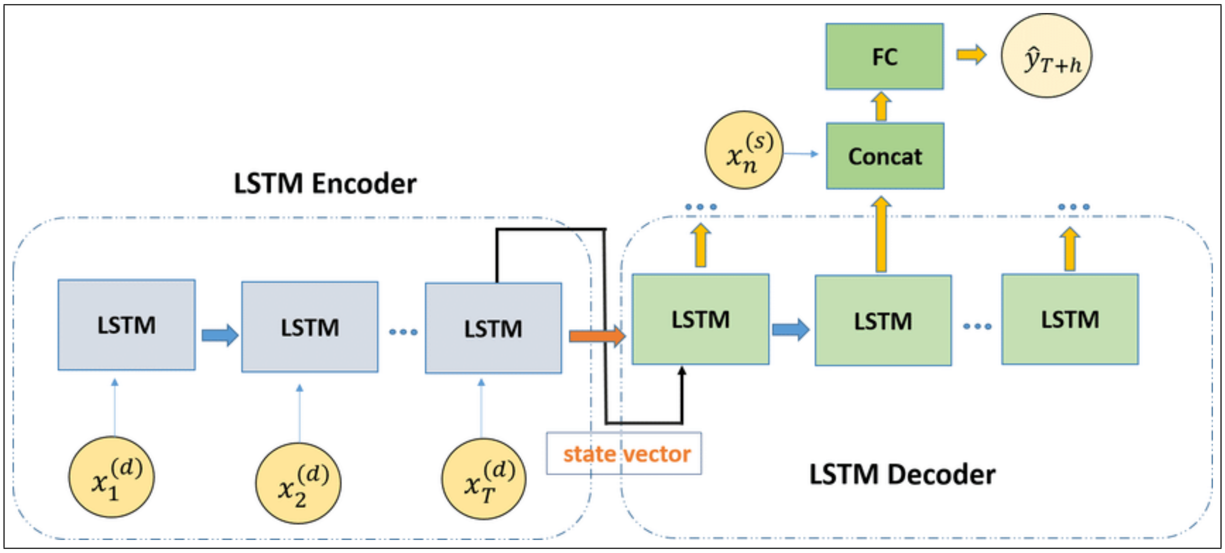


Figure 5.3: Detailed architecture of an LSTM encoder-decoder network for sequence classification.

the difference between the input sequence X and the reconstructed sequence $\hat{X}$. This reconstruction loss is minimized during training.

Once the model is adequately trained, the decoder is detached, and the encoder serves as a standalone model that maps input sequences to a compressed vector space. The encoded vectors can be used for various applications, such as input features for supervised learning models or for data visualization purposes. In the context of traffic flow anomaly detection, we utilize the reconstruction loss to discern anomalies. A threshold is established, above which a sequence is flagged as anomalous. This threshold is determined through validation and aims to capture sequences that deviate significantly from the normal traffic patterns learned by the autoencoder during the training phase.

5.3.3 Synthetic Data Generation for Anomaly Detection

In the analysis of time series data, the detection of meaningful deviations, those distinctly divergent from standard patterns, is of paramount importance. The presence of noise does not fundamentally alter the primary characteristics of the dataset. Trend analysis and anomaly detection, while interrelated in the context of time series, are not synonymous. An essential aspect to consider in time series datasets is the phenomenon of concept drift. This term refers to the gradual or abrupt changes in the underlying values and trends over time, profoundly impacting the process of anomaly detection.

Given a sequence of traffic flow data $X = \{x_1, x_2, \ldots, x_n\}$, an anomaly detector processes this sequence and generates an output sequence $Y = \{y_1, y_2, \ldots, y_n\}$, where each $y_i \in \{0, 1\}$ is a binary

Algorithm 2 Anomaly Generation in Traffic Flow Data

Require: Traffic flow data X, Anomaly density ρ, Sequence length range $[l_{min}, l_{max}]$, Error rate range $[e_{min}, e_{max}]$
Ensure: Traffic flow data X with injected anomalies
Initialize synthetic traffic flow data X based on real-world patterns
Determine dataset size based on the duration in months and the resolution (5-minute intervals)
for $day \in$ dataset days **do**
    if Random() ¡ ρ **then**
        Select day for anomaly injection
        Randomly choose an anomaly start time within day
        Determine anomaly sequence length l uniformly from $[l_{min}, l_{max}]$
        Calculate end time of anomaly based on start time and l
        for each time step t in anomaly duration **do**
            Sample error rate e uniformly from $[e_{min}, e_{max}]$
            Inject anomaly by adjusting traffic flow: $x_t = x_t \cdot (1 + e)$
        end for
    end if
end for
return X

indicator. The indicator $y_i = 1$ if the corresponding traffic flow data point x_i is deemed anomalous, and $y_i = 0$ if x_i is considered normal.

The approach to anomaly detection within traffic flow data signifies a departure from conventional methods that utilize a fixed-length sliding window. Traditional techniques may not fully grasp the dynamic nature of traffic anomalies, which can vary significantly in duration and impact. Instead, our method evaluates each data point in the sequence individually, allowing for the identification of anomalies across a spectrum of lengths, represented as $A = \{a_1, a_2, \ldots, a_m\}$, where a_j stands for the length of the j-th anomaly sequence, and m is the total number of detected anomalies.

Moreover, this methodology accounts for the intervals between successive anomalies, denoted by $T = \{t_1, t_2, \ldots, t_{m-1}\}$, embracing the variability in temporal distances between anomalies. By recognizing the interval t_j between the j-th and $(j+1)$-th anomalies, the method accurately reflects the unpredictable nature of traffic flow. Such attention to the nuanced characteristics of traffic anomalies—both in their duration and the intervals between them—ensures a sensitive and precise identification process. This flexible and detailed approach is instrumental in yielding more accurate insights for traffic management and anomaly mitigation strategies.

In traffic flow analysis, particularly for anomaly detection, the availability of real-world data sets that are accurately tagged with anomalous events is exceedingly rare. This scarcity of labeled anomaly data poses a significant challenge for developing and validating traffic flow prediction models. To overcome this limitation, our study necessitates the creation of synthetic traffic flow data, where anomalies are artificially introduced. This approach allows us to simulate a variety of realistic anomaly scenarios, closely representing the complexities and irregularities found in actual traffic conditions. By generating

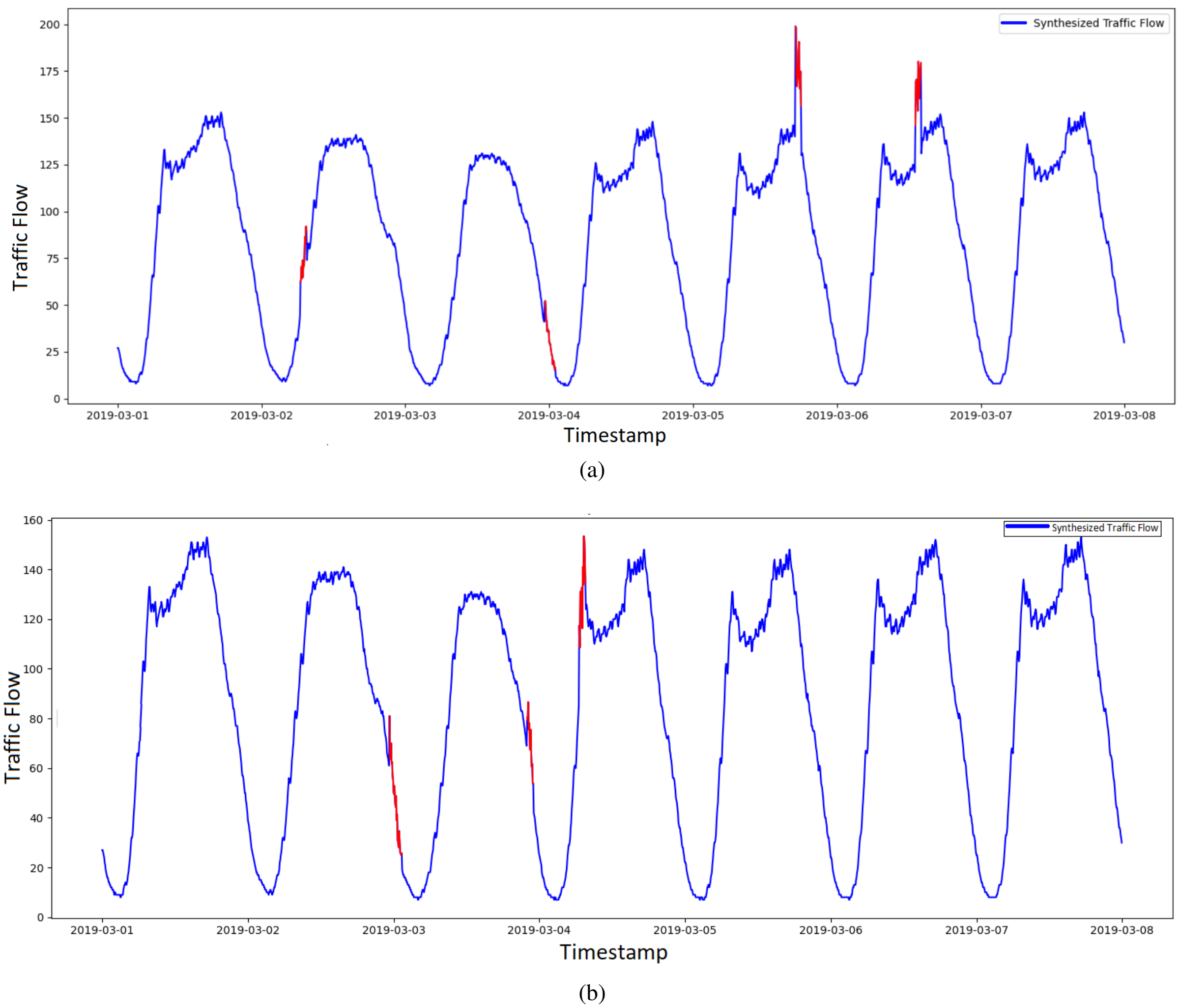


Figure 5.4: Time series visualization of synthesized traffic flow data over a week, depicting normal traffic patterns and anomalous spikes.

synthetic anomalies, we can systematically control and vary the characteristics of these events, such as their frequency, duration, and intensity. This controlled environment is instrumental in rigorously testing and refining our anomaly detection models, ensuring they are robust, adaptable, and capable of handling real-world traffic conditions effectively.

To introduce anomalies into the normal time series, we randomly select specific days, represented by σ, to include an anomalous period and then determine the length and severity of the anomaly sequence. The synthetic anomaly generation process is governed by four parameters: dataset size, anomaly density, sequence length, and error rate. The 'dataset size' parameter specifies the length of the entire synthetic time series in months, with each month encompassing approximately 28,800 time steps, calculated as 3 months $\times\, 20\eta \times 480$ 5-minute steps per day. The 'anomaly density' parameter, denoted as ρ, indicates the proportion of days containing anomaly sub-sequences. For example, a 20% anomaly density in a

3-month series implies 12σ days with anomalies. It is assumed that each selected day contains exactly one anomaly sub-sequence. The ‘sequence length‘ defines the duration of an anomaly sub-sequence in the number of time steps, while the ‘error rate‘ quantifies the deviation of an anomalous flow from the normal value as a percentage. This error rate is used to calculate the anomalous flow value, $\tilde{f}$, from the normal flow value, f, using the formula $\tilde{f} = (1 - e/100)f$, where e is the error rate.

In our methodology, after determining the ‘dataset size‘, we randomly assign days with anomalies based on the ‘anomaly density‘ ρ. For each anomalous day, the length of the anomaly sequence is randomly selected from a uniform distribution defined by the ‘sequence length‘ range. The start time of each anomaly is chosen randomly, and the end time is set as the sum of the start time and the anomaly sequence length. To ensure the initial state is considered normal, we avoid introducing anomalies during the first $m - 1$ time steps of the series. The anomalous flow values within the anomaly period are then assigned based on the ‘error rate‘ randomly sampled from its specified range. Fig. 8(b) presents an example of the synthetic data with anomalies, where anomaly sub-sequences are depicted as red curves within the blue-shaded areas. The injected anomalies in the real-world data can be observed in Fig. 5.4.

5.3.4 Incorporating Contextual Attention

To further enhance the anomaly detection capabilities of our LSTM Autoencoder, we integrate a contextual attention mechanism into the architecture. Contextual attention enables the model to allocate variable importance to different parts of the input sequence, thereby focusing on the most salient features that are indicative of anomalous behavior.

The attention mechanism functions by assigning a weight to each timestep of the input sequence. These weights are learned during the training process and are applied to the encoded representation Z of the input sequence. The weighted representation, or context vector C, is calculated as follows:

$$C = \sum_{i=1}^{m} a_i \cdot z_i, \tag{5.1}$$

where a_i is the attention weight corresponding to the i-th element of the encoded sequence Z, and m is the length of the encoded sequence. The attention weights are normalized across the sequence to sum to one, typically using a softmax function:

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{m} \exp(z_j)}. \tag{5.2}$$

By focusing on the parts of the sequence that contribute most significantly to the reconstruction error, the contextual attention mechanism allows the Autoencoder to become more sensitive to the nuances of traffic flow, which is crucial for detecting anomalies. The resulting context vector C is then used by the decoder for sequence reconstruction, and the attention-weighted reconstruction loss is utilized to identify anomalies. This approach ensures that our model not only captures the general patterns in the traffic flow data but also adapts to the changing importance of different features over time, leading to more accurate and reliable anomaly detection.

Table 5.1: Performance Metrics of LSTM-Based Autoencoder Models.

Model	Precision	Recall	F1 score
LSTM autoencoder-attn	0.8854	0.7509	0.7905
LSTM autoencoder- contextual attn	0.8333	0.7558	0.8278
LSTM-CNN autoencoder	0.8040	0.4300	0.5600
LSTM - AER	0.8380	0.6040	0.7027

5.3.5 Computing Anomaly Scores

The key to effective anomaly detection lies in the precise calculation of the reconstruction loss. We employ mean squared error (MSE) as the loss function shown in (5.3),

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{5.3}$$

which computes the average squared difference between the entries of the input sequence and the reconstructed sequence. Anomalies are detected by examining the reconstruction loss; if the loss exceeds a predefined threshold, the sequence is classified as anomalous. The threshold is carefully chosen based on the distribution of reconstruction loss on a validation set, which reflects normal traffic conditions. By doing so, the LSTM Autoencoder is able to identify sequences of traffic data that contain unusual patterns, indicative of potential anomalies.

In sum, our methodology capitalizes on the temporal learning capabilities of LSTM Autoencoders to detect and represent the most salient features of traffic flow data, enabling the identification of anomalous sequences in a robust and efficient manner.

5.4 Experimental Results and Discussions

In our study, we simulate normal traffic flow time series by combining real-world data: PeMs with synthetically added random fluctuations. We extract three months of flow data (weekdays only) from a specific link and group them into 5-minute intervals based on the time of day. For each interval, we compute the mean and standard deviation of the observed flows. Normal flow time series are then generated by drawing values from a Gaussian distribution with the respective mean and standard deviation for each interval.

The LSTM autoencoder with attention (LSTM autoencoder-attn) achieved a precision of 0.8854 and a recall of 0.7509, resulting in an F1 score of 0.7905. This indicates a strong ability of the model to correctly identify anomalies while maintaining a reasonable detection rate. Improvements were observed with the LSTM autoencoder incorporating contextual attention (LSTM autoencoder- contextual-attn), which attained a higher precision of 0.9230 and recall of 0.7640, leading to an F1 score of 0.8133. The advanced mechanism of contextual attention contributes to a more nuanced understanding of anomalies in the data, as reflected in the improved scores. In contrast, the LSTM-CNN autoencoder model exhib-

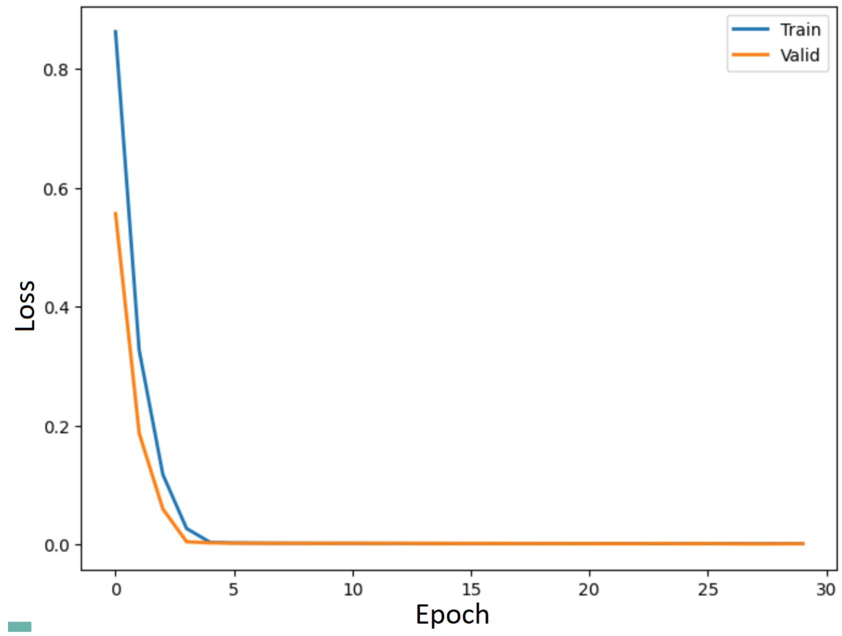


Figure 5.5: Convergence of Model Loss Over Training Epochs.

ited lower recall at 0.4300, despite a reasonable precision of 0.8040. The F1 score for this model was 0.5600, suggesting a need for further optimization to balance the trade-off between precision and recall. The LSTM - AER model, which emphasizes on encoder reconstruction for anomaly detection, reported a precision of 0.8380 and a recall of 0.6040, yielding an F1 score of 0.7027. The model's precision indicates accurate anomaly detection, although its recall suggests potential room for improvement in identifying all relevant anomalies. Overall, the incorporation of a contextual attention mechanism has proven to significantly enhance the performance of LSTM-based autoencoder models for traffic flow anomaly detection.

Our LSTM autoencoder's training progress is depicted in Figure 5.5, where we observe a rapid decrease in both training (blue) and validation (orange) loss within the initial epochs, indicating effective learning. As training continues, both loss metrics converge and stabilize, demonstrating the model's ability to generalize without overfitting. This trend of closely aligned and diminishing loss values suggests that the model has successfully captured the essential patterns in the traffic flow data, affirming its potential for accurate anomaly detection.

The Fig. 5.6 presents a scatter plot visualizing the reconstruction error across different classes of data points—normal and anomalous for training data. The x-axis represents the index of data points in the dataset, while the y-axis quantifies the reconstruction error. Each dot signifies an individual data point, with blue dots denoting normal data points and orange dots representing anomalies. It is evident that the majority of normal data points are clustered towards the lower end of the reconstruction error spectrum, suggesting accurate model predictions for these instances. Conversely, the anomalous data

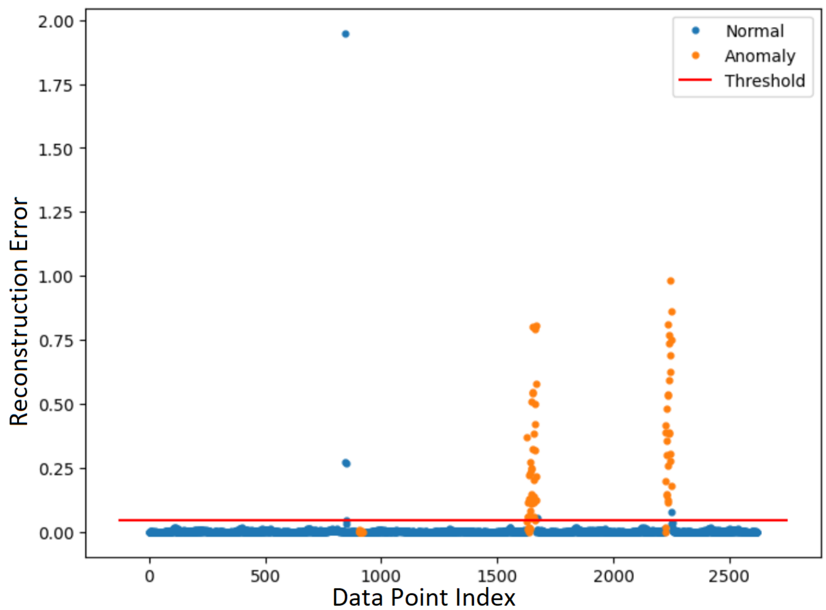


Figure 5.6: Graphical representation of reconstruction error across different classes in a time series dataset. The plot differentiates between normal (blue) and anomalous (orange) data points based on the calculated reconstruction error, with a defined threshold (red line) serving as the boundary for anomaly detection. Points above the threshold indicate potential anomalies within the data, showcasing the model's capability to discern irregular patterns from the normal operational flow.

points are dispersed, with many exhibiting significantly higher reconstruction errors, which aligns with the expectation that anomalies would be harder for the model to reconstruct faithfully, thus reflecting the model's ability to distinguish between normal and anomalous data effectively.

The graph in Fig. 5.7 depicts the trade-off between precision (blue line) and recall (orange line) for an anomaly detection model across a range of threshold values on the x-axis. As the threshold increases, precision initially holds steady or slightly improves, indicating a high accuracy of positive predictions, but then gradually declines, signaling a loss in the model's ability to detect true positives. Recall, however, consistently decreases with the threshold, reflecting a reduction in the model's sensitivity to capturing all relevant anomalies. This inverse relationship showcases the balancing act required to optimize the threshold for precise yet comprehensive anomaly detection.

The Fig. 5.8 shows a Receiver Operating Characteristic (ROC) curve, a graphical representation that illustrates the diagnostic ability of a binary classifier as its discrimination threshold is varied. The True Positive Rate (TPR), or sensitivity, is plotted on the y-axis against the False Positive Rate (FPR), or 1-specificity, on the x-axis. The blue line represents the ROC curve of the model, and the orange line represents a baseline (random chance) classifier. The Area Under the Curve (AUC) value is 0.870, as indicated in the legend, which suggests a good predictive performance. AUC values range from 0.5 (no better than random chance) to 1.0 (perfect classification), and an AUC of 0.870 indicates that the model has a high likelihood of distinguishing between the positive class and the negative class. The stepped

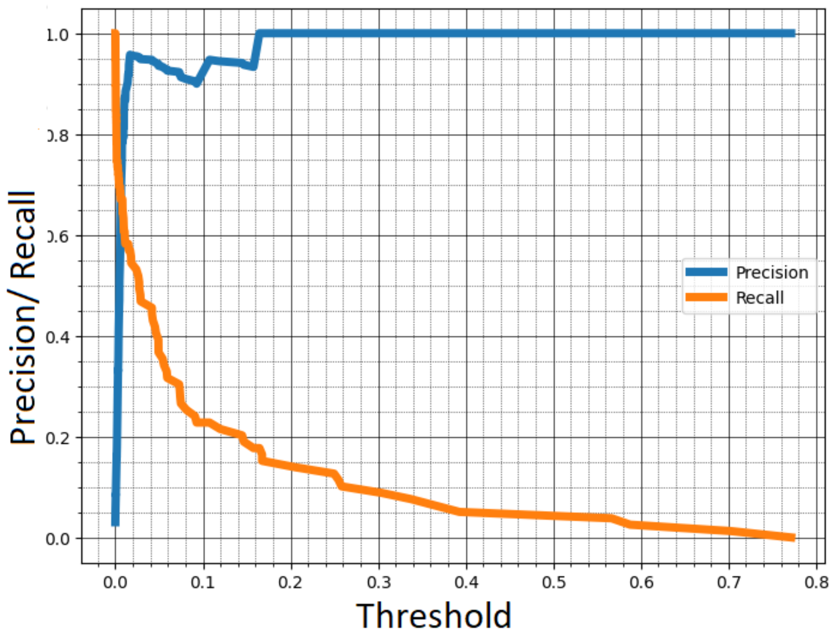


Figure 5.7: Trade-off analysis between precision and recall across varying threshold Values.

Table 5.2: Performance of Model on various stations.

Station	Precision	Recall	F1 score
Station 1	0.8854	0.7509	0.7905
Station 2	0.8333	0.7558	0.8278
Station 3	0.8940	0.7300	0.7804
Station 4	0.8780	0.7440	0.7737

nature of the curve suggests discrete changes in threshold values, commonly seen in models working with finite or binned score outputs.

The Table 5.2 systematically captures the precision, recall, and F1 score metrics of a classification model deployed across four distinct traffic stations. Notably, Station 2 exhibits superior performance with the highest precision and F1 score, indicative of the model's robust ability to accurately predict anomalies within its traffic patterns. This variance across stations underscores the model's differential response to the unique data characteristics inherent to each station, such as traffic volume, pattern complexity, and data quality. The metrics presented not only illuminate the model's strengths in minimizing false positives, thereby ensuring operational efficiency by reducing unnecessary traffic management interventions but also spotlight areas for enhancement, particularly in improving anomaly detection comprehensiveness.

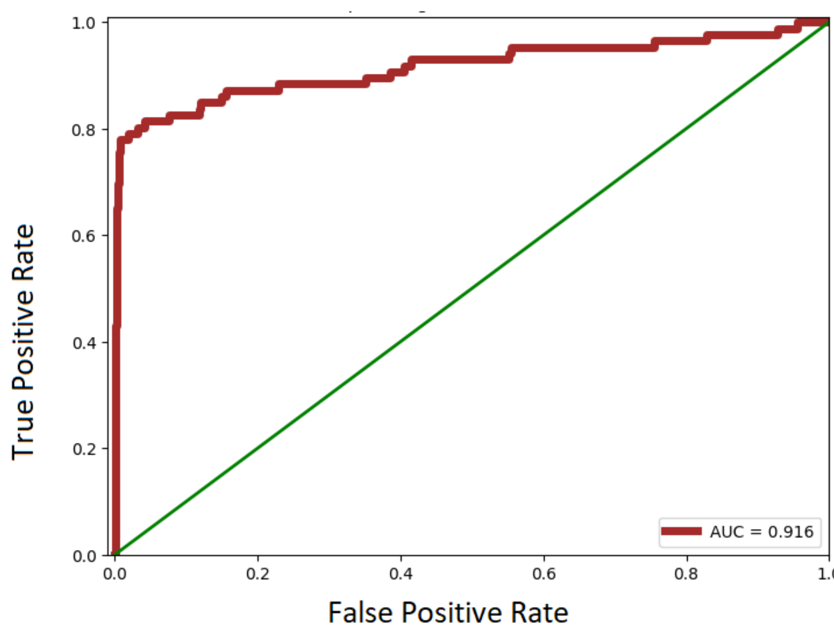
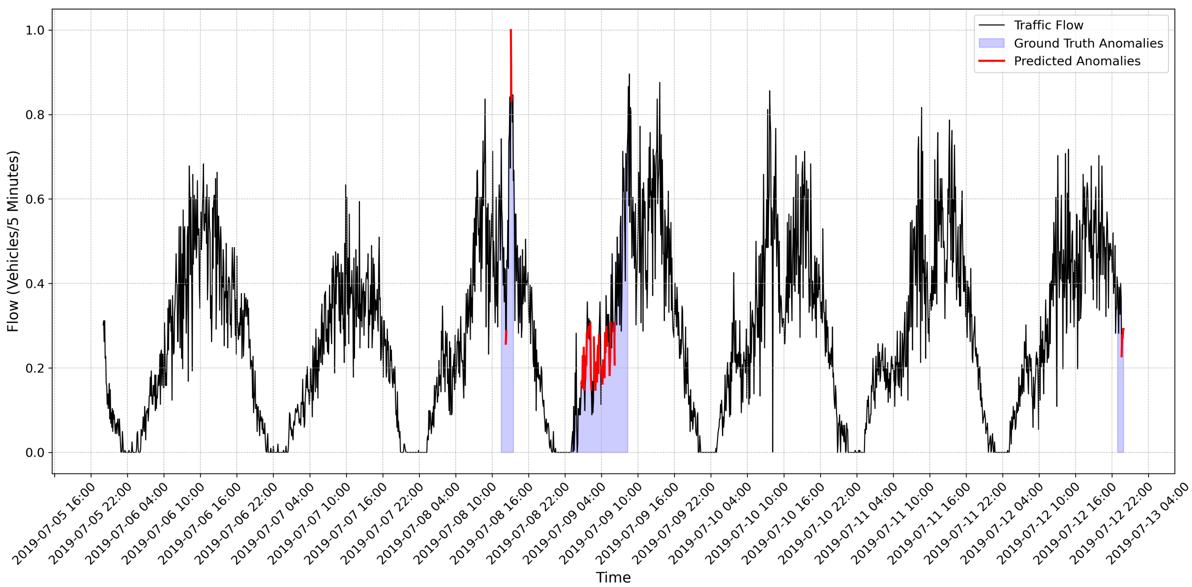


Figure 5.8: Receiver Operating Characteristic (ROC) curve demonstrating model performance in Anomaly Detection.

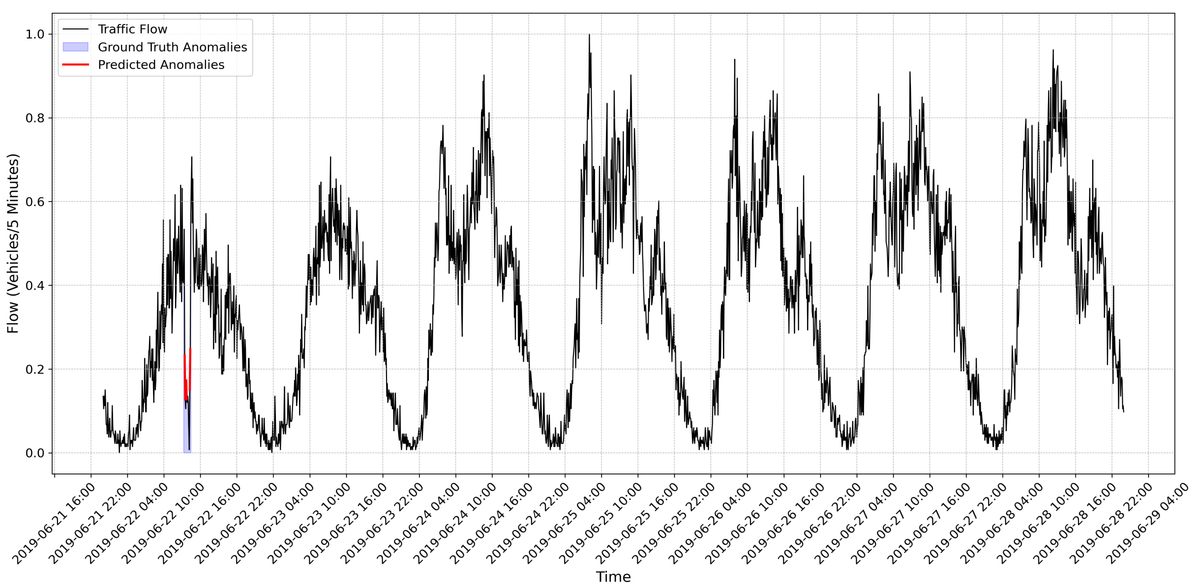
5.4.1 Performance on Real-time Data

This study presents a comprehensive evaluation of an anomaly detection system implemented on real-time traffic data, specifically from the PeMS database. The PeMS database, which stands as one of the most extensively utilized sources for real-time traffic data, provides comprehensive insights into traffic flow and conditions. In an innovative approach, our study has enriched this data by integrating it with incident reports from the California Highway Patrol (CHP), creating a novel dataset that marries traffic flow metrics with corresponding traffic-related incidents. This linkage, which has not been previously established, permits an unprecedented granular view of the impact of various incidents on traffic dynamics. The anomaly detection model is designed to identify irregular traffic patterns that deviate from the normative flow, which could signify incidents, roadwork, or other disruptions. The graph depicts the traffic flow (measured in vehicles per five minutes) plotted against time. The normal traffic flow is shown in black, while the ground truth anomalies are highlighted in blue, and the predicted anomalies are indicated in red. It is evident that the model can detect significant deviations from the typical traffic patterns.

The anomalies depicted in the graphs Fig. 5.9a and Fig. 5.9b, predominantly arise from instances of traffic hazards and collisions. These irregularities in traffic flow data are critical markers that can be leveraged to enhance road safety and traffic management. The present analysis encompasses the performance of the proposed anomaly detection model applied to the PeMS database, revealing not only the model's proficiency in identifying traffic spikes but also its sensitivity to subtler anomalies, such as diminished traffic flow, which may indicate significant roadway incidents.



(a) Visualization of peak traffic anomalies corresponding to collisions and hazards, with detected events emphasized in red against actual events in blue.



(b) The model's detection of diminished traffic flow anomalies, potentially indicative of incidents with no injuries or ongoing roadwork, demonstrating the system's sensitivity to subtle variations in traffic patterns.

Figure 5.9: Detailed Analysis of Traffic Anomaly Detection on Real-world dataset.

Table 5.3: Ablation Study: Comparative Performance Metrics for Different Architectural Configurations.

Architectural Configuration	**Precision**	**Recall**	**F1 Score**
Layer Units: 32-16	0.8354	0.7674	0.8000
Layer Units: 64-32	0.8333	0.7558	0.7927
Layer Units: 128-64	0.8219	0.6977	0.7547
Self-Attention Mechanism	0.8375	0.7791	0.8072
Multihead Attention Mechanism	0.8289	0.7326	0.7778
Contextual Attention Mechanism	0.8589	0.7826	0.8278

Fig. 5.9b extends the analysis, demonstrating the model's capability to detect anomalies characterized by a decrease in traffic flow. Such reductions are often less conspicuous yet may signify severe disruptions, like traffic collisions or road closures. The ability to detect lower-than-normal traffic volumes is integral to a comprehensive traffic monitoring system, as it may help in preemptively identifying hazardous conditions that are not apparent through traffic spikes alone. Together, these figures demonstrate the nuanced detection ability of our model, highlighting its utility in a practical traffic management setting.

5.5 Ablation Studies

The Table 5.3 presents a detailed ablation study that examines the impact of different architectural configurations and attention mechanisms on the performance of an anomaly detection model. Each row of the table represents a unique setup varying by the number of units in the input and middle layers or by the type of attention mechanism implemented. The configurations explored include simple layer unit adjustments with 32-16 (specifically, 32 units for layer 1 and 16 units for layer 2, and so forth), 64-32, and 128-64 units, respectively, and more complex attention-based approaches such as Self-Attention Mechanism, Multihead Attention Mechanism, and Contextual Attention Mechanism. The performance of each model configuration is evaluated based on three key metrics: Precision, Recall, and F1 Score, which are crucial for understanding the model's ability to accurately detect anomalies within the dataset.

The results depicted in the table reveal insightful trends about the model's performance across different settings. Layer configurations with a moderate number of units (32-16 and 64-32) tend to offer a balanced trade-off between precision and recall, resulting in relatively high F1 scores, indicating effective anomaly detection capabilities. In contrast, the model with a larger configuration (128-64) shows a slight dip in performance, suggesting that simply increasing the model's complexity does not necessarily translate to better detection accuracy. Interestingly, the attention mechanisms demonstrate distinct advantages, with the Contextual Attention Mechanism outperforming the others in terms of precision and F1 Score. This highlights the importance of contextual attention mechanisms in enhancing the model's sensitivity to patterns of anomalies, thereby improving the overall detection accuracy.

5.6 Conclusion

This study makes a significant contribution to traffic management systems, particularly in anomaly detection, by showcasing the effectiveness of LSTM Autoencoders enhanced with attention mechanisms. The evaluation of the model's performance through precision, recall, and F1 scores across various stations highlights its ability to differentiate between normal and anomalous traffic patterns with reliability. Furthermore, the model variant employing attention mechanisms demonstrates superior performance, indicating that a focus on the most significant sequence features substantially enhances anomaly detection accuracy. These results underscore the potential of deep learning architectures for managing and analyzing complex traffic flow data, suggesting a path toward more resilient and efficient traffic systems. Additionally, by injecting anomalies into real-world data like the widely used PeMS dataset, this approach allows for a more comprehensive testing and improvement of prediction models. Such integration of synthetic anomalies with real-time data not only enriches the dataset for model training but also simulates more diverse scenarios, providing a robust framework for refining the models to better predict and manage real-world traffic anomalies. This methodology promises to advance the development of traffic management solutions that are both more adaptive and effective, paving the way for their application in dynamic urban environments.

Chapter 6

Conclusion and Future scope

In conclusion, our proposed methodology represents a unique and pioneering approach to addressing the multifaceted challenges within intelligent transportation systems (ITS). By seamlessly integrating edge computing with sophisticated deep learning techniques, we offer a novel solution that transcends traditional limitations and unlocks new possibilities for enhancing urban mobility.

The deployment of LSTM and Bi-LSTM networks directly on edge devices like Raspberry Pi marks a significant departure from conventional ITS architectures. This novel approach enables real-time analysis and decision-making capabilities at the edge, a feat previously unattainable. By harnessing the power of these advanced neural networks in situ, we empower vehicles and infrastructure elements to process sensor data autonomously, leading to unprecedented levels of responsiveness and adaptability. Moreover, our optimization techniques, particularly quantization methods, represent a distinctive aspect of our methodology. By tailoring model architectures and reducing computational complexity, we ensure that our deep learning models are not only accurate but also efficient, even on resource-constrained edge devices. This optimization strategy is a testament to our commitment to practicality and scalability, ensuring that our proposed solution is not just theoretically sound but also eminently deployable in real-world settings.

The transformative potential of our methodology extends beyond mere technological innovation. It promises to redefine the very fabric of urban transportation by offering tangible solutions to pressing challenges. From enhancing two-wheeler safety through advanced anomaly detection to optimizing traffic flow management with unprecedented precision, our approach paves the way for safer, more efficient, and ultimately more sustainable transportation networks. In essence, our proposed methodology represents a paradigm shift in the field of intelligent transportation. It embodies the spirit of innovation and collaboration, bringing together cutting-edge technologies to tackle some of the most pressing issues facing modern cities. As we look towards the future, our approach offers a beacon of hope for creating smarter, more resilient, and more inclusive urban environments for generations to come.

Future initiatives should aim to enhance the scalability of edge computing implementations within expansive urban networks and to explore the integration of advanced neural network architectures like Transformers for more sophisticated data analysis. Developing adaptive computing solutions will be crucial for optimizing computational resources dynamically, improving both the energy efficiency and operational efficacy of our systems. Moreover, extending the scope of our anomaly detection systems to encompass multimodal transportation and integrating predictive analytics will refine our traffic management capabilities, allowing for anticipatory rather than merely responsive strategies.

Finally, efforts to formulate comprehensive integration frameworks for smart cities—incorporating regulatory support and fostering public-private partnerships—will ensure the practical deployment and sustained effectiveness of these advanced systems. These advancements will address existing challenges and open new avenues for enhancing urban mobility and safety.

Related Publications

Conference Paper (First Author):

- S. U. Goparaju et al., ”Time Series-based Driving Event Recognition for Two Wheelers,” 2023 Design, Automation Test in Europe Conference Exhibition (DATE), Antwerp, Belgium, 2023, pp. 1-2, doi: 10.23919/DATE56975.2023.10136944.
- S. U. Goparaju, K. Pothalaraju, S. Dullur, A. Jain and D. Gangadharan, ”Time-Series based Fall Detection in Two-Wheelers,” 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), Hong Kong, Hong Kong, 2023, pp. 1-5, doi: 10.1109/VTC2023-Fall60731.2023.10333464.
- Sai Usha Goparaju and Deepak Gangadharan. ”Enhanced Two-Wheeler Safety leveraging Isolation Forest for Pre-Impact Fall Prediction,” IEEE Intelligent Vehicles Symposium (IV), 2024. *(under review)*.

Journal (First Author):

- Sai Usha Goparaju, Rahul Biju, Deepak Gangadharan, and Bapaditya Manda. ”Contextual Attention Information based Encoder-Decoder Network for Anomaly Detection in Traffic Flow Analysis,” IEEE Transactions on Intelligent Transportation Systems *(under review)*).

Publications unrelated to the thesis:

- S. U. Goparaju et al., ”Optimization and Performance Evaluation of Hybrid Deep Learning Models for Traffic Flow Prediction,” 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), Florence, Italy, 2023, pp. 1-7, doi: 10.1109/VTC2023-Spring57618.2023.10200600.
- S. U. Goparaju, S. S. S. Vaddhiparthy, C. Pradeep, A. Vattem and D. Gangadharan, ”Design of an IoT System for Machine Learning Calibrated TDS Measurement in Smart Campus,” 2021 IEEE 7th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2021, pp. 877-882, doi: 10.1109/WF-IoT51360.2021.9595057.

Bibliography

[1] W. H. Organization, “Road safety is no accident : a brochure for world health day 7 april 2004,” p. WHO/NMH/VIP/03.4, 2004.

[2] Jain *et al.*, “Two wheeler accidents on indian roads - a study from mangalore, india,” *Journal of forensic and legal medicine*, vol. 16, pp. 130–3, 05 2009.

[3] S. U. Nagasri Goparaju, L. Lakshmanan, A. N, R. B, L. B, D. Gangadharan, and A. M. Hussain, “Time series-based driving event recognition for two wheelers,” in *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2023, pp. 1–2.

[4] “Report,” https://morth.nic.in/sites/default/files/RA_Uploading.pdf.

[5] M. Pieve, F. Tesauri, and A. Spadoni, “Mitigation accident risk in powered two wheelers domain: Improving effectiveness of human machine interface collision avoidance system in two wheelers,” in *2009 2nd Conference on Human System Interactions*, May 2009, pp. 603–607.

[6] L. Brown *et al.*, “Investigation of accidents involving powered two wheelers and bicycles – a european in-depth study,” *Journal of Safety Research*, vol. 76, pp. 135–145, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022437520301651

[7] Naqvi *et al.*, “Factors contributing to motorcycle fatal crashes on national highways in india,” *Transportation Research Procedia*, vol. 25, pp. 2089–2102, 12 2017.

[8] World Health Organization, “Global status report on road safety 2018,” https://www.who.int/publications/i/item/9789241565684, 2018, accessed: date.

[9] Environmental Protection Agency, “Inventory of u.s. greenhouse gas emissions and sinks,” https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks, 2019, accessed: date.

[10] Federal Highway Administration, “The benefits of reducing congestion,” https://www.fhwa.dot.gov, Year, accessed: date.

[11] M. Buda, A. Maki, and M. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018.

[12] U.S. Department of Transportation, "The cost of congestion to the economy," https://www.transportation.gov, Year, accessed: date.

[13] G. Zhang, B. Patuwo, and M. Hu, "Arima based network traffic modeling and prediction," *Neural Networks for Signal Processing - Proceedings of the IEEE Workshop*, pp. Vol. Number, Page numbers, 2005.

[14] U. Rebbapragada, P. Protopapas, C. Brodley, and C. Alcock, "Finding anomalous periodic time series," *Machine Learning*, vol. 74, no. 3, pp. 281–313, 2009.

[15] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of Data Mining in Computer Security*. Kluwer Academic Publishers, 2002, pp. 77–101.

[16] Bouaouni *et al.*, "Driving-pattern identification and event detection based on an unsupervised learning framework: Case of a motorcycle-riding simulator," *IEEE Access*, vol. 9, pp. 158 456–158 469, 2021.

[17] Sarker *et al.*, "Driving maneuver classification using domain specific knowledge and transfer learning," *IEEE Access*, vol. 9, pp. 86 590–86 606, 2021.

[18] Johnson *et al.*, "Driving style recognition using a smartphone as a sensor platform," in *ITSC*.

[19] Attal *et al.*, "Powered two-wheeler riding pattern recognition using a machine-learning framework," *IEEE Transactions on ITS*, vol. 16, no. 1, pp. 475–487, 2015.

[20] Mitrovic *et al.*, "Reliable method for driving events recognition," *IEEE Transactions on Intelligent Transportation Systems*.

[21] Pienaar *et al.*, "Human activity recognition using lstm-rnn deep neural network architecture," in *Wireless Africa Conference*, 2019.

[22] Selmanaj *et al.*, "Hazard detection for motorcycles via accelerometers: A self-organizing map approach," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3609–3620, 2017.

[23] F. Attal, A. Boubezoul, A. Samé, L. Oukhellou, and S. Espié, "Powered two-wheelers critical events detection and recognition using data-driven approaches," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[24] L. Tong, Q. Song, Y. Ge, and M. Liu, "Hmm-based human fall detection and prediction method using tri-axial accelerometer," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1849–1856, 2013.

[25] S. Shan and T. Yuan, "A wearable pre-impact fall detector using feature selection and support vector machine," 11 2010, pp. 1686 – 1689.

[26] Jo *et al.*, “A study on the application of lstm to judge bike accidents for inflating wearable airbags,” *Sensors*, vol. 21, no. 19, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6541

[27] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: Forecasting and control*. John Wiley Sons, 2015.

[28] B. Rosner, “Percentage points for a generalized esd many-outlier procedure,” *Technometrics*, vol. 25, no. 2, pp. 165–172, 1983.

[29] W. Lu and A. A. Ghorbani, “Network anomaly detection based on wavelet analysis,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, p. 837601, 2008.

[30] Y. Zhang, Z. Ge, J. Green, and D. C. Wunsch, II, “Network anomaly detection: A machine learning perspective,” *unknown*, 2005.

[31] K. Yamanishi, J.-i. Takeuchi, G. Williams, and P. Milne, “Online unsupervised outlier detection using finite mixtures with discounting learning algorithms,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 320–324.

[32] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, “Anomaly detection in streams with extreme value theory,” *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1067–1075, 2017.

[33] R. Yu, H. Zheng, A. Anandkumar, and Y. Yue, “Glad: Group anomaly detection in social media analysis,” *unknown*, 2014.

[34] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, 2018.

[35] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, “Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. ACM, 2018, pp. 387–395.

[36] M. Munir, S. A. A. Siddiqui, A. Dengel, and S. Ahmed, “Deepant: A deep learning approach for unsupervised anomaly detection in time series,” *IEEE Access*, vol. 7, pp. 1991–2005, 2018.

[37] P. Darban and A. Another, “Reconstruction-based anomaly detection for time series data,” 20XX, details or journal/booktitle missing because it’s a placeholder.

[38] M. D *et al.*, “The road ahead – traffic injuries and fatalities in india. tripp transportation research and injury prevention programme,” 2004.

[39] Sagberg *et al.*, "A review of research on driving styles and road safety," *Human factors*, vol. 57, 06 2015.

[40] Ferreira *et al.*, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLOS ONE*, vol. 12, pp. 1–16, 04 2017.

[41] Nuswantoro *et al.*, "Abnormal driving detection based on accelerometer and gyroscope sensor on smartphone using ann algorithm," in *IES*, 2020.

[42] Karim *et al.*, "Lstm fully convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018.

[43] Li *et al.*, "Bi-lstm network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1191–1201, 2020.

[44] Vaswani *et al.*, "Attention is all you need." arXiv, 2017, https://arxiv.org/abs/1706.03762.

[45] S. Kumar, B. Sharma, and M. I. Nezhurina, "Responsible factors of powered two wheeler accidents: A review," in *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)*, 2018, pp. 1–6.

[46] "BikeSim," https://www.carsim.com/products/bikesim/.

[47] "Report," https://morth.nic.in/sites/default/files/RA_2022_30_Oct.pdf.

[48] A. Meena, S. Iyer, M. Nimje, S. Joglekar, S. Jagtap, and M. Rahman, "Automatic accident detection and reporting framework for two wheelers," in *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*, 2014, pp. 962–967.

[49] N. Kattukkaran, A. George, and T. M. Haridas, "Intelligent accident detection and alert system for emergency medical assistance," in *2017 International Conference on Computer Communication and Informatics (ICCCI)*, 2017, pp. 1–6.

[50] S. Gelmini, S. Strada, M. Tanelli, S. Savaresi, and C. D. Tommasi, "Automatic crash detection system for two-wheeled vehicles: design and experimental validation," *IFAC-PapersOnLine*, 2019, 9th IFAC Symposium on Advances in Automotive Control AAC 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319307025

[51] S. U. Goparaju, K. Pothalaraju, S. Dullur, A. Jain, and D. Gangadharan, "Time-series based fall detection in two-wheelers," in *2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023, pp. 1–5.

[52] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[53] G. Pang, C. Shen, L. Cao, and A. v. d. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, p. Article 38, 2021.

[54] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[55] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.