UNIVERSITY OF CATANIA

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING

PhD IN SYSTEM, ENERGY, COMPUTER AND TELECOMMUNICATIONS ENGINEERING

XXXVI CYCLE

Road Safety: Abnormal Driver Behavior Detection through Deep Learning

SHUMAYLA YAQOOB

Coordinator: Prof. Paolo Pietro Arena

Tutor: Prof. Salvatore Damiano Cafiso

Prof. Giacomo Morabito

Company Tutor: Xenia Network Solutions

Contents

List of Terms and Abbreviations

AI	Artificial Intelligence
NMEA	National Marine Electronics Association
GNSS	Global Navigation Satellite System
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Network
TCs	Traffic Conflicts
VVL	Video Vbox Lit
SGF	Svitzky Golay Filter
GPS	Global Positioning System
ANN	Artificial Neural Network
AE	Autoencoder
CM	Confusion Matrix
ML	Machine Learning
PCA	Principal Component Analysis
GIS	Geographic Information System
CSE	Critical Safety Event
RNN	Recurrent Neural Network

List of Figures

List of Tables

ABSTRACT

Road safety is a pressing concern, with the number of road accidents and bicycle-related fatalities on the rise. Both driver and cycling behavior play crucial roles in road safety. Therefore, understanding and detecting abnormal behaviors in these contexts are of paramount importance to reduce accidents and enhance road safety. Detecting abnormal driving behavior has become a significant issue. The first part of this research focuses on safe driving. Driver behaviors such as drowsy, aggressive, and distracted driving, significantly contribute to road crashes. To address this, extensive research has been conducted to monitor and model driver behavior. Initially we critically review the existing literature, categorizing approaches into traditional mathematical, machine learning, and deep learning-based schemes. It provides a comparison table and taxonomy based on various metrics and highlights open research questions for future exploration.

Turning to cycling, an increasingly popular and sustainable mode of transportation, safety remains a challenge. With the growing number of cyclists, the availability of suitable crash data becomes more complex. Smart cities and new technologies offer opportunities for data collection and analysis. This research introduces the "BeST-DAD" model, utilizing deep learning techniques like Convolutional Neural Networks and Autoencoders for anomaly detection in cycling behavior. Results show the model outperforms traditional statistical approaches, achieving a 77% F-score and 100% recall. The next activity of this research for cycling safety is to employ deep transfer learning to proactively detect anomalies in cycling behavior, which could lead to traffic conflicts or near-miss accidents. The study introduces a customized model, "DTL_AD," tailored to individual riders. Data collected using Global Navigation Satellite System (GNSS) instruments on bicycles is used to identify riding anomalies. This innovative approach holds promise for enhancing cycling safety in urban environments. Furthermore, It helps to reduce the extensive requirements for data labeling and model training.

In addition, the study explores the role of convolutional layers in deep neural networks for scenarios involving user-environment interactions. It investigates whether these layers are more specific to the user or the environment, aiming to streamline data collection and model training efforts in such contexts.

To address abnormal behavior in vehicular contexts, a deep learning model centered on convolutional autoencoders is presented. It utilizes vehicle data, including speed, acceleration, and heading, to identify irregular behavior. The model's performance is compared to established machine learning methods for anomaly detection. Furthermore, the research delves into multi-sensor fusion, combining data from GPS, OBD, and Mobileye sensors sourced from vehicles. The objective is to identify the most effective sensor combinations for detecting abnormal driver behavior, benchmarking against other anomaly detection algorithms.

To validate the proposed methodology's real-world effectiveness, a case study visually depicts anomalies in cycling behavior using Geographic Information Systems (GIS) maps. The clustering of data in high-risk areas is emphasized, showcasing practical applications in enhancing road safety within cities, as demonstrated in Catania, Italy. This comprehensive research contributes to improving road and cycling safety.

Keywords: *Abnormal Driving*, *Anomaly*, *Convolutional Autoencoder*, *Deep Learning*, *Transfer Learning.*

Chapter 1

Introduction

With the high numbers of injured people and fatalities in road crashes, safe driving is a serious and challenging concern and a prolific research area, and this contact driver behavior has a great influence on road safety [1]. In fact, driver behaviors such as drowsy driving, aggressive driving, distracted driving, and safe driving flairs, may lead to road crashes which are the cause of massive human and material losses annually both in developed and mainly in developing countries [2]. Therefore, in the new smart society, in-vehicle monitoring of drivers and detecting abnormal driving behavior as anomalies can reduce the rate of road crashes. A huge amount of research has been done to detect on–road driver behavior by monitoring driver health or vehicle operating conditions such as speed, acceleration, etc.

Driver behavior encompasses more than just those operating cars or buses; it also applies to cyclists, as they are also considered drivers on the road. This emphasis highlights the significance of comprehending and examining the conduct of every individual using the road, contributing to a holistic approach to safety and traffic control.

Cycling is recognized as an eco-friendly and healthy means of transportation, especially pertinent in light of escalating concerns about greenhouse gas emissions and pollution [3]. Policymakers are increasingly inclined to promote cycling as a sustainable commuting option. The COVID-19 pandemic further underscored cycling's value as a personal mobility choice. However, the safety of bicyclists has become a significant challenge, particularly with the rising number of cyclists in the 21st century. In contrast to traditional road safety data collection methods, gathering appropriate data related to bicycle-related accidents is notably more complex. As a result, the emergence of smart city technologies offers new avenues for data collection and analysis, potentially addressing these challenges in innovative ways.

Despite the general improvements in road safety, with the growing number of bicycle users, cycling safety is still a challenge as demonstrated by the fact that it is the only road transport mode with an increase in the number of fatalities in EU cities. Traditional approaches relying on crash statistics for network screening are reactive and less effective, largely due to the inadequacy of available bicycle-related data.

Therefore Abnormal driver behavior on the road poses significant risks to safety, leading to accidents and road-related incidents. Identifying and mitigating these behaviors is crucial for enhancing road safety [4][5]. This problem formulation aims to address the detection of abnormal driver behavior using advanced technologies and data-driven approaches. The challenge at hand is to develop an efficient and accurate system for the real-time detection of abnormal driver behavior, encompassing actions like aggressive driving, distracted behavior, and reckless maneuvers. This necessitates the integration of diverse data sources, including in-vehicle sensors, telematics devices, cameras, and external data, to create algorithms and machine learning models capable of distinguishing abnormal behavior from regular driving patterns.

This thesis work aims to analyze aspects related to road safety in the context of abnormal driver behavior. In particular, the main objectives of this thesis are:

- critically reviews the existing literature on driver behavior modeling and abnormal behavior detection-based approaches for safe driving. More specifically, existing approaches are classified in a coherent taxonomy by reviewing traditional mathematical, machine, and deep learning-based schemes.

- a deep learning-based approach "BeST-DAD" to detect anomalies and spot dangerous points on the map for bicyclists to avoid a critical safety event (CSE). BeST-DAD follows a Convolutional Neural Network and Autoencoder (AE) for anomaly detection. BeST-DAD performs better than traditional PCA statistical approaches for anomaly detection by achieving 77% of the F-score.

- a deep transfer learning model to detect anomalies in cycling behavior that can be associated with traffic conflicts or near-miss crashes. The paper presents how to build a users' tailored riding model named DTL_AD to detect and localize riding anomalies by using a set of data in the National Marine Electronics Association (NMEA) string of Global Navigation Satellite System (GNSS) recorded with instrumented bicycles by different cyclists.

- A case study demonstrates the identification of anomalies in cycling behavior visually represented on Geographic Information Systems (GIS) maps, showing how data clustering is well located in high-risk areas.

- investigate the role of convolutional layers in deep neural networks for application scenarios involving interactions between users and the environment.

- Exploit Deep Transfer Learning for Anomaly Detection.

- Reduce the need for data labeling and model training.

- ROAD-DAD, a deep learning model. It integrates convolutional autoencoder and a self-directed algorithm for anomaly detection, using car data like speed, acceleration, and heading. The model's performance is validated by comparing it with established machine learning methods for anomaly detection to ensure its effectiveness.

- integrate data from multiple vehicle sensors, including GPS, OBD, and Mobileye. Our goal is to utilize the ROAD-DAD model for anomaly detection, experimenting with different sensor combinations to identify the most effective configuration for detecting abnormal driver behavior. After identifying the best sensor combination, we assess the results by comparing them with those generated by alternative anomaly detection and classification algorithms.

- Use-case based on real data collected with high-frequency GNSS.

- Application of the scheme is presented by plotting detected anomalies on a map in order to identify dangerous locations in the city of Catania Italy.

The rest of the thesis is organized as follows: Chapter II outlines the major contributions of the thesis in detail. Chapter III provides an in-depth discussion of safe driving components, the underlying motivation for this research, and an examination of the existing landscape in terms of abnormal driving behavior schemes, including their taxonomy. Chapter IV delves deeply into the first use case, which centers on bicycles, exploring the unique challenges associated with bicycle transportation and presenting proposed solutions. Additionally, it addresses three distinct tasks connected to anomaly detection, leveraging transfer learning, and the role of convolutional layers concerning user and environment interaction. Chapter V offers a comprehensive examination of the second use case, focusing on cars, where abnormal driving behavior detection and multi-sensor fusion analysis are thoroughly explored. Lastly, Chapter VI serves as the conclusion of the thesis, summarizing key findings and suggesting potential avenues for future research.

Every chapter has its own specific objective, background, methodology, results, and remarks.

The thesis structure is visually depicted in Figure 1.1 to facilitate comprehension. This visual aid is included to help readers better understand the flow and sequence of topics, chapters, or sections within the thesis. It serves as a road map to navigate through the content effectively.

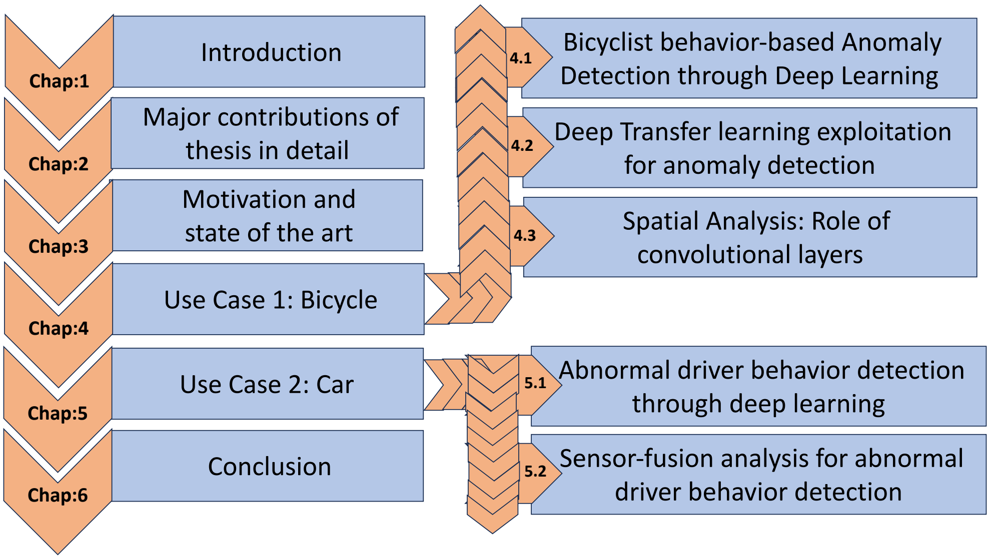


Figure 1.1: Thesis flow diagram

Chapter 2

Major contributions of the thesis in detail

The proposed methodology in this thesis work focuses on enhancing road safety by addressing abnormal driver behavior. Road safety typically involves minimizing accidents, injuries, and fatalities associated with various modes of transportation, such as driving, cycling, or walking. To improve or make safer the conditions on roads, abnormal driver behavior needs to be addressed in time. The methodology's specific focus is on understanding and dealing with abnormal driver behavior. Abnormal driver behavior refers to actions or patterns of behavior exhibited by drivers that deviate from what is considered normal or safe. Examples might include aggressive driving, impaired driving, drowsy driving, and distracted driving.

This chapter indicates that the thesis work has several key goals or objectives, and these objectives are the central components of the research. The objectives serve as a roadmap for what the research aims to achieve. In summary, these lines introduce the reader to the overarching purpose of the thesis work, which is to improve road safety by addressing abnormal driver behavior.

The overview of the proposed methodology includes:

1. **Comprehensive Survey:** A comprehensive review of existing literature is conducted to categorize and evaluate various driver behavior models and abnormal behavior detection approaches including mathematical, machine learning, and deep learning-based methods. Next, the survey is structured and organized into a hierarchical classification system, commonly referred to as taxonomy. In Chapter 2, a thorough examination of the motivation and the current state of the field is presented in detail. Additionally, visual

representations in the form of graphics or images are included to enhance understanding.

2. **BeST-DAD Model:** The thesis introduces a sophisticated deep learning model known as "BeST-DAD." This model employs advanced techniques, including Convolutional Neural Networks (CNN) and Autoencoder (AE), to identify abnormal patterns and pinpoint high-risk locations on maps, with a specific focus on enhancing safety for cyclists.

   Convolutional Neural Networks (CNNs) are a class of deep learning algorithms designed for processing grid-like data, such as images or spatial data. CNNs are known for their ability to automatically learn and extract relevant features from data through a series of convolutional layers. In the context of BeST-DAD, CNNs are instrumental in recognizing intricate spatial patterns and relationships within the geographic data, aiding in the identification of potential safety risks for cyclists.

   Additionally, BeST-DAD utilizes Convolutional Autoencoders (CAEs). Autoencoders are neural network architectures employed for unsupervised feature learning and data compression. CAEs are a variant of autoencoders that incorporate convolutional layers, making them especially well-suited for image and spatial data. CAEs excel at dimensionality reduction and feature extraction, which are crucial for identifying anomalies in complex geographical data.

   It's noteworthy that BeST-DAD's utilization of CNN and CAE technologies enables it to outperform traditional statistical approaches like Principal Component Analysis (PCA). Specifically, it achieves an impressive F-score of 77%, demonstrating its efficacy in enhancing safety for bicyclists by identifying and mitigating potential risks on the road.

   You can find a more comprehensive explanation of this aspect of the proposed methodology in Chapter 4.1, complete with an accompanying model diagram.

3. **DTL-AD Model:**

   Another aspect of the proposed methodology involves the incorporation of a novel deep transfer learning model known as "DTL-AD." The methodology leverages the capabilities of deep transfer learning to detect anomalies, aiming to reduce the extensive requirements for data labeling and model training. This objective is realized through the implementation of the specialized DTL-AD model, which is meticulously designed to identify deviations in cycling behavior that could potentially result in traffic conflicts or near-miss incidents. DTL-AD relies on data derived from the National Marine Electronics Association (NMEA) strings of the Global Navigation Satellite System (GNSS), which have been collected from bicycles equipped with specialized instrumentation and utilized by a diverse group of cyclists.

For a more detailed insight into this component of the proposed methodology, please refer to Chapter 4.2, which includes accompanying illustrations and charts for enhanced clarity.

4. **Convolutional Layers Investigation:** The research explores the role of convolutional layers in deep neural networks, particularly in scenarios involving user-environment interactions, where user-environment interactions play a crucial role in road safety.

   If you seek a more comprehensive understanding of this aspect within the proposed methodology, I encourage you to consult Chapter 4.3. This section is enriched with accompanying visuals and graphs to improve clarity and facilitate a deeper grasp of the topic.

5. **ROAD-DAD:** This phase of the proposed methodology represents a deep learning model that centers on the principles of convolutional autoencoders, coupled with a self-directed algorithm designed for the detection of anomalies. This model harnesses data derived from vehicles (cars), including information such as speed, acceleration, and heading, among others, to identify irregular behavior or anomalies. To ensure the effectiveness of this approach, the model's performance is validated by benchmarking it against established machine learning methods commonly used for anomaly detection.

   If you are looking to gain a more thorough comprehension of this particular facet within the proposed methodology, I strongly recommend referring to Chapter 5.1. This section has been enhanced with the inclusion of visual aids and graphs to enhance clarity and facilitate a deeper understanding of the subject matter.

6. **Multi-sensor Fusion:**In this aspect of the proposed methodology, we engage in the fusion of data from multiple sensors, all sourced from the vehicle itself. These sensors include GPS, OBD, and Mobileye. The objective here is to employ the ROAD-DAD model for the purpose of anomaly detection, making use of various combinations of sensor data. Our aim is to determine

which combination of sensors yields the most effective results for detecting abnormal driver behavior. Once we identify the optimal sensor combination, we proceed to evaluate the outcomes by comparing them with the results generated by other anomaly detection and classification algorithms.

It's important to note that there are instances when abnormal behavior, as identified by anomaly detection algorithms, may not truly constitute an anomaly. Instead, these behaviors could be attributed to other common road maneuvers such as overtaking, responses to traffic lights, or reactions to road construction. Consequently, we group these behaviors into a distinct category and have achieved favorable outcomes through classification algorithms.

For a more comprehensive understanding of this particular aspect within the proposed methodology, I encourage you to consult Chapter 5.2.

7. **Practical Application:** The proposed methodology is put to practical use through a case study that involves visually depicting anomalies in cycling behavior using Geographic Information Systems (GIS) maps. This visualization effectively emphasizes the clustering of data in high-risk areas. Additionally, the research demonstrates the practicality of the approach by applying it to the city of Catania, Italy, where detected anomalies are mapped to pinpoint hazardous locations. This showcases the methodology's real-world effectiveness in enhancing road safety within the city.

Chapter 3

Motivation and State Of The Art

3.1 Objective

The worldwide rate of accidents is high due to careless driving behaviors [6][7][8]. Real-time driver behavior identification and warning are very important to reduce the rate of road accidents. This is an essential issue as around 1.25 million people lose their lives every year as a result of road accidents [9]. Many more are injured and the cost to society is huge. Indeed, road safety is still an extreme challenge for societies [10][11][1]. Even in the last 20 years, novel technologies have been developed in a progressive effort to manufacture automated vehicles, driver behavior is still a key factor for road safety [11][1][12]. Although road accidents are not the result of a unique factor, including driver, road, and vehicle, generally driver's health or behavior plays a relevant role. Driver's behavior can be affected in several and extremely different cases. For example, bad physical and mental health conditions of the driver may lead to distraction that results in accidents [1][13][14]. Also, some secondary tasks may distract the driver's attention and affect driving performance directly [15]. As a result, nowadays, the analysis of driver behavior is a prolific research area [16][17] and many researchers have been working to detect dangerous driver behaviors and develop techniques. Recently numerous machine learning and deep learning-based approaches have been proposed for the detection of anomalies in driver behavior, aimed at improving road safety [17][18]. Indeed, for a decade research has been in progress which focuses on driver behavior monitoring schemes [19][20]. Health monitoring includes psychological, physiological, and neurological measures that may lead to assessing various abnormal driver behaviors [20]. Abnormal driver behaviors include drowsy, distracted, fatigued, and aggressive behaviors [21]. All of them may lead to road accidents [22][23][24] as shown in Figure 3.1.

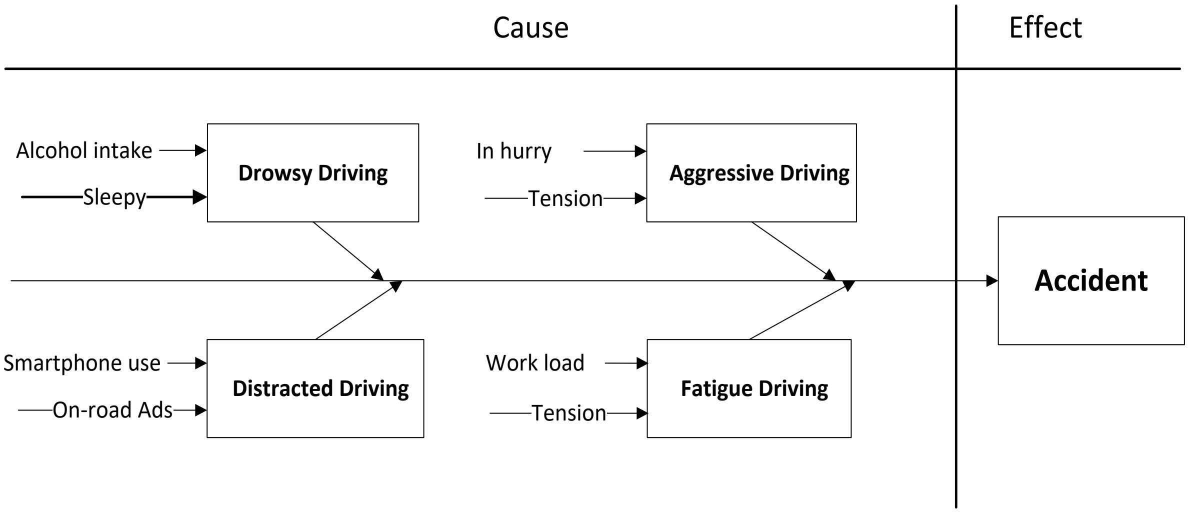


Figure 3.1: Cause and Effect of abnormal driving behaviors

Drowsy, fatigue, and aggressive driving depend on the driver's physical or mental health conditions [25][26]. Drowsy driving can be the side effect of drugs and psychoactive substances (alcohol, narcotic materials) or a sleepy state. Sudden lane change or unintended over-cross leads to aggressive driving that can be the result of tension or in-hurry conditions [27][28]. Distracted driving may be the result of diverted attention by reading any roadside advertisement or using a smartphone [29]. Fatigue driving is possibly the result of high workload or personal tension. Mostly driver drowsiness can be inferred from the driver's facial expressions [30]. Initially, researchers proposed the use of sensors available in smartphones, especially cameras to collect images used to infer the current health and operation conditions of the driver [31]. Later, in-vehicle sensors have been used to collect data in the form of time series. In fact, according to previous studies such time series can be effectively used to predict future vehicle position and driver behavior. In-vehicle sensors are often coupled with the Global Navigation Satellite System (GNSS) to measure driving practices [20]. Given the relevance of the topic, several survey papers have been published on the analysis of driving behaviors. Existing survey papers consider any one of the driver behavior conditions and explore it either concerning the driver's health or vehicle monitoring. Some of these simply analyze driver behaviors to highlight which one is most dangerous [32]. In [14], authors explain various driver fatigue detection schemes. In [8], authors discussed distraction detection methods to avoid road fatalities. In [33], authors reviewed literature stated possible driving styles, and proposed artificial intelligence (AI) based algorithms. In [34], authors discussed various driver behavior detection-based schemes and classified them into real and not-real-time schemes. In [35], the authors highlight driver monitoring styles and their related states at different levels of driving automation. This paper presents a comprehensive review of the techniques proposed so far for driver behavior detection including health monitoring. The search criteria of this review follow databases in Scopus, Web of Science, and IEEE Explore. The period of selected articles was between 8 and 10 last year (2013 – present). The keywords are used to search related work as shown in Table 3.4. By using keywords, we identified 100 papers while we opted for 67 articles after reading abstracts. After the literature and analysis, several issues are identified from the schemes that can be the focus area for researchers in the future. A few directions are proposed to disseminate driver's behavior warnings through the Internet of Vehicles (IoVs). The main contributions of this work are the following.

- We explore the existing work on driver's behavior modeling by presenting an ontology then we explore the effect of abnormal driving behavior on road safety.
- Furthermore, we dig into the literature for abnormal driver behavior detection where the focus is on analyzing the driver's health and vehicle parameters along with the driving behavior. We also consider AI-based solutions to deduce the linkage of driving disorders with the driver's health and vehicle's conditions.
- We survey the main contributions of the existing surveys regarding this field and identify the main research gaps.
- We present the taxonomy of the schemes for driver's behavior detection by considering driver's health and driving patterns through vehicle monitoring with AI-based algorithms.
- We analytically discuss the performance of existing schemes and then identify the open research issues for future work.

3.2 Preliminaries

In this section, we overview the preliminaries such as various driver behaviors and possible parameters to identify each of them with the help of an ontology diagram (Figure. 3.2). Then, how abnormal driver behavior causes accidents in real life. Finally, abnormal driver behavior identification architecture is discussed to understand the meaning and path of abnormal driver behavior identification.

3.2.1 Driver behaviors ontology

Driver behavior plays a vital role in the field of road safety [1][12]. Mainly driver behavior is categorized as normal and abnormal. Normal driving behavior refers to safe driving while abnormal driving behaviors are (i) drowsy, (ii) aggressive, (iii) fatigue, and (iv) distracted driving that causes physical and financial loss [36][37]. Physical losses may include minor to severe injuries or even death. Financial losses may include materialistic items, loss of passengers, and damaged vehicles. From the literature, we have identified that the driving behaviors can be classified as shown in the ontology shown in Figure. 3.2. Existing abnormal driver behavior detection systems are either based on driver's health or vehicle monitoring. Each driver behavior monitoring is either done based on biotic or physical features [38]. Biotic features-based schemes are categorized as driver physiological examinations. Biotic features use non-graphic topographies to detect driving behavior through sensors on the driver's body monitoring biomedical parameters of the driver. Physical features use graphic topographies to detect driving behavior by using driver facial expressions recorded through a camera. Physical features-based systems exploit computer vision techniques to detect drowsy expressions. Computer vision techniques use visual features like eye-state, eye-blinking, and mouth yawning examination [39][40][41]. These physical features-based schemes are not applicable when the driver is wearing black sunglasses or laughing during a talk with someone. Vehicle monitoring is based on signals such as changes in steering direction, speed, sudden break, etc. Driver's health monitoring uses sensors to collect the signals including the electrocardiogram (ECG) for monitoring the heartbeat, electroencephalogram (EEG) to keep a record of brain activity, and continuous glucose monitoring (CGM) checks glucose every minute. Aggressive driving behavior is also detected through both driver and vehicle monitoring. Distracted driving follows fast turn, sudden break, and weaving [42]. In Section 3 various biotic and physical-based schemes are discussed for driver behavior analysis.

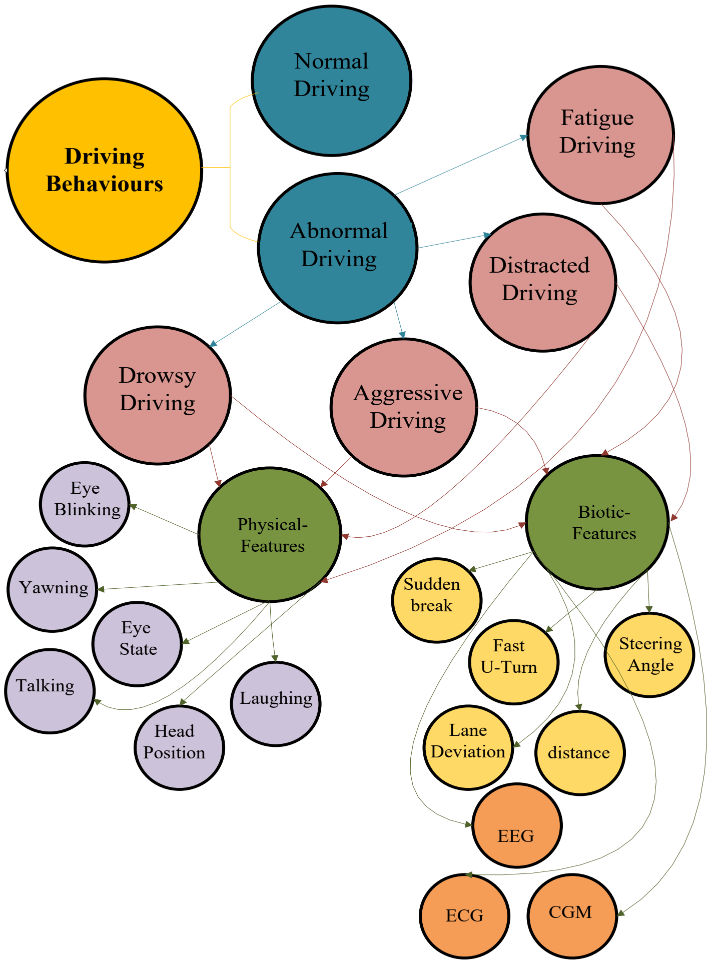


Figure 3.2: Driver behavior's ontology

3.2.2 How abnormal driving behavior causes accidents?

In the literature many studies have analyzed the relationships between abnormal driver behavior and roadway accidents [11][16]. Abnormal driver behavior affects the severity level of roadway crashes [16]. Abnormal driving such as drowsy, fatigue, aggressive or distraction due to sickness, mental distraction, or tiredness can lead to road accidents [6][8][14][26][36]. Abnormal driving behavior effects on the road in various daily driving scenarios as shown in Figure 3.3. It illustrates various abnormal driving effects on the road that can increase the rate of accidents such as (A) less vehicle distance (B) vehicle collision due to fast speed and sudden turn (C) sudden break (D) Distracted vehicle from central road line (E) blind crossing (F) no yield at highway intersections (G) ignore traffic signals.

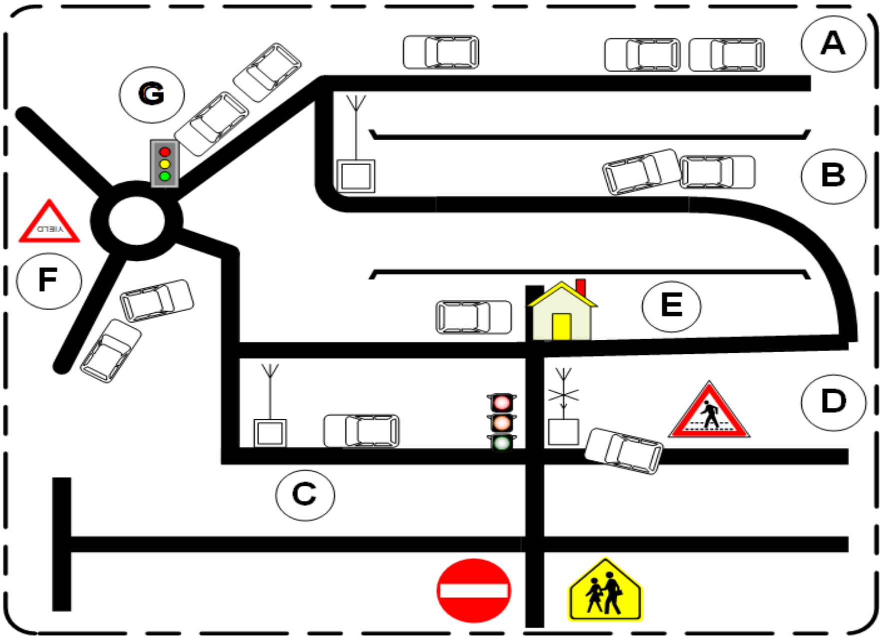


Figure 3.3: Abnormal driving effects on the road: Various scenarios where abnormal driving can increase the rate of accidents.

3.2.3 Abnormal Driving Behavior Detection

Abnormal driver behavior detection can be addressed as an anomaly detection problem. Anomaly detection in driver behavior requires various steps as shown in Figure 3.4. In the first step, the driver and vehicle are two separate entities monitored by specific sensors. In the subsequent phase, the appropriate datasets are gathered, these may consist of signals and/or images. More specifically, signals represent how certain relevant parameters (such as speed, acceleration, heading angle for what concerns the vehicle operating conditions, and EEG ECG and CGM for the driver's health conditions) change over time. In the third step, collected data is used for driver behavior characterization and modeling. This section gives an overview of driving behavior modeling and detection solutions according to the taxonomy. In the fourth step, various possible anomaly detection methods are done that cause accidents.

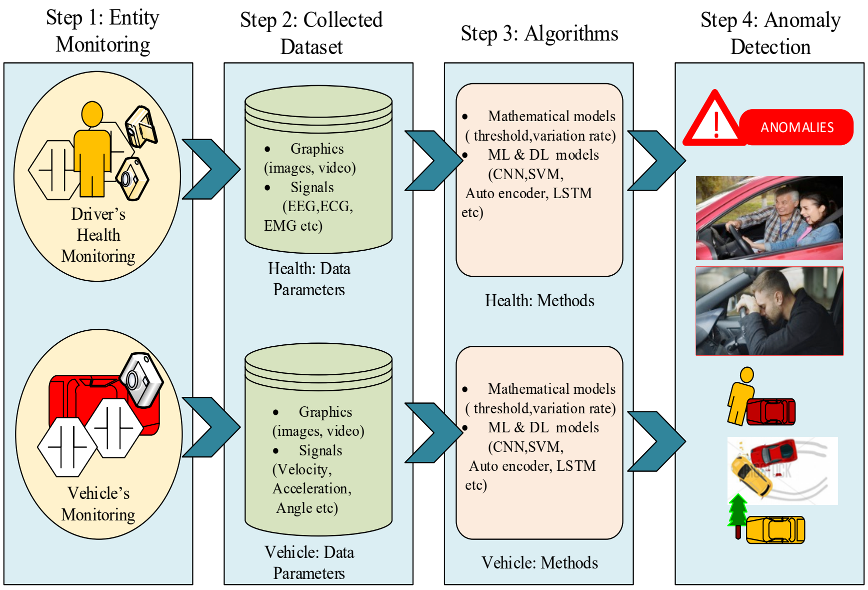


Figure 3.4: Driver Anomaly identification architecture: The major Idea of is to use vehicle/driver data to classify driver behavior.

3.3 Taxonomy of Driver Behavior Detection

In this section, a detailed description of driver behavior detection schemes is described and mapped into a taxonomy as shown in Figure 3.5.

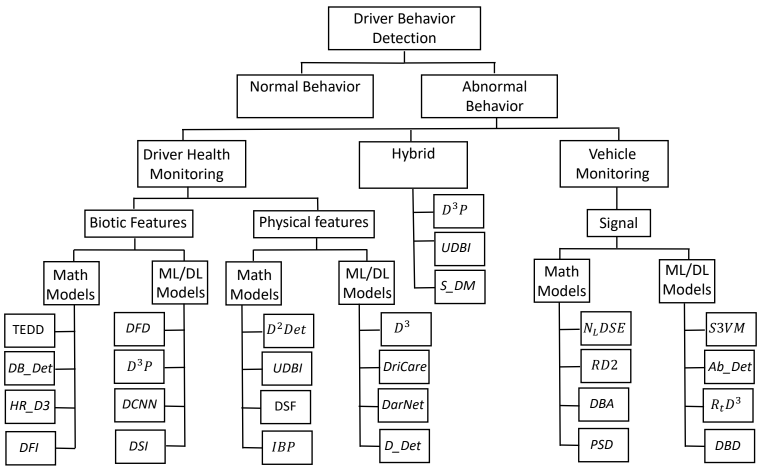


Figure 3.5: Taxonomy for driver's behavior detection-based schemes

Abnormal driver behavior detection is based on two major categories for driver's behavior detection based on (1) driver's health monitoring and (2) vehicle monitoring. Furthermore, each major category is divided into two further categories named mathematical model-based schemes and artificial intelligence-based schemes.

Artificial intelligence-based schemes are stated as machine learning and deep learning schemes.

3.3.1 Driver's Health Monitoring

Health monitoring is based on wireless sensors and smart devices that monitor various health conditions such as heart rate using ECG, brain activity using EEG, body glucose level using skin signals, etc. [43]. The above measurements take place via mobile devices or on-board computers which treat data as arrays shown in Figure 3.6. It consists of 3 levels, tier-1 The driver is equipped with sensors to collect health signals like ECG, EEG CGM, etc. The data accuracy depends on the quality of sensors that have good sensing and processing skills. At tier 2, collected data is temporarily stored for basic operations. Once preprocessing is done on collected data then information passes to the internet and server for Health services at tier 3. At tier-3 servers may include cloud or fog servers as central repositories for data analysis, decision-making, and generating alerts in time. Health monitoring is highly appreciable for the physiologic system, drug overdose, possibility of heart attack, or major trauma [13].

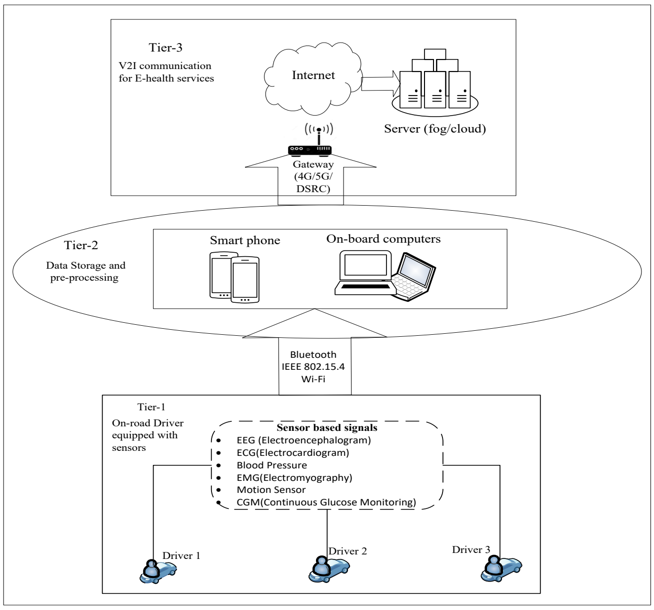


Figure 3.6: Driver's Health monitoring architecture

Driver's E-health monitoring is further categorized into two following categories;

3.3.2 Biotic Feature-based Schemes

This section describes solutions proposed in the literature that analyze driver behavior by exploiting biotic features. Biotic features refer to biological conditions of the driver such as brain activity, heart rate, and glucose level. When the biological condition of the driver is not good the driving performance decreases which may lead to abnormal driving behavior. In the past, biotic features based on mathematical, machine learning (ML), and deep learning (DL) schemes have been proposed to identify driver behavior. In the following we will overview the major contributions that exploit mathematical models then we will overview the solutions based on AI.

3.3.2.1 Mathematical models-based schemes

In [24], authors propose a driver drowsiness detection index named Thoracic Effort Derived Drowsiness index ($TEDD$), based on respiratory rate variability (RRV) to detect driver drowsiness. It uses capnography or bioimpedance-based sensors to record driver RRV values. Experimental results followed by recorded RRV values in the driving cabin simulator environment. By comparing existing work TEDD reduced false alarms but must be assessed in real environments. In [44], authors propose a driver behavior detection scheme (referred to as DB_{Det} in Figure 3.5) to identify various abnormal driver behaviors. This scheme exploits the data produced by body sensors connected by various specific body sensor networks (BSNs) integrated with vehicular ad hoc networks. The purpose of *DB_ Det* is to monitor the driver's health to identify critical conditions such as drowsiness, aggressiveness, or distraction. The scheme utilizes several Roadside Units (RSUs) to build the communication infrastructure. However, the requirement for numerous health monitoring sensors and central RSUs presents drawbacks for *DB_ Det*. In the future, integration with the Internet of Vehicles (IoVs) can be explored as an alternative to Vehicular Ad-Hoc Networks (VANETs).

In [45], a driver drowsiness detection scheme (referred to as *HR_ D3* in Figure 3.5) based on heart rate variability (HRV) is proposed. It employs recorded driver health parameters as input data. HRV utilizes a multivariate statistical distribution for anomaly detection. A notable strength of this scheme is the validation of driver drowsiness identification through EEG data during sleep. The proposed scheme has undergone validation using a driving simulator. Further validation in real driving scenarios and expansion to detect other driving states can be considered in future research.

In [46], the authors identify driver fatigue (referred to as *DFI* in Figure 3.5) using a fixed threshold-based mathematical model. It relies on health signals collected from the driver using sensors. However, a limitation of *DFI* is its limited accuracy in detecting other driver behaviors when using a fixed threshold.

3.3.2.2 ML and DL based schemes

In [47], the authors propose a driver fatigue detection (referred to as DFD in Figure 3.5) scheme by using a Gaussian Model (GM) with Support Vector Machine (SVM) that takes input health signals collected in real-time from the driver. GM with Support Vector Machine improved accuracy with the limitation of high calculation cost. In [48], authors propose deep convolutional neural networks (referred to as DCNN in Figure 3.5) to detect aggressive driver behavior by using bio-signals of the driver such as heart rate, blood pressure level brain activity, etc. These are collected by using body sensors and camera sensors. In the future, other deep learning models can be explored to detect various driving behaviors using bio-signals. In [49], the authors propose driver stress identification (referred to as DSI in Figure 3.5) scheme. The purpose of this scheme is to detect abnormal driving behavior by observing driver stress levels. DSI uses EEG signals with SVM and random forest models to classify driver behavior. Limited accuracy with the threat of false alarms is a drawback. Furthermore, in this case, other driver behaviors can be considered as future extensions.

3.3.3 Physical feature-based Schemes

This section describes driver behavior identification schemes that are physical features of the driver, i.e., features that are visible or tangible such as body movement and facial expression. In the past, several driver's physical features-based schemes have been proposed. In this section, we will discuss the distinguishing two subcategories mathematical and AI-based schemes.

3.3.3.1 Mathematical models-based schemes

In [37], the authors propose a driver drowsiness detection scheme (referred to as *D2_Det* in Figure 3.5) that also assists drivers in preventing lane departure accidents. Experimental results are obtained through simulation, and the proposed solution has not been validated in real-world environments.

In [50], authors introduce a driving safety features scheme (referred to as *DSF* in Figure 3.5) designed to identify abnormal driving features for safe driving. *DSF* utilizes image datasets captured by a camera mounted in front of the driver. It employs the histogram algorithm in combination with an SVM classifier. However, *DSF* may encounter misclassification due to misleading facial expressions, such as laughing or talking, which could be improved in future research.

In [51], an Individual Behavioral Pattern-based scheme (referred to as *IBP* in Figure 3.5) is proposed for identifying individual driver behavior patterns. This scheme employs multilinear regression analysis to specify patterns for individual driver behavior identification. While *IBP* has been validated through simulation, assessment in real-world environments is yet to be conducted.

In [52], the authors introduce an Unsafe Driving Behavior Identification scheme (referred to as *UDBI*) aimed at identifying drowsy driving behaviors using a fixed threshold-based mathematical model. *UDBI* utilizes driver and vehicle-based datasets collected through smartphones and sensors. Future research could explore the implementation of mobile applications for recognizing other abnormal driving behaviors.

3.3.3.2 ML and DL based schemes

ML and DL-based schemes use artificial intelligence to detect driving behavior. Driver behavior detection for road and vehicle safety can be improved by using machine and deep learning significantly, also because of the high volume and variability of the data that can be collected [10] [17][18][30]. In [42], The authors introduce a Driving Detection and Identification scheme (referred to as *DDD* or D^3 in Figure 3.5) to detect abnormal driving behaviors such as fast U-turns, speedy turning, and sudden braking. *DDD* utilizes smartphones for data collection,

Sensors are used to collect real-time data, and machine learning (ML) techniques are employed to detect abnormal driving behavior. The D^3 scheme leverages a Support Vector Machine (SVM) to train a Neural Network (NN) for real-time classification of driving behaviors. However, D^3 has limitations, including the identification of only individual driver behavior and high time consumption.

In [39], the authors introduce a lightweight drowsiness detection scheme (referred to as *LWD2*) that utilizes images captured by a camera mounted in front of the driver. *LWD2* employs the Viola-Jones algorithm for detection. A limitation of this approach is that recorded facial expressions may lead to confusion during behavior classification.

In [38] and [39], a scheme named *Driver Care* (referred to as *DriCare*) is proposed for real-time driver drowsiness detection using computer vision techniques. Real-time data is collected via a vehicle-mounted camera, and a Multi-convolutional Neural Network (CNN) is used to detect driver drowsiness from video images. Major limitations of *DriCare* include potential confusion of drowsy facial expressions with other expressions, such as talking, laughing, or surprise.

In [53] and [54], authors propose data collection and analysis schemes for detecting distracted driver behavior, named *DarNet* and *D_Det*, respectively. Both schemes leverage deep learning models like CNN and RNN, using images for image processing and time series analysis.

In [55], authors propose a distracted driving identification scheme (referred to as *D2I*) using real-time data collected by a front-mounted camera. *D2I* identifies distracted driving behavior using CNN, with a focus on reducing computation costs and false alarms.

In [56], authors introduce a deep sparse autoencoder (DSAE) to identify driver behavior by extracting hidden features from recorded vehicle parameters such as acceleration and speed. DSAE is an unsupervised approach, and a limitation is that the degree of freedom in the feature space varies with time.

In [57], authors propose an aggressive driver behavior detection scheme (referred to as *ADB_Det* in Figure 3.5) that employs a CNN taking the driver's facial images as input. Limitations of both schemes include potential confusion caused by facial expressions during talking and laughing, and a focus solely on aggressive driving behavior, leaving room for exploration of other behaviors in the future.

From the physical feature-based schemes mentioned above, it can be concluded that facial expressions are commonly used inputs for detecting anomalies. In Equation 3.1, M represents the difference between input and reconstructed output, where recorded facial expressions (f_{E1}) and reconstructed facial expressions (df_{E1}) are compared. The input is represented as a matrix $\mathrm{MT}_1[a_1, b_1]$ containing a_1 and b_1 columns, which represents the image. Detected anomalies (DA_s) and undetected anomalies ($n\mathrm{Da}$) are compared, and if the difference between detected and original anomalies is greater than zero, it indicates successful anomaly detection, as shown in Equation 3.1, otherwise, it is not detected, as shown in Equation 3.2.

Furthermore, it can be observed that driver health parameters such as heart rate, brain activity, and glucose level are collected using body sensors in the form of signals. In [57], the authors propose an aggressive driver behavior detection scheme (referred to as *ADB_Det* in Figure 3.5) that utilizes a Convolutional Neural Network (CNN) taking the driver's facial images as input. However, the major limitations of these schemes are related to facial expressions during talking and laughing, which can lead to confusion in behavior detection. Additionally, this work focuses solely on aggressive driving behavior, leaving room for exploration of other driving behaviors in future research.

From the above physical feature-based schemes, we can conclude that facial expression is a usual input to models for detecting anomalies. In Equation 3.1 M refers to the difference between input and reconstructed output. Recorded facial expression and reconstructed facial expressions are represented by f_{E1} and df_{E1} respectively. Input is a matrix $\mathrm{MT}_1[a_1, b_1]$ which represents the image and contains a_1 and b_1 columns. Detected anomalies and not-detected anomalies are represented by by DA_s and $n\mathrm{Da}$ respectively. If the detected and original anomalies difference is greater than zero it means the anomaly is detected successfully as shown in Equation 3.1 otherwise not detected as shown in Equation 3.2. Furthermore, we can also conclude that the driver's health parameters such as heart rate, brain activity, and glucose level are collected using body sensors in the form of signals. We show the driver behaviors, attention to the driver's health parameters considered for detection, and the resulting quality of the estimation variations concerning the driver's health parameters as shown in Table 3.1. We report for each type of anomalous driver behavior the level of attention (i.e., "H": high; "M": medium, and "L": low) received by the major health parameters that can be collected by body sensors (i.e., "Heart rate, brain activity, and glucose level) in the scientific literature, and the achieved level of quality of the estimation in terms of Recall and Precision. We observed how the driver's health-based signals are affecting driver behavior detection.

$$M = \|f_{E1} - df_{E1}\| > 0 \rightarrow \mathrm{DA}_s \tag{3.1}$$

$$M = \|f_{E1} - df_{E1}\| \approx 0 \rightarrow nDa \quad (2) \tag{3.2}$$

Where $f_{E1} = \mathrm{MT}_1[a_1, b_1]$ and $\mathrm{df}_{E1} = \mathrm{MT}_1[\Delta a_1, \Delta b_1]$

Note: L, M, H for Low Medium, and High respectively

3.3.4 Vehicle Monitoring based Schemes

In Vehicle monitoring-based schemes the vehicle is equipped with some sensors to measure and record parameters such as speed, position, acceleration, steering angle, etc. These parameters help in detecting driving behavior. The resulting

Table 3.1: Driving Behavior and Health Parameters

Driving Behavior	Heart rate	Brain activity	Glucose level	Recall	Precision
Drowsy	H	M	M	H	M
Aggressive	L	M	H	M	M
Normal	H	M	H	H	H
Distracted	L	H	M	M	L
Fatigue	L	M	M	M	M

schemes can be classified into two categories.

3.3.4.1 Mathematical models-based schemes

These schemes are based on statistical or mathematical analysis.

In [58], the authors propose a Non-linear Driver Steering Estimation (N_L DSE) approach to detect driver behavior by observing the angle of the steering wheel. It uses sensors to acquire steering wheel torque data in real time. N_L DSE follows a two-point visual driver model along with filters such as Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) for identifying abnormal driver behavior. However, the N_L DSE approach considers only expert driver scenarios, which is a limitation.

In [59], authors introduce a scheme called Risky Driver Detection (RD2) to detect risky driver behavior by monitoring real-time speed time series. It employs a mathematical method that measures speed parameter variation concerning various driving patterns. RD2 is easy to implement because almost all vehicles have a speed measurement system. However, its limitation is that testing is done on a specific area (the ring road of Beijing).

In [60], the authors present a Driver Behavior Analysis (referred to as DBA) technique that exploits a Gaussian Mixture Model (GMM) to identify patterns through statistical analysis. The dataset for DBA is collected from in-vehicle sensors using the Control Area Network (CAN) bus, Inertial Measurement Unit (IMU), and the Global Positioning System (GPS). The strength of this work is in classifying driver behavior and identifying aggressive driver behavior, with the detection of other behaviors considered for future work.

In [61], the authors propose a personal steering-based driver behavior detection scheme (referred to as PSD). It uses recorded steering wheel values to predict driver behavior and employs a compensatory feedforward transfer function. However, results are obtained only through simulation.

In [62], the authors introduce an aggressive and normal driving classification scheme (referred to as ANDC) that is data fusion-based and classifies normal and aggressive behaviors of drivers. The scheme uses data collected by in-vehicle sensors such as GPS, mounted cameras, etc. A limitation of ANDC is the involvement of expensive in-vehicle sensors.

In [63], the authors present a lane deviation-based driving behavior identification scheme (referred to as LDB2I). It uses image data and applies mathematical models based on thresholds to analyze and classify driver behavior. The computation cost is high due to manual mathematical calculations, which can be improved in the future with the use of AI.

3.3.4.2 ML and DL-based schemes based on Signals data

ML-and DL-based schemes use artificial intelligence to detect driving behavior. Machine and Deep learning-based schemes results are significantly better than those from mathematical schemes in driver behavior recognition [18]. In [64], the authors propose an approach named Semi-Supervised Support Vector Machine (S3VM) to classify abnormal driver behavior. This approach is validated by using offline vehicle dataset parameters (i.e. velocity, acceleration) and simulation as a test environment. It uses the k-means clustering method to label training data. SEVM reduces labeling effort in classifying driver behavior and increases accuracy by 10% compared to simple SVM. In the future S3VM can be explored for other driver behaviors specifically. In [2], the authors propose an abnormal driving behavior detection scheme (referred to as Ab_Det) using a methodology named soft thresholding and temporal convolutional neural network (STCNN). This approach leverages real-time data collected with sensors (i.e., GPS, inertial, cameras) to significantly improve accuracy compared to existing schemes.

In [22], authors introduce a scheme called Real-time Distracted Driving Detection (referred to as $R\_t\ D^3$), which aims to detect driver distraction behavior using the Data Mining (DM) method. DM utilizes vehicle dynamic data (speed, acceleration, etc.) as input to a recurrent neural network-based classifier. $R\_t\ D^3$ collects data from in-vehicle sensors to address visual distraction detection but has limitations in considering real scenarios and dynamically exploring other driver behaviors.

In [65], the authors present the Distracted Driving Behavior Detection scheme (referred to as DDb_Det), which identifies distracted driver behavior by tracking lane-changing records on a one-way road. DDb_Det consists of three stages and employs the K-nearest neighbor (KNN) method to detect distracted driving. It collects real-time speed and steering angle data from the CAN bus, preprocess it, applies spectral time-frequency-based segmentation, and uses the KNN and Hidden Markov Model for behavior classification. Accuracy may be lower during road curves and roundabouts but can potentially be improved in the future. In [21], the authors propose the Driver Behavior Detection scheme (referred to as *DBD*) to identify various driver behaviors, including drowsy, distracted, and aggressive driving. *DBD* uses real-time signals such as revolutions per minute (RPM), speed, acceleration, throttle, etc. These signals are converted into 2D images, and image processing techniques are applied to classify driving styles. Future work may involve calculating accident probabilities and improving signal processing without the need for image conversion.

In [66], the authors propose the Dangerous Driving Behavior Detection scheme (referred to as DDB_DET). The primary purpose of DDB_DET is to identify abnormal driving behaviors using recorded video as a dataset. Specifically, it employs particle swarm optimization neural networks with support vector machines. However, a major limitation of this work is the high false alarm rate obtained during DDB_DET validation.

From the above schemes, it is evident that vehicle parameters such as speed, acceleration, headway distance, and steering are collected from in-vehicle sensors in the form of signals and are instrumental in detecting abnormal driving behavior. These schemes use signal transformations into time series (TS), as shown in Equation 3.6. They employ a threshold based on various algorithm parameters. An anomaly is considered detected if changes in the time series (ΔTS) are equal to or greater than the threshold; otherwise, it is not detected. Table 3.2 explores driver behaviors, attention to vehicle parameters used for detection, and the resulting quality of the estimation variations with respect to extracted vehicle parameters.

Table 3.2: Vehicle Parameters and Quality Estimation for Driving Behavior

Driving Behavior	**Velocity**	**Headway Distance**	**Acceleration**	**Steering**	**Recall**	**Precision**
Drowsy	M	L	H	M	M	L
Aggressive	M	L	H	H	M	M
Normal	H	M	H	H	H	H
Distracted	L	M	M	M	M	M
Fatigue	M	M	M	L	H	M

Note: L, M, H for Low Medium, and High respectively

We report for each type of anomalous driver behavior the level of attention (i.e., "H": high; "M": medium, and "L": low) received by the major vehicle parameters that can be collected by in-vehicle sensors (i.e., "velocity, headway distance, acceleration, and steering angle) in the scientific literature, and the achieved level of quality of the estimation in terms of Recall and Precision.

$$\mathrm{TS}_1 = \{x_1y_1, x_2y_2, x_ny_n\}, \quad \mathrm{TS}_2 = \{x_1y_1, x_2y_2, x_ny_n\} \quad (3) \tag{3.3}$$

$$\Delta\mathrm{TS}(\mathrm{TS}_1 - \mathrm{TS}_2) \geq \text{Threshold} \rightarrow \mathrm{DA}_S \tag{3.4}$$

$$\Delta\mathrm{TS}(\mathrm{TS}_1 - \mathrm{TS}_2) < \text{Threshold} \rightarrow \mathrm{NDA} \tag{3.5}$$

$$\text{Where} \quad \text{Threshold} = \mathrm{AVG}_n \tag{3.6}$$

3.3.5 Hybrid Schemes

Hybrid schemes include schemes that monitor both driver and vehicle parameters to identify abnormal driver behavior.

In [67], the authors propose an Unsafe Driving Behavior Identification (referred to as *UDBI*) to identify drowsy driving behaviors using a fixed threshold-based mathematical model. UDBI utilizes driver and vehicle-based datasets collected by smartphones and sensors. Future work may involve implementing mobile applications to recognize other abnormal driving behaviors.

In [68], the authors propose a Driver Drowsiness Detection and Prediction (referred to as D^3P in Figure 3.5) scheme that utilizes an Artificial Neural Network (ANN). The D^3P scheme uses real-time signals such as the driver's heart and respiration rate variabilities, as well as vehicle parameters like speed, steering angle, and lane position. This scheme's strengths include drowsy behavior prediction in addition to detection. However, the scheme has been validated only through simulation, and assessment in a real environment is still pending. Future work may involve extending the scheme to detect other critical driver behaviors besides drowsiness.

In the project detailed at `https://safedemon.it/en/home-page-english`, the authors propose an innovative solution that considers both driver, vehicle, traffic, and road factors within a single platform to detect driver behavior, as shown in Figure 3.7. Once driver behavior is detected, the proposed architecture is responsible for informing upcoming traffic and Intelligent Transportation Systems (ITS) to avoid potential dangers. This architecture consists of three modules.

In the first module, drivers and vehicles are equipped with sensors to record the dataset. The vehicle is equipped with Advanced Driver Assistance Systems (ADAS), GPS, Inertial Measurement Unit (IMU), On-Board Unit (OBU), and Maps to collect parameters such as position, speed, heading, front vehicle headway, and road alignment. Drivers wear Electroencephalography (EEG) and Continuous Glucose Monitoring (CGM) sensors to capture data in signal waves.

In the second module, a convolutional auto-encoder processes the recorded signals as time series and converts them into sequences. Once a dataset is treated as a sequence, it is passed to the model as a string. The convolutional auto-encoder comprises various layers where the original input is encoded into a latent vector and then decoded. Finally, the decoded string is compared with the original input to detect anomalies. Anomaly detection in this context is equivalent to driver behavior detection.

Once driver behavior is detected, the third module is responsible for communicating with oncoming traffic and ITS to take precautionary steps and prevent major losses. This project explores data collection methods to identify accidents resulting from alcohol or drug use. Additionally, it compares accidents caused by driver illness with those due to drowsy conditions. The project concludes that accidents due to illness are more prevalent in the age group over 60, while accidents due to drowsy conditions are most common in the age group below 45.

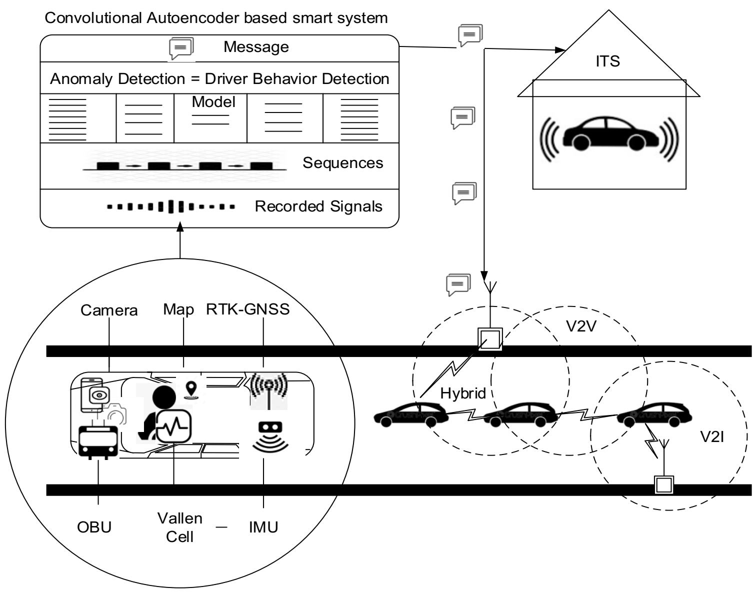


Figure 3.7: Safe-Demon Architecture

3.4 Analytical Discussion and Open Research Issues

In this section, a detailed comparison of driver behavior detection schemes is carried out on the basis of a few metrics including data collection tool, test environment, method, strength, limitation, and remarks of each scheme. After comprehensive comparison, some open research issues are identified that need to be explored in the future. A comparative summary of driver's behavior detection-based schemes is presented in Table 3.3. We identified that driver behavior detection techniques are based mainly on two factors i.e. driver's health monitoring and vehicle monitoring. The 1st column of the table reflects the scheme name with its reference while the 2nd column highlights the monitoring model which means whether the driver's behavior detection is based on driver health monitoring (D), vehicle monitoring (V), or hybrid (H). The 3rd column illustrates the test data environment which means whether the results are carried out by using a real-world (RW) dataset or mapped on a simulator (SM). In the 4th column, the data collection method is mentioned whether the dataset is collected by using smartphone (SP), sensors (S), or mounted cameras (MC) in the form of images or signals. The 5th column explores the methodology of each scheme which is the most important element of any scheme. Another two columns compare the algorithms using the strength and limitation of each scheme respectively. Finally, the last column highlights some valuable remarks of each scheme that can help to improve driver behavior detection in the future. From Table 3.3, we conclude that driver behavior detection mainly depends upon driver and vehicle parameters as shown in the 2nd column. Driver behavior identification schemes used either real-world datasets or simulation tools to validate results. Furthermore, real-world datasets are categorized into offline or real-time datasets. In [21][22][38][42][46][58], authors identified driver behaviors in real-time by using in-vehicle sensors and cameras. In [59][53][56][57][60][54][47] [67], authors detected driver behavior by using off-line real-world data parameters of vehicle or driver. From Table 3.3, we concluded that abnormal driving behavior identification is mostly done by using smartphones, cameras, and sensors-based dataset [38][42][46]. For the real-world dataset, we observed that sensor-based data gives more false alarms and takes high execution time due to noisy signals collected at high frequency (i.e. f$>$10 Hz) [44][47][62]. Smartphone and mounted camera-based driving behavior detection approaches also raise false alarms [38][53][57] due to misleading facial expressions. For the simulation test environment (3rd column), we observed that mostly static simulators are considered to lead to a high rate of false alarms and limited accuracy when implemented in real scenarios [22][37][45][61][64]. For the real world test environment, it is also observed that some existing schemes considered small sample size or hotspot area for result validation that can be a challenge for real-world applications [21][59][67]. For sensors as a data collection tool, the observed limitation is the high cost of involved hardware and devices [44][45][62]. From Table 3.3, mainly observed that most of the schemes addressed any one driver behavior at a time [39][66][47][53][57]. It also has been observed that ML and DL based schemes improved accuracy and labeling effort significantly [66][57][42] [64] [2]. After the literature review and analytical discussion, we identified the following open research issues that need to be addressed in the future. Addressing such issues will require a large research effort and we believe that interested researchers might benefit from the overview of Table 3.4 where the relevant publication venues are explored.

A. Reduce false alarms generated from noisy signals and confusing facial expressions

From [38] [42] [53][57] [58], we concluded that driver behavior detection based on the sensors or mounted cameras causes more false alarms. Noisy signals and images of facial expressions collected from various sensors and cameras lead to false alarms. Basically, false alarms are misleading detections that occur due to unexpected peaks of noisy signals or confusing facial expressions such as laughing or talking, etc. Therefore, dynamic and efficient systems are required to reduce false alarms in the future.

B. Reduce delay as time is one of the parameters to measure the efficiency of any system

From [21][49][55], we concluded that combinations of multi-models are used for driver behavior detection that takes more model parameters and results in high time consumption. Moreover, time series models, denoising, and data smoothing need to operate in a time window that can be more than one second depending also on the data acquisition frequency. High time consumption during abnormal driver behavior detection causes accidents and loss of precious lives. Time is a well-renowned parameter to verify the model's efficiency. Therefore, it is required to build models with less time cost and real-time response to identify abnormal driver behavior and take precautionary actions in time.

C. Need to follow real-world datasets and test environment instead of virtual simulation with static network

From [22][37][45][61][64], we concluded that driver behavior identification based on simulations does not work well when implemented in real scenarios. In real scenarios, simulation-based systems create more false alarms and give a limited rate of abnormal driver behavior identification. Therefore, it is required to consider real-world scenarios and dynamic environments for abnormal driver behavior identification to prevent accidents in real scenarios. **D. For a real-world test environment, hot spot areas and small sample sizes for testing are considered as a limitation, therefore consider real scenarios with large datasets**

From [21][59][52], we concluded that some of the existing work considered small sample size and hotspot area or selected road environment to validate the results of abnormal driver behavior identification. Basically, the test carried out by using a small sample size and hot-spot area is not sufficient and suitable to implement in a dense real environment. The system trained with a limited dataset of specific users and simple road scenarios cannot work well for other users and various real road conditions. Nevertheless, it is obvious that developing a specific model for each user in each environment is not practical, and appropriate solutions are needed to overcome such limitations.

E. Reduce the cost of the hardware devices involved

From [44][45][62], we concluded that sensors based on abnormal driver behavior identifications are highly costly. Although the installation costs will be reduced if data can be collected by the sensors already embedded in automated vehicles of level 1+, in low-medium developing countries it is difficult to afford expensive sensors and vehicles. Therefore, it is important to explore cheap hardware devices and sensors to collect data to identify abnormal driver behavior.

F. Mainly observed that most existing research articles address any one driver behavior, need to address various abnormal driver behaviors at a time

From Table 3.3, we concluded that several schemes identify only abnormal driver behavior at a time while a driver can suffer from various abnormal driver behaviors at one time. If a system can identify various abnormal behaviors, it would be more dynamic and effective in the application. For example, if a system could not identify abnormal driver behavior from one aspect, maybe it could do so for another one. Therefore, it is required to explore a dynamic scheme that can identify various abnormal driver behaviors.

G. Need ML and DL-based smart schemes to improve accuracy, reduce labeling effort and false alarms for abnormal driver behavior identification

From [42][57][57][64][2], we observed that ML and DL-based smart schemes reduce data labeling effort and improve accuracy significantly. Moreover, ML and DL are more suitable to handle big and heterogeneous data. Therefore, it would be great in the future to continue to explore ML and DL for abnormal driver behavior identification.

H. Require hybrid schemes based on driver's health and vehicle monitoring

From Table 3.3, we can identify that most of the schemes are either based on the driver's health or vehicle parameters to identify abnormal driver behavior. This is a unidirectional way to identify driver behavior. There are very few hybrid schemes that are based on both driver's health and vehicle monitoring. Hybrid schemes are a bidirectional approach to identifying abnormal driver behavior in an effective way to reduce false alarms by comparing detections from different simultaneous models. Therefore, we need to explore more hybrid schemes in the future.

3.4.1 Remarks

In this part of the thesis, we have reviewed the principal solutions proposed for driver behavior detection. These solutions are categorized into three main categories: those that exploit driver's health parameters only, those that use parameters coming from the vehicles, and those that consider both. Each solution is described in terms of the data that it exploits, the tools that it exploits, and its strengths and weaknesses. Based on the analysis of such solutions we have identified the data collection for these schemes follows two ways i.e. by using cameras and sensors. It might be a smartphone camera or a mounted camera in a vehicle that collects data in the form of images. Sensors are devices to collect data from vehicles and drivers in the form of signals. Test environment scenario based on real-time/real world and non-real-time/no real world categories such as simulation-based driver behavior detection. Real-time techniques where result evaluation is processed on real-time collected data. Non-real techniques include simulation-based results. After the literature review, a taxonomy is included for a clear understanding of driver behavior detection schemes. After a detailed literature review and taxonomy construction, an analysis section is included to highlight the literature gaps and how it can be possibly improved in the future. Future directions of this study include: 1. Explore various driving behaviors. 2. Detect maximum driving behaviors by using a single AI-based method. Explore dynamic algorithms to identify driver behaviors. Reduce the cost of expensive sensors. For graphic topography-based schemes recognition of facial expressions and tracking remains a challenge.

Table 3.3: Algorithm Analysis and Comparison for Driver Behavior Detection

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
DDD^3 [42]	V	RW	SP	Support Vector Machine Neuron Network	High accuracy.	Only abnormal driving style is addressed. Time-consuming due to a lot of parameters of SVM and NN. Need a smart algorithm to train and classify real-time datasets for better results.
N_L DSE [58]	V	RW	S	Two-Point Visual Driver Model Estimated Kalman Filter (EKF)	Steering wheel based abnormal driving detection.	Only experienced driver is considered. High time. Need to classify abnormal driving behaviors.
IBP [51]	D	SM	S	Multiple linear regression analysis	A base to establish a solid logic for the integration of driver.	Explore other driver behaviors such as aggressive behavior. Need other driving behaviors and implement using real data instead of virtual simulation.
S3VM [64]	V	SM	S	K-means Clustering	Reduce labeling effort. Improve accuracy.	Real dataset is considered as a limitation. Needs to explore deep learning methods using real scenarios.

Continued on next page

Table 3.3 – continued from previous page

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
RD2 [59]	V	RW	S	Speed change based mathematical calculation	Identify risky driver behavior using speed.	Easy to implement. Hotspot area is considered. Static approach. Need a dynamic approach to monitor driver behaviors online.
R_t D^3 [22]	V	SM	S	Data Mining (DM) method	Driver distraction detection.	Static network. A deeper investigation of the LRNN classifier is required for other driving behaviors.
TEDD [24]	D	SM	S	Respiratory Rate Variability (RRV)	Reduce false alarms.	Results are not compared with counterparts. Need to work with real datasets for real driving drowsiness detection.
DriCare [38]	D	RW	MC	Multi CNN	Handle the challenge like driver's height. Improved accuracy.	During talking, laughing facial expression can be similar to a drowsy state. Need dynamic technique for driver drowsiness detection.
DarNet [53]	D	RW	MC	CNN for video analysis + RNN for IMU time series analysis	Align data across various IoT modalities. Improve accuracy.	Laughing facial expressions can be similar to distracted expressions. Need to consider other driving behaviors under privacy concerns.

Continued on next page

Table 3.3 – continued from previous page

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
DBD [21]	V	RW	S	2D CNN + Recurrence plot technique	Multiple driving behavior detection.	Probability of accident is not calculated. Hotspot area is considered. Recorded signals for anomaly detection instead of images, which is time and cost-consuming.
PSD [61]	V	SM	S	Steering angle variation	Less time consumption.	Limited accuracy in real-world scenarios. Need to consider real dataset parameters.
D^2 Det [37]	D	SM	C+MC	Driver Assistance system with dual control scheme	Prevent lane departure accidents.	Eyelid movement can confuse. A real scenario is required to be considered.
DSAE [56]	V	RW	S	Deep Sparse Autoencoder (DSAE) scheme	Dynamic visualization identify driver behavior.	A Practical support system is required for this method. Need possible application of DSAE for vehicles.
ADB_Det [57]	D	RW	S+MC	CNN	High accuracy.	Single-driver behavior and facial expression can lead to confusion. Need to consider other driving behaviors.
DB_Det [50]	D	RW	S	Integration of BSNs	Identify multiple driver behaviors in a real scenario.	More central devices for communication. High time. Need to implement this approach for IoVs.

Continued on next page

Table 3.3 – continued from previous page

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
DBA [60]	V	RW	S	Gaussian Mixture Model (GMM)	Driver behavior analysis and aggressive behavior identification.	Only one behavior identification. Need to identify other driver behaviors.
ANDC [62]	V	RW	S	Data Fusion	Aggressive and normal driver behavior classification.	Hardware is involved. High time. Need to identify various driver behaviors.
LWD2_det [39]	D	RW	MC	Viola-Jones algorithm	Driver drowsiness detection in a real scenario.	Facial expressions can mislead the algorithm. Need to explore other driving behaviors.
HR_D3 [45]	D	SM	S	HRV analysis by comparing with EEG	Driver drowsiness detection and its validation.	Driver equipped with many wires and sensors. High cost. Need to explore other driving states.
DDb_Det [65]	V	RW	S	K-nearest neighbor	Distracted driving behavior detection.	Poor accuracy on corners or curve roads. Need to consider more realistic scenarios.
D_Det [54]	D	RW	MC	CNN	Distracted driving detection using images.	Limited resolution of images. Computer vision for other abnormal driver behaviors.
Ab_Det [2]	D	RW	SP	Soft Threshold and Temporal CNN	Improve accuracy and robustness.	Individual driver behavior is not classified. Need to explore accuracy for individual driver behavior detection.

Continued on next page

Table 3.3 – continued from previous page

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
DFD [47]	D	RW	S	Gaussian + SVM	Significantly detect driver fatigue.	High calculation load. Need to explore efficient methods for various driver behaviors.
DFI [46]	D	RW	S	Threshold	Detect driver fatigue automatically.	Static threshold. Need to explore dynamic solutions.
D2I [55]	D	RW	MC	CNN + Weight pruning	Less false alarms and high accuracy achieved.	Limited to one driver behavior. Can improve accuracy.
D^3P [68]	H	SM	C+MC	ANN (Artificial Neural Network)	Predict driver drowsiness along with drowsy behavior detection.	Real dataset for experiments is required. Consider a real scenario for driver behavior detection.
DSF [50]	D	RW	C	HOG + SVM	Abnormal driving behavior detection.	Limited accuracy due to distracted facial expressions. Improve accuracy in the future.
UDBI [52]	H	RW	S+SP	Threshold-based	Predict drowsy driving behavior using pulse rate and vehicle position.	Small sample dataset. Implement for other dangerous driving behaviors.
DCNN [48]	D	RW	S	CNN	Real-time aggressive behavior detection.	Limited to one driver's behavior detection and accuracy. Need to explore other deep models for driver behavior detection.

Continued on next page

Table 3.3 – continued from previous page

Technique	Model	Test Env.	Tool	Method	Strength	Limitation
DSI [49]	D	RW	S	SVM + Random forest	Driver stress identification.	Limited accuracy. Need to explore other driving behaviors using deep learning.

Table 3.4: SURVEY CRITERIA

No.	Publication Venue	Publication-Year (Count)	Type	No. of articles	Keyword criteria	Impact Factor 2021
1.	IEEE Transactions on Intelligent Transportation Systems	2013, 2015, 2017, 2019, 2020	Journal	5	Safe driving, Driver Behavior detection, AI	6.492
2.	IEEE Transactions on Human-Machine Systems	2016, 2017 (3), 2018	Journal	5	Driver behavior, Accidents	2.968
3.	MDPI Sensors	2015, 2016, 2018, 2019	Journal	5	Driving monitoring, Driver behavior, Traffic safety	3.576
4.	Transportation Research Part F: Traffic Psychology & Behavior	2019 (3)	Journal	3	Health, driving patterns	3.261
5.	IEEE Access	2019 (2)	Journal	2	Health based driver behavior identification	3.367

6.	IEEE Intelligent Vehicles Symposium	2013, 2016	Conference	2	Driver behavior recognition and classification	-
7.	IEEE Communications Surveys & Tutorials	2018	Journal	1	Road Safety	23.7
8.	ACM Computing Surveys	2018	Journal	1	Driver behavior, stress	10.28
9.	IEEE Network	2018	Journal	1	Vehicle Safety Improvement	10.693
10.	International Journal of Vehicular Technology	2016	Journal	1	Driver Behavior	-
11.	Accident Analysis & Prevention	2020	Journal	1	Road Safety	3.655
12.	Safety Science	2018	Journal	1	Driver behavior	4.877
13.	Engineering Applications of Artificial Intelligence	2020	Journal	1	Driving behavior, AI	6.212
14.	IEEE International Conference (FSKD-12th)	2016	Conference	1	Driver behavior, Deep learning	-
15.	Journal of Safety Research	2018	Journal	1	Driving Behavior, road safety	3.487

16.	IEEE Intelligent Transportation Systems Magazine	2015	Journal	1	Driver Behavior	3.419
17.	Expert Systems with Applications	2020	Journal	1	Driver behavior detection, AI	6.954
18.	International Journal of ITS Research	2020	Journal	1	Driver behavior	1.385
19.	Physiological Measurement	2021	Journal	1	Abnormal driver behavior, Health	2.833
20.	Frontiers in Neuroscience	2018	Journal	1	Driver states	3.566
21.	International Conference PRASA (RobMech)	2017	Conference	1	Driver behavioral measures	-
22.	Transportation Research Part A: Policy & Practice	2019	Journal	1	Driver behavior, Road Safety	5.594
23.	Pattern Recognition	2018	Journal	1	Road safety, Safe Driving	7.740
24.	Security and communication networks	2020	Journal	1	Abnormal driver behavior detection	1.791
25.	Sustainability	2019	Journal	1	Driver behavior	3.251
26.	Intl. Journal of Applied Engineering Research	2018	Journal	1	Driver behavior detection	8.10
27.	IEEE Signal Processing Magazine	2017	Journal	1	Driver behavior detection	11.64

28.	MDPI Symmetry	2020	Journal	1	Driver Behavior, Road Safety	2.713
29.	IEEE I2MTC	2013	Conference	1	Driver behavior	-
30.	International ICOEI	2021	Conference	1	Driver behavior	-
31.	IEEE Transaction on Mobile Computing	2017	Journal	1	Driving behavior detection	5.112
32.	International Middle	2017	Conference	1	Driver behavior	-
33.	IEEE Transactions on Biomedical Engineering	2019	Journal	1	Driver behavior, Drowsiness	4.538
34.	IEEE Transactions on Intelligent Vehicles	2017	Journal	1	Lane change detection	6.72
35.	Accident Analysis and Prevention	2019	Journal	1	Diver drowsiness prediction	4.993
36.	Journal of ambient intelligence and humanized computing	2019	Journal	1	IoVs	4.81
37.	IEEE Internet of things	2018	Journal	1	IoVs	9.936

Chapter 4

Use Case 1: Bicycle

4.1 Bicyclist behavior-based Anomaly Detection through Deep Learning

4.1.1 Orientation

Cycling is a key component of any sustainable urban mobility in terms of environment and public health and as an alternative to driving a car [69]. The Netherlands is leading the ranking in Europe with 27% of trips done by bicycle with other countries (e.g. Denmark, Belgium, and Germany) already beyond the 10% threshold. Below 5% we find countries like Norway (4,3%), Italy (3,3%), France (2,7%), and Luxembourg (2%). Anyway, all of them report considerable increases in bicycle usage further pushed due to the Corona crisis in 2020 [70]. Unfortunately, as bicycle use increases, at the same time, the rate of bicycles involved in road crashes has increased, as well. Due to the vulnerability of bicyclists to serious injuries, it has been estimated that riding a bike is seven times more unsafe than traveling by car [71]. Data coming from European statistics shows that the rate of fatal accidents for cyclists on urban roads has increased from 2010 to 2018 by +6% in contrast to the decrease of all the other modes of transport (Figure.4.1) (EU Commission Road Safety – Key figures, 2020).

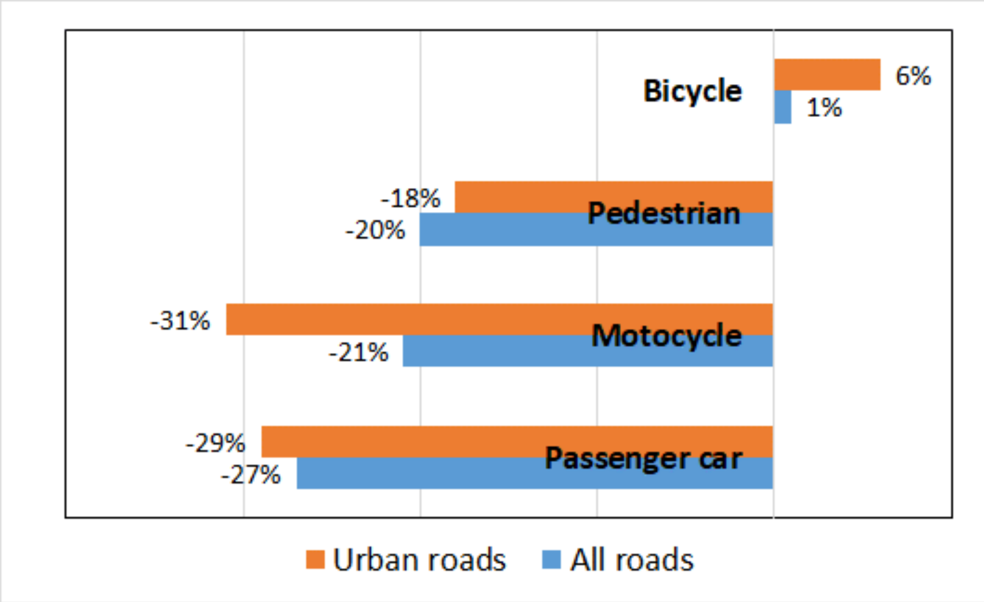


Figure 4.1: Trends 2010-2018 of fatalities in crashes involving cyclists and other transport modes. Source: (EU Commission Road Safety – Key figures, 2020)

Moreover, the high risk of crashes when perceived by the users acts as a strong barrier, dissuading people from using bicycles as a form of transport [72]. A critical area for cycling safety research is the under reporting of cyclist crashes [73][74] and the lack of reliable data about cycling traveled distance. Even in highly cycling countries, 50% of bicycle-involved traffic accidents are not reported in police statistics [75][76]. Consequently, the crash statistics are biased in the magnitude and exposure, and less feasible than for motorized vehicles [77]. Accordingly, Traffic Conflicts, near misses, or Critical Safety Events (CSEs) are spreading as crash surrogates in safety studies [78]. CSE is defined as "A traffic event that requires a rapid evasive maneuver by the subject vehicle, or any other vehicle, pedestrian, cyclist, or animal to avoid a crash. A rapid, evasive maneuver can be steering, braking, accelerating, or a combination of control inputs". Bivariate extreme value models showed that pairs of temporal and speed-related indicators should be combined in order to properly predict the severity of surrogate measures of safety [79].

Loss of control, turning, braking, and overtaking are recurrent maneuvers in CSE involving bicyclists [80][3]. These "evasive maneuvers" are "anomalies" in the normal ride behavior whose identification is a complex task due to high dimensionality and heterogeneity trajectory data (e.g. speed, acceleration, direction) for which deep learning models for anomaly detection [81] may be more efficient than traditional statistical methods [82].

In this framework, the main contribution of this study is the development of an experimental framework where convolutional neural network (CNN) deep learning is originally applied to integrate multiple GPS data streams of bicycle kinematics to detect anomalies in the cycling behavior that are associated with evasive maneuvers in the occurrence of CSEs. Validation of the results with real data and higher performance than traditional threshold-based, and statistical techniques makes the proposed approach promising in order to identify locations with potential hazards for cyclists by using mobility data that can be easily collected in smart cities and communities. A case study in the city of Catania is presented as well. The paper is organized in the following sections: (Section 2) Related background: This section includes an overview of observational studies about bicyclist safety and the application of CNN to road mobility and safety. (Section 3) Method: the overall method is presented in its different conceptual and operative modules which include Dataset preparation (GPS data collection (typology and frequency), signal smoothing and cycling parameters calculation, data labeling into normal and abnormal) and Neural Network Model Synthesis (training of the convolutional autoencoder by defining architecture and setting the model parameters to perform anomaly detection). (Section 4) Results: The detection performance methodology is validated with real SCEs; different model settings are compared, and superior performance is observed over traditional detection techniques. Validation through Case study is also carried out in Section 4 to demonstrate how results can be used in practical application. (Section 5) Conclusions about the proposed method, results, and future recommendations are reported at the end of the paper.

4.1.2 Background

Section 4.1.2.1.a we will provide an overview of observational studies in cyclist safety, while in Section 4.1.2.1.b we will focus on the research activities related to the use of deep learning and CNNs for road safety and bicyclist mobility.

4.1.2.1 A. Observational Studies

Literature is extensive about safety assessment using observational studies, but in comparison, a limited number of studies are applied to bicyclists [78]. In the InDeV project, the Safe VRU app was developed for self-reporting of accidents and near-accidents and has been used by more than 400 participants (Lund University, 2015). The target of the UDRIVE project was to identify factors in CSEs involving a bicyclist; CSEs were identified manually and correlated to the features of the infrastructure [83]. In the project BIKEALYZE, data was collected by mobile eye tracking, a GPS-based motion data acquisition complemented with acceleration and steering direction data; CSEs (e.g. collision avoidance, way-giving violations, abrupt braking, abrupt turnout) have been identified by video-based analysis and elicitation interviews [84]. In several studies, participants had an active role in indicating any Critical Safety Events (CSEs) they experienced through various means. For example, in some studies, participants used a push-button installed in the vehicle [85], a smartphone app developed by Lund University, or online questionnaires [86] to report CSEs.

A study conducted in Sweden by Dozza et al. [87] collected movement data from 20 bicyclists using an Inertial Measurement Unit (IMU) and GPS installed on instrumented bicycles, and it analyzed cycling kinematics. Notably, longitudinal and lateral accelerations were considered relevant for analyzing cycling behavior. Another study highlighted the importance of collecting GPS data with a frequency of at least 1 Hz to provide suitable speed profiles and detect hard braking by cyclists [88]. Additionally, vertical accelerations, acquired at a minimum of 50 Hz using accelerometer sensors, were found necessary to analyze cyclist comfort and safety in response to pavement unevenness [89].

Research by Mehta et al. [90] focused on motorized vehicles' overtaking behavior concerning cyclists. They measured the lateral distance between the bike and the passing vehicle and developed a statistical model to predict the probability of an unsafe critical maneuver. Additionally, they investigated cyclists' safety perception.

In a study by Candefjord et al. [91], an algorithm was developed to detect a cyclist's fall by combining acceleration and rotation thresholds. Strauss and Miranda-Moreno (2017) proposed a procedure that used cyclist GPS data captured by a smartphone to calculate decelerations and correlate thresholds with the number of injuries. Despite promising results, they concluded that further work, including more granular data and validation, is needed to enhance the reliability of the correlation.

Perceived risk in bicycle paths was consistently associated with the frequency of CSEs in a study where experts analyzed video recordings alongside speed and heading GPS data [92]. These research efforts primarily rely on the identification of safety-critical events through self-reporting, manual video review with predefined thresholds, and statistical methods for data analysis.

An extensive review [93] reported that the existing solutions for trajectory outlier detection were "algorithm-based" (e.g. distance-based; density-based; pattern mining–based) with emerging machine learning–based schemes that learn the outlier detection from the training trajectories to identify anomalies in the newly inserted trajectories. Moreover, the research focused more on vehicle mobility [94][95] and not on micro-mobility, such as bicycles, which also suffer from a "digital divide" when compared to the increasing opportunities for data collection through connected and automated vehicles. More specifically, no studies are reported for cycling trajectories [93].

4.1.2.2 Deep learning in road safety and bicyclist mobility

Recently, deep learning and Convolutional Neural Networks (CNN) have been applied in road safety studies [96] [97][98][99] and driving style analysis [100][101]. The convolutional autoencoders (CAE) allowed the extraction of valuable information from large quantities of complex and heterogeneous data, showed fast convergence due to the convolutional layers, and provided better performance with multi-dimensional data compression and feature learning making the procedure well suitable for managing the mobility data characterized by high volume, variability and velocity (i.e. big Data) [102]. Dong et al. (2016) made the first attempt of adopting a deep neural architecture, based on Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), to extract features on the driving style directly from GPS data. More recently, Bichicchi et al. (2020) applied an unsupervised Denoising Stacked Autoencoder (SDAE) to provide output layers from kinematic measures tracked with an in-vehicle 10 Hz GPS device. The RGB colors of the outcomes were associated with different path geometries encountered during the driving. When applied to cyclist mobility, deep learning, and CNN have been used in the bike-sharing prediction modeling, because the use of shared bicycles is susceptible to time dependence and external factors [103], such as weather [104][105], bike rebalancing and land use characteristics. In [106], authors applied the Self Organizing Map artificial neural network to identify atypical trajectories from video sequences at fixed locations. More recently in [107], authors used video records from fixed cameras and trajectory data extracted by means of computer vision algorithms and Advanced Artificial-Intelligent (AI) techniques to model cyclists' behavior and their interactions with pedestrians in a shared space. The limitations of the existing works for the classification of abnormal cycling behavior are summarized as follows: observational studies applied to bicyclist safety mainly rely on traffic conflict techniques applied to video tracking from fixed positions. Few studies used trajectory data to identify SCE but by self-reporting or handled classifications. In [108], authors use text mining analytics and an Artificial Neural Network (ANN) to extract information from near-miss and collision event descriptions, acquired from BikeMaps.org, a global tool for mapping collision and near-miss events. Deep learning is becoming widely applied to transport and road safety studies, but applications to cycling are mainly focused on mobility choices. Results from previous studies about CNN for anomaly detections or modeling of motorized driver behavior, cannot directly be transferred to cycling because of its specific kinematic features and limited availability of advanced equipment for data collection as in standard naturalistic studies. To the best of our knowledge, this is the first work extending the use of deep learning CNN to extract features of the riding style of bicyclists from GPS data and to detect anomaly events in cycling behavior.

4.1.3 Methodology

The overall methodology consists of different tools that comprise data preparation and convolutional Neural Network training and testing as illustrated in Figure 4.2.

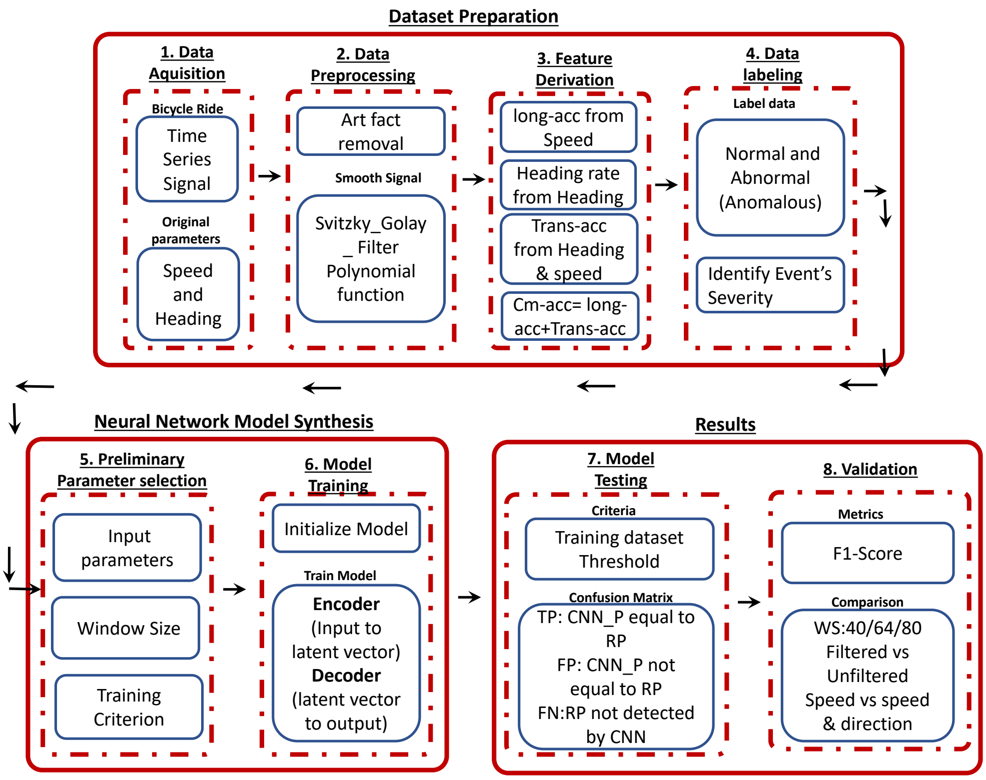


Figure 4.2: Flow diagram of method

4.1.3.1 Dataset preparation

Dataset preparation includes both the collection and treatment of GPS data to extract features optimized to train the CNN model.

a. Data Acquisition

The data source is an instrumented bicycle with GPS (Global Positioning System) and HD video system (Video Vbox Lite). The Video Vbox Lite (VVL) records an extended GPS NMEA dataset not limited only to latitude, and longitude, but including also speed and heading at 10 HZ frequency synchronized with a video recording of 2 HD cameras. Data accuracy and resolution are reported in Table 4.1. Worthily to mention that GPS in the standard acquisition without augmentation has limited position accuracy, but good data quality in speed and heading derived from the Doppler method and Carrier Phase observations [109].

Table 4.1: Accuracy and resolution of data

Data	Accuracy	Resolution
GPS Speed	0.1 km/h	0.01 km/h
GPS Heading	0.1°	0.01°
GPS 2D Position	+3 m 95% CEP*	(*) 95% CEP means 95% of the time the position readings will fall within a circle of the stated radius
GPS Time	50 nanoseconds	1 millisecond
Camera	25 frames per second	720x576 pixels

Data was collected from 10 cyclists, named from ID-1 to ID-10, who participated in controlled test rides. The ten cyclists included eight males and two females. Participants were between 27-65 years of age, whereas 40 % were over 40 years old. On average, the cycling experience of users was uniform with weekly cycling use. Only ID-4 was a highly experienced commuter cyclist with daily use of a bicycle. Participants were instructed to ride the instrumented bicycle following their normal behavior. The test was carried out in normal weather, daylight, and traffic hours. The ride path was long around 4 km, traveling different road infrastructures, to provide different traffic and road environment conditions that include cycle track, bicycle/bus shared line, cycle track termini, and one roundabout [88]. The dataset collected for each rider contains around 9000 samples at a 10 Hz acquisition frequency. Regardless of the limited number of participants such a dataset is appropriate for training purposes as we will discuss in the conclusions. CSEs that occurred during the test were identified by the research team reviewing the videos together with the test rider who explained the occurrence of an actual CSE. A total of 41 CSEs have been detected and classified over about

2.5 hours of tests. As an example, Figure. 4.3 shows a test path section with bicyclist GPS positions and time of NMEA data. The two blue dots mark the position of one CSE, while the blue boxes show the time interval of the CSE in the speed and heading profiles. Figure. 4.4 includes a screenshot of the recorded video at different times.

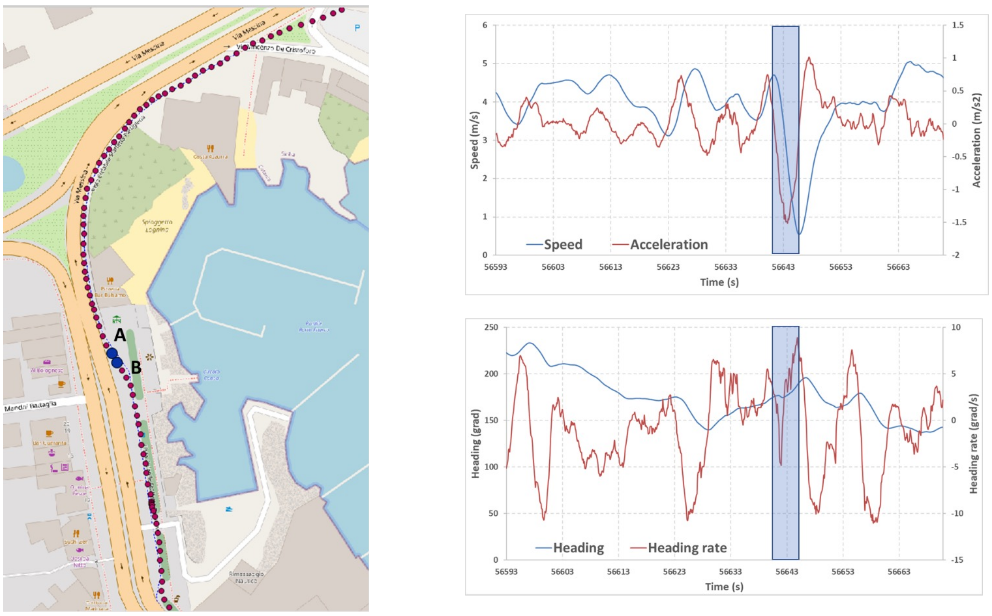


Figure 4.3: Map location along dependent and independent parameters

(a) (b)

Figure 4.4: Video screenshot at the time (A) and (B) of the CSE

b. Data Preprocessing and feature derivation

Once data was recorded, different Python routines were applied to 1) improve the data quality, 2) interpolate for smoothing, 3) calculate derived parameters and 4) create the data set for training and testing the CNN. In the present application, speed (S) and heading (H) define the recorded time data in the NMEA string, while longitudinal acceleration (LA), traveled Distance (D), heading rate (HR), transversal acceleration (TA), and combined acceleration (CA) are derived by as shown in Equations (4.1-4.5).

$$LA = \frac{S_{i+1} - S_i}{\Delta T} \tag{4.1}$$

$$HR = \frac{H_{i+1} - H_i}{\Delta T} \tag{4.2}$$

$$D = \frac{S_i + S_{i+1}}{2\Delta T} \tag{4.3}$$

$$TA = \left(\frac{S_i + S_{i+1}}{2}\right)^2 \cdot \frac{HR}{D} = \frac{(S_i + S_{i+1}) \cdot (H_{i+1} - H_i)}{2 \cdot \Delta T} \tag{4.4}$$

$$CA = \sqrt{LA^2 + TA^2} \tag{4.5}$$

where $\Delta T = 0.1$ sec, S is speed in m/s, H is heading in radians, HR is in rad/s, and LA, LT, and CA are in m/s$^{\mathbf{2}}$.

Speed and heading data from GPS have a good standard accuracy as reported in Table 4.1. Anyway, environmental factors such as satellite view, signal blockage, and atmospheric conditions can affect precision. Moreover, pedaling produces riding oscillation with frequencies around 2.5 Hz in the longitudinal speed and 1.2 Hz in the lateral direction [87] which can be considered as noise in the S and H signals, emphasized by the high-frequency rate. Therefore, we applied a Savitzky-Golay smoothing filter (SGF) to the speed and heading profiles, before calculating their derivatives (i.e., LA, HR, TA, CA). SGF is a digital filter [110] [111], well applied in GPS trajectory data of urban buses [112], that we adapted to our time series dataset of speed and heading to increase the data precision without deforming the actual signal frequencies and shape, reducing noise and determining a smoothed trend line for deriving the other parameters. Figure. 4.5 illustrates the original data parameters speed and direction (heading) and their derivate LA and HR before (black line) and after (red line) applying SGF. It is evident the improvement in the signal smoothness, especially for the derivate of LA and HR.

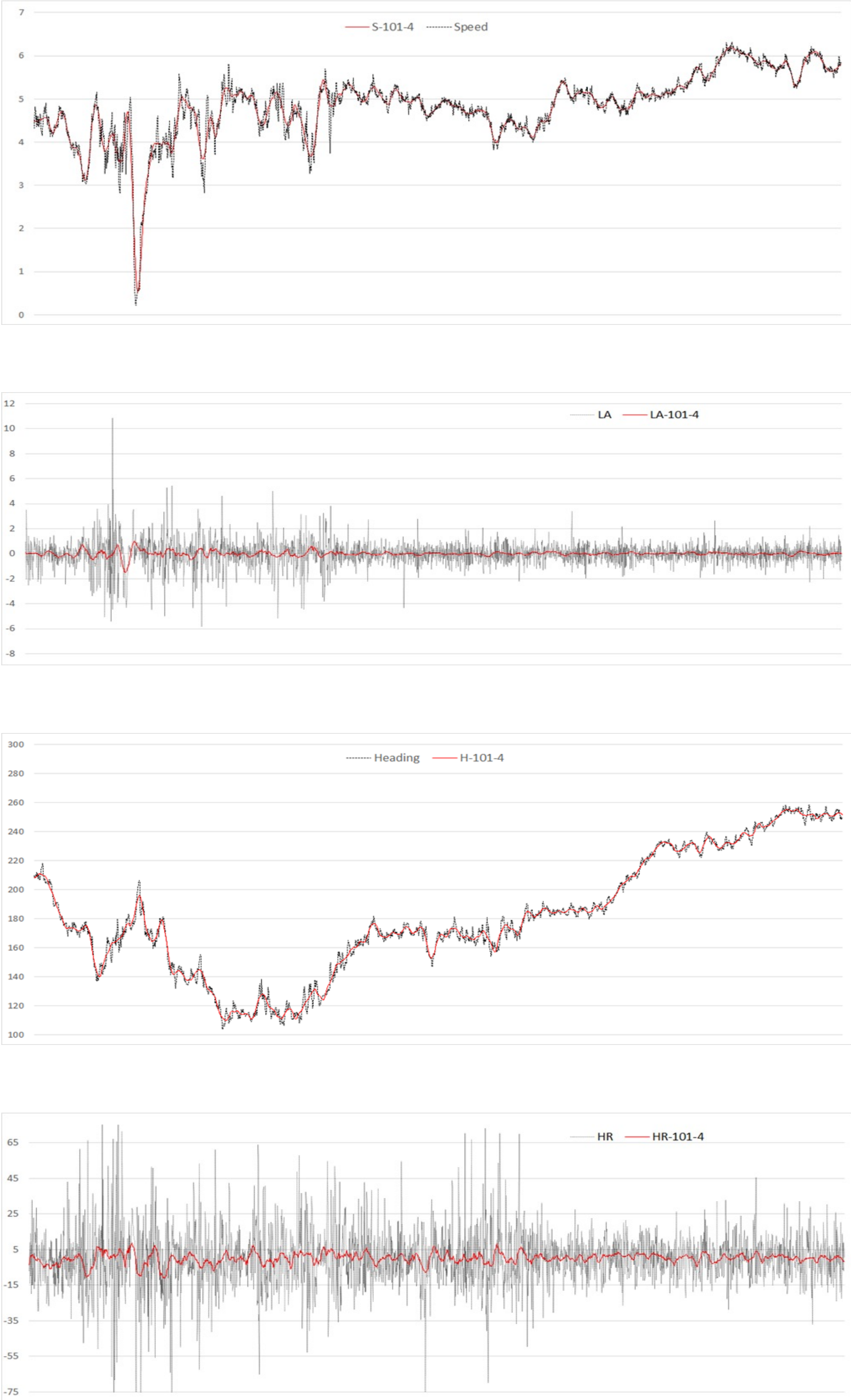


Figure 4.5: Speed, Heading with derivate LA, HR before and after SGF (101-4)

4.1.4 Neural Network Model Synthesis

In this section, we propose a methodology that exploits a convolutional autoencoder for anomaly cycling detection in time series. Therefore, we will first provide CNN background and then, a discussion on convolutional autoencoders (Section 3.2.1). Then, we will present the proposed methodology for anomaly detection in the application scenario (Section 3.2.2).

4.1.4.1 Preliminaries: Convolutional Neural Network (CNN)

Deep learning is a recent technology used in several scenarios including the identification of anomalous points [113][114]. Convolutional Neural Networks (CNNs) are types of deep learning algorithms, introduced to process images efficiently and are quite popular for anomaly detection as well [28][81].

CNNs automatically extract features from the data that are used for classification purposes [115]. The architecture of a CNN includes several layers that are classified into convolutional layers, pooling layers, and fully connected layers (Figure.4.6). The convolutional layers are the first layers of a CNN, which contain filters in the form of a weighted matrix (C1) and recognize patterns efficiently by reducing the variable's dimension. Convolutional layers are followed by pooling layers (S) which can be repeated several times to summarize features. The last layer is the fully connected layer whose neurons (NN) take the extracted features as their input as shown in Figure 4.6.

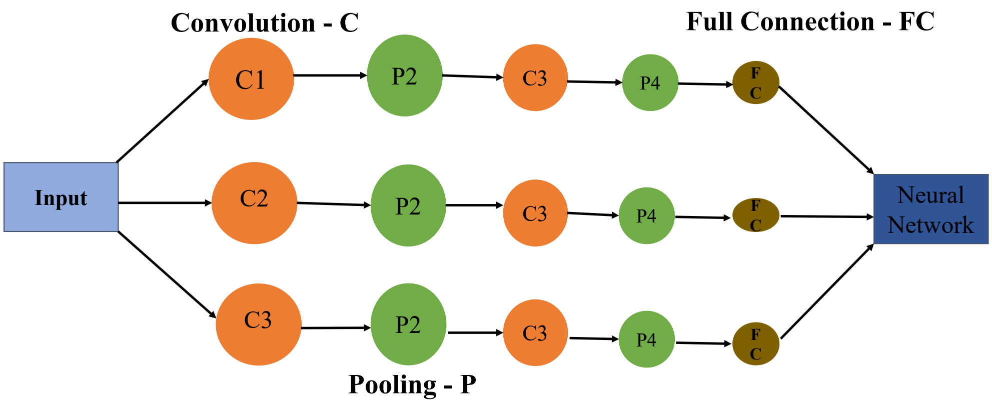


Figure 4.6: Basic CNN Architecture

In our work, we use Convolutional Autoencoders and thus, we will provide initial information on Autoencoder (AE). An AE is a type of artificial neural network and popular for anomaly detection, AE consists of two main modules: the encoder and the decoder (Figure. 4.7). The encoder maps the input data into a latent vector while the decoder tries to reconstruct the input from the latent vector. The encoder is a neural network that is specified by a set of parameters which we call w_E. As we already said, the encoder takes as input an n-tuple X and gives as output an m-tuple Z, which we call the latent vector, with $m \ll n$. Obviously, the latent vector Z is a function of the parameters w_E and the input X as shown in Equation 4.6.

$$Z = f_E(w_E, X) \tag{4.6}$$

The decoder is a neural network, specified by the parameters w_D, which takes as input the latent vector Z and gives as output an n-tuple $\hat{X}$, i.e., Equation 4.7. The AE training minimizes the difference between the input X and the model $\hat{X}$ as shown in Equation 4.8:

$$\hat{X} = f_D(w_D, Z) = f_D(w_D, f_E(w_E, X)) \tag{4.7}$$

$$\hat{X} \approx X \implies f_D(w_D, f_E(w_E, X)) \approx X \tag{4.8}$$

Where $\|\hat{X}, X\|$ represents a measure of the difference between $\hat{X}$ and X as shown in Equation 4.9. Several ways of measuring such differences can be applied. Notable examples include the Mean Absolute Error (MAE), the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE). In our work, we applied the MAE because it gave the best performance. This was expected; in fact, MSE and RMSE square errors before averaging, and therefore, they give higher weight to large errors. We opt for MAE over Mean Square Error and Root Mean Square Error (RMSE) due to data distribution and error size suitability. Therefore,

$$MAE = \|\hat{X}, X\| = \frac{1}{n}\sum_{i=1}^{n} |\hat{X}_i - X_i| \tag{4.9}$$

Note that convolutional autoencoders (CAE)s are capable of learning the most useful feature patterns in the input data [116] and anomaly detection [81].

4.1.4.2 BeST-DAD Model: The Proposed CNN application for Anomaly Detection

We call the complete scheme proposed for anomaly detection in the scenario of interest: 'Bicycle Safety through Deep learning-based Anomaly Detection' (BeSt-DAD). Best-DAD employs a 1-D CAE as depicted in (Figure. 4.7). The input consists of a sequence of time-data samples $X_1, X_2, \ldots, X_6$ generated at a frequency of 10 Hz. The generic X_i is a 6-tuple of values, i.e., $X_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5}, x_{i6}]$ which represent the speed, heading, heading-rate, longitudinal acceleration, transversal acceleration, combined acceleration, calculated as discussed in the previous Section 3.1. Thus, the input data is a 2-dimensional matrix in nature, as shown in Fig. 6, that we flatten as a 1-dimensional input sequence of type as follows:

$x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{21}, x_{22}, x_{23}, x_{24}, x_{25}, x_{26}, \ldots$

The number of samples j composing the input sequence as window size. Experiments show that a good value for j is $j = 40$. Therefore, the input size of the encoder is $n = 40 \times 6 = 240$.

Figure 4.7, shows the overall architecture of BeSt-DAD where the encoder consists of 2 convolutional layers. We applied Stride as an advanced convolutional parameter which is capable of replacing max pooling with less computation. Padding is used to maintain the output dimension as input while the activation function is responsible for neuron activation. In our case, each convolutional layer reduces the input dimension of a factor equal to the stride, i.e., four. As a result, the output of the first convolutional layer has a dimension equal to 60, whereas the output of the second layer, that is, the latent vector Z, has a dimension equal to m=15. The decoder consists of two de-convolutional layers and a dropout layer which avoids model overfitting. The output of the decoder will have a dimension again equal to 240 and compared to the input by calculating the mean absolute error (MAE). If this MAE is higher than a given threshold an anomaly warning is issued. Observe that the value of such a threshold is a critical parameter. We will discuss how to select it in the next section.

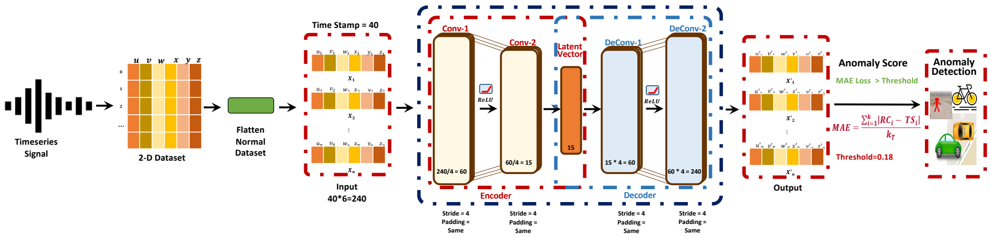


Figure 4.7: BeST-DAD scheme

4.1.5 Results

In the following section, the proposed scheme will be validated by comparing the actual CSE detected as discussed in Section 3.1.1, which we call "real positives", to the anomalies in cycling behavior detected by BeST-DAD, which we call "CNN positives" (section 4.2). For comparison analogous validations have been carried out by using Principal Component Analysis as a robust standard statistical approach for feature reduction and anomaly detection, and the more widespread method based on setting a threshold in the longitudinal acceleration to identify hard breaking. Results are presented and analysis is done in Section 4.3 and 4.4, based on quantitative performance metrics presented in Section 4.1.

4.1.5.1 Performance metrics

In a binary classification, ''Positive" and ''Negative" assignments refer to the classifier's prediction, and the terms ''True" and ''False" refer to whether that prediction corresponds to the real observation. Given these definitions, the confusion matrix (CM) describes the performance of the classification model as shown in table 4.2

Table 4.2: Confusion Matrix (CM)

	Real Positive	Real Negative
CNN Positive	TP	FP
CNN Negative	FN	TN

CM is useful for calculating two metrics of classification performance called Recall and Precision. Precision (P) measures the rate of true positive (TP) over the total predicted positive ($TP + FP$). The Recall (R) computes the model's ability to detect TP over the total number of real positives ($TP + FN$) as shown in Table 4.3. For our classification with unbalanced data due to the small number of real positives, the F-measure (F_β) is an effective quantitative metric to select the model setting that minimizes the errors [117]. The F_β score, in Equation 4.10, is the weighted harmonic mean of precision and recall, ranging between 0 and 1.

$$F_\beta = \frac{(1+\beta^2) \cdot P \cdot R}{\beta^2 \cdot P + R} \tag{4.10}$$

As commonly used to emphasize Precision against Recall, we applied a weight $\beta = 2$, because we are more interested in limiting FN (i.e., missing detection of CSEs) rather than FP.

4.1.5.2 Criteria for classification of CNN-Positive

Our event detection criteria are illustrated in (Figure.4.8) and defined as follows. Considering the observed time extension of a real CSE in the range of 0.8-3.1 secs with an average of 1.4 secs and the high time variability of the kinematic parameters in the cyclist riding, events with MAE>threshold of less than one second were not classified as "CNN positive" and detection sequences within 5 seconds have been classified as one CNN positives.

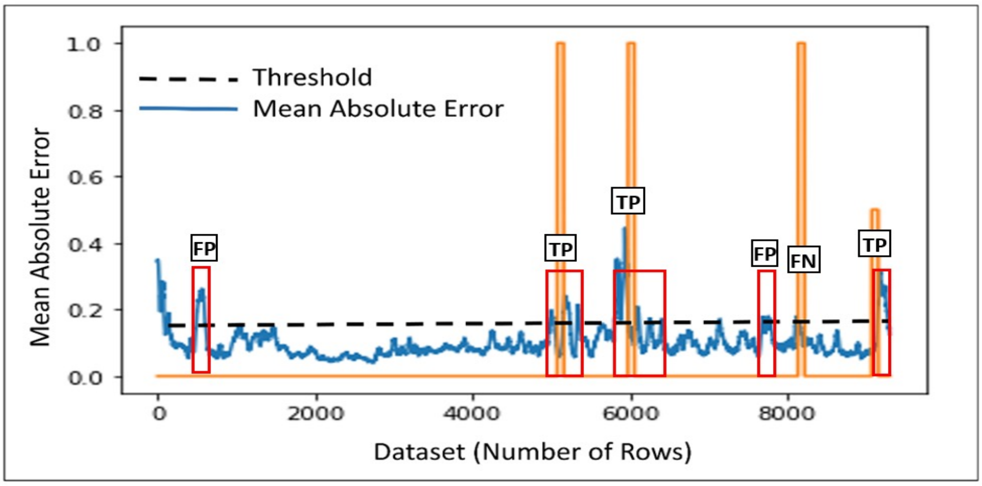


Figure 4.8: Anomaly identification example. Orange bar: Real positive; red bar: CNN positive

4.1.6 Model Testing

The model performance measured by R, P, and F2 is affected by several factors related to the data and CNN setting and varies by changing the classification threshold. Therefore, to evaluate the results, two major comparisons are carried out: the first comparison case is carried out by using various model settings and features, while another comparison case is done between BeST-DAD and alternative detection approaches like Principal Component Analysis (PCA) and breaking acceleration threshold. To maximize the performance of the model and to learn new insights about cycling behavior modeling, model setting and features relate to the following comparisons: CNN setting by changing (A) Training and testing, (B) Thresholds, and (C) Size of the time windows; input dataset by varying (D) Use of SGF filter and (E) Use of only speed-related variables: speed and longitudinal acceleration or (F) Use of only heading related variables: heading and Heading rate Furthermore, two scenarios have been selected for training and testing of the model: A1) Training and Testing CNN for each user, by using 80% of data for training and 20% of data for testing. The average training time in scenario A1 is 30.142 seconds for each user. A2) Training with the data collected by considering the entire dataset related to only one user and testing with 100% of the data from all the other users. The average training time in scenario A2 is 26.642 seconds. Results and values of Recall, Precision, and F2 are reported in Table 3. For the first scenario (A1), where each user adopts its own model, we reached the absolute best performance with an average F2=0.77. In the second scenario (A2), the model is trained with the full dataset data of one-by-one users and tested with 100% of data from the other users with an average F2=0.72. Anyway, training with ID-4 as a reference user and testing with all other users gave the best F2 score of 0.77. While comparing the two different training approaches, we applied the best CNN settings in Table 3, with SGF, 40 TS-window, and all input parameters. Results in Table 3 for scenario A1 show that training tailored CNN models for each user returned the best results. Anyway, in practical application, scenario A1 means the need to train the BeST-DAD model for each user sharing his/her cycling data. In scenario A2, transferring the model trained on one user returned slightly worse performances, but an overall average F2 comparable with the previous scenario, as well.

Scenario A1: Training = 80% of dataset, Testing = 20% of dataset, Preliminary Selection: Filtered data with TS_window size 40

User	T	RP	CNN	Performance Metrics		
				Recall	Precision	F_2
ID-1	0.21	3	4	1	0.75	0.94
ID-2	0.21	4	3	0.75	1	0.79
ID-3	0.21	1	2	1	0.5	0.83
ID-4	0.21	0	3	-	-	
ID-5	0.21	1	1	1	1	1.00
ID-6	0.21	3	3	1	1	1.00
ID-7	0.21	1	2	1	0.5	0.83
ID-8	0.21	2	3	0.5	0.33	0.45
ID-9	0.21	1	2	0	0	0.00
ID-10	0.21	2	2	1	1	1.00
Weighted Average	**0.21**	**18**	**26**	**0.83**	**0.60**	**0.77**

Scenario A2: Training = 100% dataset of any one user, Testing = 100% dataset of all other users, Preliminary Selection: Filtered data with TS_window size 40

Users for training	T	RP	CNN	Performance Metrics		
				Recall	Precision	F_2
ID-1	0.18	33	47	0.76	0.53	0.70
ID-2	0.18	37	39	0.70	0.67	0.69
ID-3	0.18	38	47	0.76	0.62	0.73
ID-4	**0.18**	**41**	**43**	**0.78**	**0.74**	**0.77**
ID-5	0.18	38	46	0.74	0.61	0.71
ID-6	0.18	35	47	0.77	0.57	0.72
ID-7	0.18	39	49	0.80	0.65	0.76
ID-8	0.18	34	41	0.76	0.63	0.73
ID-9	0.18	37	45	0.73	0.60	0.70
ID-10	0.18	37	41	0.65	0.59	0.64
Weighted Average				**0.75**	**0.62**	**0.72**

Figure 4.9: Table 3. F-Score based performance evaluation for proposed scenarios.

The previous result illustrates the reason of "ID-4" selection as 100% training dataset for the further comparisons, given the good performance and availability of a larger dataset for testing purposes (i.e., 100% of data for user ID-1, ..., ID-9) by using the full set of 41 real-positives for testing. Results for the other validation scenarios (B, C, D, E, F) are presented in Table 4. First, we tested CNN settings for different thresholds (T) and time window sizes (TW) (scenarios B, and C). Results in Table 4 confirm the best performance when T=0.18 and TW=40. The best values of TW and T also have meaning in explaining the cycling behavior. MAE=0.18 is equal to the 88th percentile of the overall Loss values, while MAE=0.21 is the 93rd percentile, showing that anomaly cycling behaviors for evasive maneuvers are quite rare (12% of the cycling time), but not exceptional events in the riding path especially in shared lanes as will be highlighted in section 5. Without the application of the SGF filter the high-frequency time-variability of the cycling data returned many CNN-positive with high Recall, but also many FPs returning a very low Precision (P=0.36). It is worth noting that merging speed and heading parameters had the most important impact on improving the model performance. Results showed that without including heading-derived parameters (i.e., HR, TA, CA), the model performance decreases significantly by missing several detections with more FNs and FPs (low R and P in Table 4). That is expected for cyclists, rather than other road users, because they apply both braking and swerving as evasive maneuvers. Analyzing results, when Precision and Recall are compared it is noteworthy that Recall is always higher than Precision. This is significant because, in our application scenario missing CSE (FN) is of higher concern than False Positive. Moreover, FP may not be the wrong detection of cycling anomalies but often have been identified as changes in cycling behavior related to other events not classified as CSE (e.g., hard braking at traffic lights, avoiding pavement bumps, potholes, etc).

Table 4.3: Comparison results for different model settings and existing approach.

Validation	Threshold	Recall	Precision	F2
B1) Threshold = 0.15	0.15	0.80	0.54	0.73
B2) Threshold = 0.21	0.21	0.46	0.68	0.49
C1) TS_window_size = 64	0.18	0.68	0.59	0.66
C2) TS_window_size = 80	0.18	0.56	0.59	0.57

Continued on next page

Table 4.3 – *Continued from previous page*

Validation	Threshold	Recall	Precision	F2
D) No SGF Filter+ speed and Heading parameters	0.18	0.95	0.36	0.72
E) SGF+Only Speed parameters	0.18	0.61	0.57	0.60
F) SGF+Only Heading parameters	0.18	0.54	0.67	0.56
A1) Training each user (80/20) TS_window_size=40, SGH+ Speed parameters + Heading parameters	0.21	0.83	0.60	0.77
A2) Weighted Average of A2		0.75	0.62	0.72
PCA – training each user (80/20), TS_window_size=40	0.17	0.66	0.34	0.45
PCA – ID-4 training (100%), TS_window_size=40, SGH+ Speed parameters + Heading parameters	0.17	0.49	0.44	0.47
Breaking Threshold - ID-4 training (100%), only Speed parameters	variable	< 0.30	< 0.30	< 0.30

Finally, to further evaluate the effectiveness of deep learning in detecting CSEs, the performance metrics have been calculated also by applying a robust PCA statistic and the traditional empirical approach based on breaking acceleration threshold [118]. Results in Table 4 confirm the higher performance of deep learning for event classification.

4.1.7 Validation through Case Study and Risk Assessment

In order to show the practical results of BeST-DAD for risk assessment and ranking, the procedure was applied to the trajectory data and anomaly detections located with the GPS coordinates in the different roadway sections composing a mixed cycling path (i.e. cycle tracks, roundabouts, cycle track termini, and bicycle/bus shared lane) (Fig. 4.10).

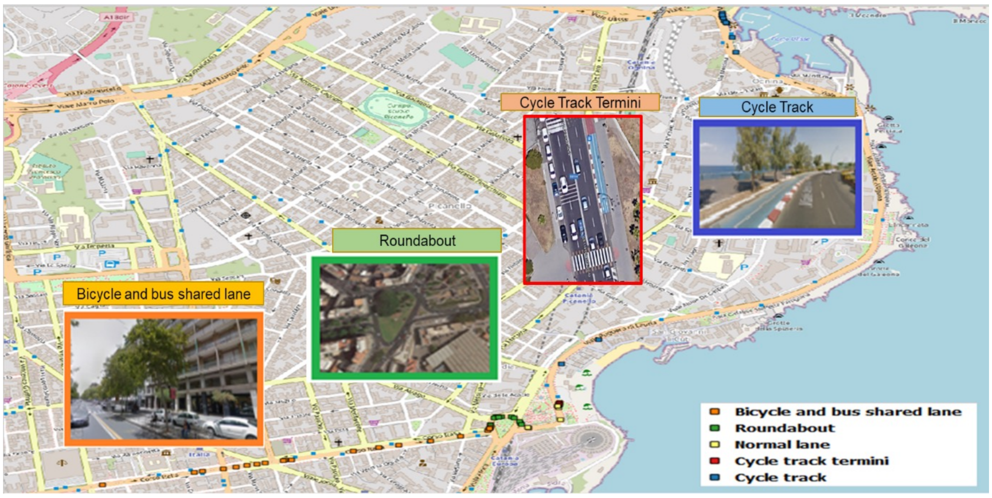
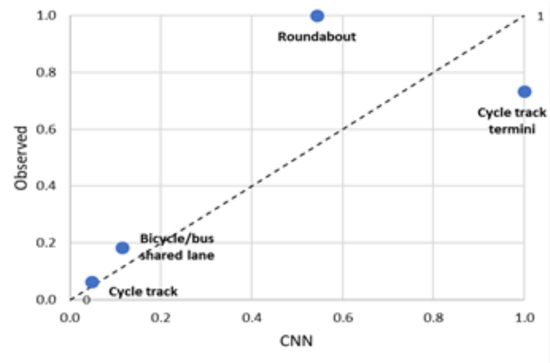


Figure 4.10: Map of BeST-DAD anomaly detections

The exposure to the occurrence of a conflict rises with the time the bicyclist spends in the road section and therefore cycling time was considered as an exposure metric to rate the number of anomalies among different roadway components to make the results comparable. Risk Rate = N. of anomalies/cycling time. The mean cycling time is reported in Figure 4.11 which also shows the total number of BeST-DAD anomaly detections in the different road sections traveled during the test and the comparison between observed and CNN risk rates in a normalized scale (0-1).



Roadway component	Travel Time (sec)	Number of anomalies	Risk Rate
Roundabout	44	12	0.27
Cycle track termini	10	5	0.50
Bicycle and bus shared lane	278	16	0.06
Cycle track	467	11	0.02

Figure 4.11: Travel Time and Risk Rate in various Road Typologies

Figure 4.11 shows cycle track termini at the highest risk rank followed by the roundabout. The normalized Risk Rate is also calculated to allow for comparisons with the observed risk rate reported in the previous study [92]. It should be noted that in Figure 4.11, the very good consistency between the two risk ratings for cycling track and shared lane. Also consistent are the higher ratings for lane termini and roundabout, even if BeST-DAD has classified a higher risk at the Cycle track termini than the one observed by actual CSE. The cycle termini end with a sharp curve before the lane crossing (Fig. 4.11) where cyclists were required to steer and often stop showing as anomalies in riding behavior (e.g hard braking and steering) that have been correctly detected by BeST-DAD although not specifically related to traffic conflicts and therefore classified as FPs.

4.1.8 Summary of 4.1

Cyclists are vulnerable road users, and their safety is a serious issue to be addressed with an increasing number of fatalities among cyclists. Due to the lack of reliable data for crash analysis and the opportunities to collect new data in smart cities and bicyclist communities, innovative observational studies can offer new approaches for a network-wide safety assessment of VRUs consistent with the EU directive. Auto-encoders are mainly utilized for dimensionality reduction, feature extraction, image denoising, anomaly detection, and image compression [101]. The authors- knowledge that is the first attempt to use both speed and direction GPS data with customized Convolutional Autoencoder to automatically detect anomalies in cycling behavior that can be associated with critical safety events (CSE) and plotted on a map as risky points. Performance of the classification was very good considering the low rate of FN with a Recall of 100% in 6 out of 9 tests after individual training of the model (Table 3). Furthermore, in scenario A2 (Section 5), we have seen that a model trained using the dataset for one selected rider can be effectively transferred to the other riders with R=0.78 and F2=0.77. This result is interesting because, in large-scale applications, the use of a pre-trained model results in the reduction of communication and energy resources and is more suitable for protecting user privacy. Performance evaluation of BeST-DAD for different model settings (Table 4.3) demonstrates that adding direction information (heading, heading rate, transversal acceleration) to the more traditional only speed parameters (speed and longitudinal acceleration), improved the capability of the model to detect anomalies in cycling. Data filtering by using SGF played a positive role in reducing the FPs, although CNN showed good capability to handle noise and extract features from raw input data as we observed the weighted average of scenario A2 of Table 3. The advantageous application of CNN was also proven by the best performance of the proposed

BeST-DAD over other traditional statistical techniques like PCA or the heuristic threshold-based method applied to the cyclist braking rate. A case study showed the practical application and consistency of risk assessment and ranking.

4.1.9 Lessons learned and future needs

Despite the good performance of the CNN trained on the reference cyclist, we can expect larger deviance increasing the number of users. Because the CNN model depends on both the user and the specific road environment, transfer learning, and cooperative learning can be applied in real-time to the model trained and transfer its knowledge to the specific user and road environment. To model the cycling behavior, our study used extended GPS NMEA contents (i.e. Speed, Heading) at a high 10 Hz acquisition frequency which is not common in present smartphone and data mobility-data providers (e.g. Strava) following mainly the targets of profiling users' destinations and flows (e.g. positions at 0.2 Hz) or fitting activity (e.g. distance, elevation). Frequencies up to 1 Hz are not yet available in smartphones, moving mainly in the direction of improving the localization accuracy while already appropriate are speed and heading. To achieve a suitable higher frequency, an alternative to be evaluated is the capability of using SGF also for sub-sampling the GNSS signals at higher frequencies than the actual sampling rate [88].

4.2 Deep Transfer learning exploitation for anomaly detection

4.2.1 Objective

Cycling is one of the most sustainable and green transportation modes. These advantages include the relief of congestion, the reduction in emissions, and improvements in the well-being of cyclists [119]. Many people worldwide have been switching to bicycles, especially to e-bikes, and cycling has increased even more due to the COVID-19 pandemic [120]. Unfortunately, with the growing rate of bicycle usage, bicycle-based road crashes also increased. Bicyclists face a high risk of serious injuries, and, taking into account unreported bike incidents, it is estimated that the collision rate for bikers is nearly 20 times that of car users [121]. Among all categories of road users, cyclists stand out as the only group that has not experienced a decrease in fatalities since 2010, with the proportion of cyclists within the total number of road fatalities growing from 7% in 2010 to 9% in 2019 [122] as shown in Figure 4.12. The fatalities in crashes involving cyclists are virtually always the cyclists themselves (98%). Also in crashes involving other vulnerable road users than cyclists, 9 out of 10 fatalities are the vulnerable road users themselves [122]. Therefore, cyclists as vulnerable road users need special attention and protection as a priority.

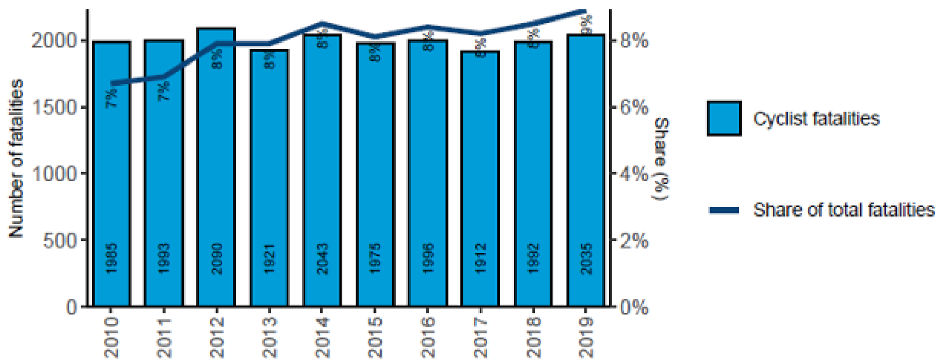


Figure 4.12: Annual number of cyclist fatalities, and their share in the total number of fatalities in the EU27 (2010-2019). Source: European Road Safety Observatory, 2021

Given that actual bicycle, crashes remain a rare event, widely scattered in the road network and often not fully reported [74], Traffic Conflicts (TCs) are most commonly used as a surrogate measure for safety analyses [123] [124][125]. In the state of the art, near miss or traffic conflict is defined as a situation in which a cyclist is required to act to avoid a crash, such as braking, speeding, swerving, or stopping [Ibrahim, 2021]. In such a framework, unusual changes in speed and directions can be assumed as "anomalies" in the normal cycling behavior [101].

Then the main issue is how can we detect such anomalous riding events? The most diffused applications make use of video recording and tracking from fixed cameras or, more recently, drones which are able to cover a limited area (e.g. an intersection) [79]. Despite being effective, this approach cannot be used to investigate the overall extension of the road network. While technology offers wide opportunities to increase the level of bicycle smartness to monitor the cycling behavior and surrounding road environment (using networking technologies, GNSS, accelerometers/gyroscopes, Light Detection and Ranging (LIDAR), speed and pedal sensors, radar, and ultrasonic sensors, cameras) [120][86] the identification of these anomalies remains a difficult task due to the high dimension and heterogeneous nature of the collected data making traditional statistical methods not adequate [82]. In this area, machine and deep learning-based models are more versatile and effective for anomaly detection [81], [126].

In this context, the main contributions of this work are:

- Identifying the most appropriate data in the NMEA string and treatment by using GNSS as a source is quite easy and highly suitable in smart communities of bicyclists or bike-sharing systems.

- Developing an experimental framework that exploits "unsupervised deep transfer learning" to detect anomalies in the cycling behavior, showing that anomaly detection using transfer learning solves the issues of data preparation and labeling and reduces training complexity to adjust the model for different users.

- Quantitatively demonstrate the opportunities of using transfer learning for anomaly detection by showing its practical application in a selected scenario in the city of Catania (Italy).

Let us stress that the major contribution of this paper when compared to the previous literature is that the focus is on the analysis of the effectiveness and the resulting benefits of transfer learning to obtain accurate models tailored to each user, increasing the performance of the models and reducing dramatically the need for data.

The paper is organized as follows. Section 4.2.2 provides an overview of the state of the art in bicycle safety studies by using traffic conflict and artificial intelligence. In Section 4.2.3 we describe the proposed methodology which we assess in Section 4.2.5 where the findings from the experiments are explained. Lastly, in Section 4.2.7 we draw our conclusions.

4.2.2 Background

Machine learning is emerging as a powerful tool also in the field of road safety and it has become crucial to analyze the complex and heterogeneous data that are today available from new technologies. Anyway, its application to bicycle safety is still limited in the wider field of road safety.

In our previous work [126] we have presented a *convolutional neural network* (CNN) to identify abnormal patterns in cycling behavior that showed promising performances. The present paper represents an advancement in the modeling which applies a transfer learning technique to adapt the general model to the specific users' cycling style. This new publication gives also the opportunity to update the state of the art in this field.

A recent review paper [123] confirmed surrogate measures of safety as a complementary approach to traditional crash studies and an emerging theme in bicycle safety research, but mainly based on video analysis. Analogously, another extensive literature review [127] identified several papers that used instrumented bikes for studies on traffic conflicts and their causes, but they were mainly based on observational and traditional statistical methods and not all specifically related to road safety. Ibrahim M.R. et al. (2021) reviewed the current methods, challenges, and potential of AI-embedded systems in analyzing cycling near misses [128]. The study selected 19 studies (in Web of Science, Google Scholar, and Scopus (2010-2019) emphasizing the significance of accurately detecting near-miss events to enhance risk assessment and improve cyclist safety. The authors identified in manual labeling of data, the limited scope of studies to specific types of near misses, and the functionality of sensors as the main gaps in the literature for near miss identification. It is worth mentioning that the present paper addresses all these issues by using an unsupervised deep learning approach for anomaly detection which does not need labeling and definition of near miss typology. Moreover, the method is based on standard NMEA GNSS data.

Our search in Scopus, on June 8th, 2023, revealed, in the last 15 years (2009-2023), 4,383 articles focusing on bicycle safety, of which 127 were by using traffic conflict techniques, but only 15 with the application of AI modeling and mainly in the most recent years (8 of 15 from 2020). Moreover, after review, six of them were further excluded because not specifically pertinent to the topic.

Although not exhaustive, these quantitative results highlight the increasing interest in this field, also the usefulness of exploring applications of deep learning techniques to identify critical safety events based on data collected with instrumented bicycles, because of the specific mobility patterns and limitations in the availability of sensors when compared to cars and other motor vehicles.

The state-of-the-art based on the selected 9 publications explores the significance of machine learning in domains of cycling safety including rider behavior analysis, road-user interaction, and personal mobility driving.

Kwayu K.M. et al. (2022) presented a methodology for automatically extracting topics from crowdsourced cyclists' near-miss and collision reports using text mining and artificial neural networks [108]. The approach enables efficient analysis of cyclist experiences and contributes to identifying common risk factors. Kozu R. et al. (2017) proposed a user-participatory approach for constructing open hazard data to prevent bicycle accidents. The study integrates participatory sensing and machine learning techniques to collect and analyze user-generated data, highlighting the importance of user involvement in identifying and mitigating risks for bicycle safety. Both the previous studies are based on user-subjective risk assessment and self-reporting of traffic conflicts.

Rostami A.D. et al. (2020) focused on predicting critical bicycle-vehicle conflicts at signalized intersections [129]. The study aimed to develop models that can anticipate potential conflicts that have been labeled using the Post Encroachment Time (PET) by utilizing video frames and machine learning techniques. Alsaleh R. and Sayed T. (2021, 2022) proposed a multi-agent adversarial inverse reinforcement learning approach to model cyclist-pedestrian interactions in shared spaces [107] [130]. The studies demonstrate the application of video analysis to track users' paths and machine learning techniques in modeling complex interactions and optimizing safety measures in mixed-use environments. Gu Y. et al. (2019) developed a deep learning framework for classifying cycling maneuvers [131]. By utilizing a video survey, different models were compared (i.e., multi-Logit, artificial neural network, support vector machine, random forest, and gradient boosting decision tree), and CNN exhibited superior performance in the classification of 5 pre-defined cycling maneuvers (i.e., passing, avoiding, carriageway-occupied, sidewalk-occupied, and regular riding). Karakaya A.-S. et al. (2023) proposed Cycle Sense, a system for detecting near-miss incidents in bicycle traffic using mobile motion sensors (i.e. GPS coordinates, accelerometer, and gyroscope) [132]. The pre-processing of data needs a manual label-cleaning procedure. Therefore, the machine learning classification techniques partially automated the detection of near-miss incidents. All these papers applied classification.

All these studies show that deep learning and computer vision algorithms can be integrated to identify near-miss scenes and types of near misses. The only limitations are the need for installation and maintenance of camera systems, the lighting conditions, and the limited area covered by the cameras.

From a more general point of view, these studies demonstrate the state-of-the-art advancements in applying machine learning techniques to detect near misses crash by using different sensors (e.g. fixed cameras and instrumented bicycles) and different type of information based on physical features (e.g. speed, trajectory) or user risk perception and reporting. Naturalistic studies with instrumented bicycles have the potential to cover an extensive part of the road network, while site video analysis is severely limited in its ability to capture only local factors. Even though the instrumented bicycle approach makes it possible to collect rich data related to near misses, the most applied classification approach needs to label the data that is time-consuming and focus only on certain types of near misses. A combination of naturalistic studies with self-reporting has the potential to capture the broadest range of information.

4.2.3 DTL_AD Methodology

This research work consists of various steps such as data collection, data preparation, model optimization, testing, and result validation. Mainly this section consists of two phases. The first phase illustrates the procedure of data collection, and data pre-processing steps, which are applied to the collected GNSS data to extract suitable features while the second phase explains specifically CNN model training.

4.2.3.1 Data Collection

The source object of data collection is an equipped bicycle with an HD video system (Video Vbox Lite VVL) as shown in Fig. 4.13. The Video VBOX Lite (VVL) is a standard GPS single frequency with an accuracy of 3m in the position and 0.1km/h in the speed. The VVL provided a time series signal including GPS data, speed, and acceleration, synchronized with a video recording, with a sampling frequency of 10Hz. One camera recorded the front view of the cyclist while the other one was forward facing. The videos were recorded with a resolution of 720 x 576 pixels at 25 frames per second [88]. The experimental dataset is gathered from 10 bicycle users with uniform cycling experience. Furthermore, the participants were instructed to ride in a normal way. The ride duration for each test was about 15 minutes. During the ride, the conflicts between the test rider and another road user have been identified by reviewing the video records. The research team, along with the test rider, reviewed the videos together to identify the TCs or real anomalies that occurred during the test. A total number of 41 TCs have been identified over 2.5 hours of ride test.

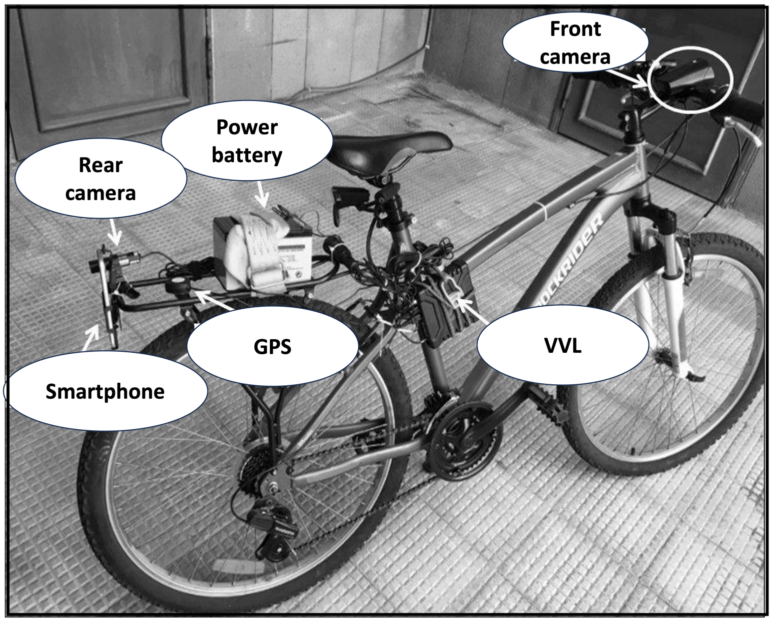


Figure 4.13: various data collection components

4.2.3.2 Data Preparation

In general, Various Python techniques were employed after data collection to process and analyze the collected data. Some of these techniques include data filtering, smoothing, feature extraction, data visualization, dataset splitting, and machine learning algorithms by exploiting built-in Python functions and libraries.

In our case, after data collection, various Python techniques were utilized such as:

1. Svitzky Golay Filter (SGF) to enhance the quality of the recorded data by employing smoothing methods.

2. feature extraction to Compute other parameters based on the collected data parameters.

3. train/test split method to Create a dataset that could be utilized for training and testing a Convolutional Neural Network (CNN) model.

In detail, this research work includes speed SP and heading HD as the basic recorded time series data in the NMEA string. Before the feature extraction process, it was observed that SP and HD have noise. Therefore, we planned to remove noise at first from basic recorded signals i.e. SP and HD. After exploring various filters for time series data, we opt for Savitzky-Golay-Filter (SGF) to apply basic and achieve smooth time series data. SGF is a kind of digital filter used to apply on GPS trajectory data [133][134].SGF gives smooth spline by fitting data points into polynomial function [134]. Here, the polynomial function is a filter kernel of SGF as stated in the Equation (4.11).

$$P(\vartheta) = \sigma_{j=0}^{j=n}\omega_j\vartheta^j = \omega_0 + \omega_1\vartheta^1 + \omega_2\vartheta^2 + \omega_3\vartheta^3 + ... + \omega_n\vartheta^n \tag{4.11}$$

By utilizing the fundamental filtered data parameters SP and HD, additional data parameters are computed for the CNN model and more promising results. For example, the heading rate HDR is derived from the HD parameter, while the longitudinal acceleration LA is derived from the speed parameter SP as shown in Equations (4.12 and 4.13). Moreover, the transversal acceleration TA and combined acceleration CA are obtained by considering both the speed and heading parameters, as depicted in the Equations (4.14 and 4.15).

$$HDR = \frac{(HD_{(i+1)} - HD_i)}{\triangle T} \tag{4.12}$$

$$LA = \frac{(SP_{(i+1)} - SP_i)}{\triangle T} \tag{4.13}$$

$$TA = \frac{[(SP_i + SP_{i+1})/2]\ ^2.HDR}{D} = \frac{(SP_i + SP_{i+1}).(HD_{i+1} - HD_i)/2.HDR}{2.\triangle T} \tag{4.14}$$

$$CA = (LA^2 + TA^2)^{0.5} \tag{4.15}$$

where $\triangle T = 0.1$ sec, SP in m/s, HD in radiant and CAin m/s^2.

4.2.4 Deep-Transfer learning

The proposed methodology exploits the *transfer learning* concept applied to the model of a convolutional autoencoder. As shown in Figure 4.14, Convolutional autoencoders (CAE)s combine the convolutional layers of CNN and autoencoder concepts which we explain in the following Sections 4.2.4.1 and 4.2.4.2.

4.2.4.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is an artificial neural network used to identify patterns in datasets by exploiting the property of the convolution operation.

Its applications range from image processing and automatic feature extraction to anomaly detection. The architecture of a CNN includes three main groups of layers: the input layers, the hidden layers, and the output layers. The hidden layers, which are typically the largest layers, consist of convolutional layers, connected layers, and pooling layers. Convolutional layers are responsible for transforming the input data stream into feature maps and passing them to the next level layer. While the pooling layers reduce the dimensionality of the data. The fully connected layer at the end is responsible for connecting the neurons of one layer to another and classifying the data.

CNN is composed of multiple layers of neurons. Each neuron operates using a mathematical function, $f(\cdot)$, with input M and output P as shown in eq. (4.16).

$$P^{(k)} = f(M^{(n-1)} * w^{(n-1)} + c^{(n-1)}) \tag{4.16}$$

In the eq. (4.16), w represents the weight vector, a set of nonzero weights, which determine the strengths of connections between neurons in the network. Each input in a neural network has a corresponding weight, which influences the sharpness of the activation function. The bias, c, is a constant added to the function to control the triggering of the activation function. There are several activation functions to choose from, including Rectified Linear Unit (ReLU), sigmoid function, and hyperbolic tangent function. The activation function is a crucial component in the neural network, as it is responsible for transforming inputs into outputs.

4.2.4.2 Autoencoder

An Autoencoder (AE) is a type of Artificial Neural Network (ANN) that has two main components: the encoder and the decoder. The encoder part of the AE compresses the input data into a compact form known as the latent vector or bottleneck. The decoder's task is to reconstruct the output from the latent vector and compare it to the original input. To evaluate the performance of the decoder, the reconstruction error (Loss), which measures the difference between the reconstructed output and the original input, is calculated. Just like other feed-forward neural networks, the AE uses backpropagation during training to minimize the reconstruction error.

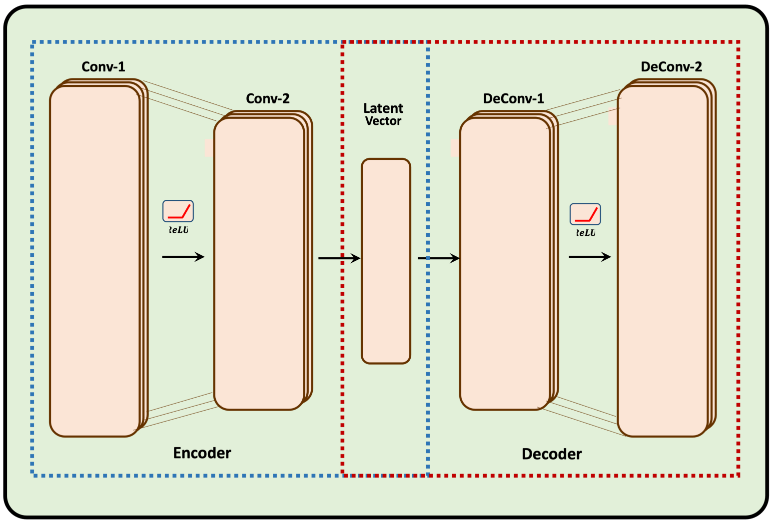


Figure 4.14: Architecture of Convolutional Autoencoder

4.2.4.3 Learning and anomaly detection

For what concerns learning, we consider the 4 cases described below and summarized in Table 4.4:

1. **Baseline case:** In this case, no transfer learning is applied. It will be used as the term of comparison of the approaches exploiting transfer learning;

2. **Transfer learning without model refinement:** In this case, we train the CAE model by using the data of a given bicyclist and we use the resulting model, as it is, i.e., without any further training, for other bicyclists. Note that according to such an approach it is unnecessary to collect the data for all users, which is extremely efficient;

3. **Transfer learning with model refinement:** In this case, we train the CAE model by using the data of a given bicyclist and we use the resulting model as the starting point for the training of new models for other bicyclists. Using such an approach, the amount of data needed regarding the new bicyclists should be lower and convergence should be faster when compared to the baseline case;

4. **Transfer learning with partial model refinement:** In this case, we train the CAE model by using the data of a given bicyclist and we use the resulting model as the starting point for the training of new models for other bicyclists. The difference when compared to the previous case is that only a subset of the model parameters will be retrained, i.e., for some of

Table 4.4: Summary of the learning approaches considered in our study.

Identifier	Name	Synthetic explanation
Case-1	Baseline	Transfer learning is not utilized
Case-2	Transfer learning without refinement	The model identified for a bicyclist is used as it is for other bicyclists
Case-3	Transfer learning with refinement	The model identified for a bicyclist is used as the starting point of the training of the models of the other bicyclists
Case-4	Transfer learning with partial refinement	The model identified for a bicyclist is used as the starting point of the training of a subset of the parameters of the models of the other bicyclists

the layers the parameter `trainable` will be set equal to `false`. In this way, the execution of the training on the new bicyclist is expected to be faster as the optimization will be executed on a subset of the model parameters only. The cost of such increased velocity is expected to be paid in terms of performance because at best the optimization algorithm can identify a *sub-optimum*.

Once the model has been trained it is used for anomaly detection at run-time. More specifically, the collected data is provided as input to the model and the reconstruction error, i.e., the difference between the output of the model, which should be a reconstruction of the input data, and the actual input is evaluated. An error higher than a given threshold is evidence that the current input is significantly different from the data that was used for training the model and therefore, there is an anomaly going on.

4.2.5 Experimental results

In this section, we show the experimental results obtained by applying the proposed approach with real data collected in the city of Catania in Italy. More specifically, in Section 4.2.5.1 we will present the experimental setup. Then, in Section 4.2.5.2, we will show results demonstrating the advantages of transfer learning in the model optimization process. Finally, in Section 4.2.5.3 we will validate the proposed anomaly detection approach as a whole.

4.2.5.1 Experimental Setup

Our experimental campaign exploits the GPS data collected by a bicycle equipped with a Video VBOX Lite (VVL) and ridden by 10 bicyclists. Each riding test was run on the same path of about 4 km length, composed of different types of road infrastructures: cycle track, bicycle/bus shared lane, roundabout, signalized intersections Finally, the dataset is a time series signal where each user includes about 9000 samples regarding location (longitude, latitude) speed, and heading of the bicycle path collected at a sampling rate of 10 Hz.

For each ride, we log the real positives and CNN positives as actual and detected anomalies respectively. Real positives are the occurrences of Traffic Conflicts (TC), identified during the in-field experiment while CNN positives are the anomalies detected by the model.

Results are evaluated for the different learning approaches described in Section 4.2.4 and summarized in Table 4.4. More specifically, in Section 4.2.5.2 we first compare the performance in terms of the reconstruction error. Then, the ability to detect anomalies in the resulting model is evaluated.

To this purpose, note that our problem is a typical binary classification problem. In fact, data is fed to our model that must give as output “Positive” if it detects an anomaly, and “negative” in the opposite case. Indeed binary classification, "Positive" and "Negative" refer to the predictions made by the classifier, while "True" and "False" indicate whether the prediction matches the actual observation or not.

The confusion matrix (CM) is a tool used to evaluate the performance of the classification model, and it helps to calculate the key performance indicators *Recall* and *Precision* as shown in Fig. 4.15. True Positive (TP) represents cases where the model correctly identified a real positive. On the other hand, False Positive (FP) indicates cases, where the model detected an anomaly that did not actually exist as a Real Positive, and False Negative (FN), means there was an actual "True" that the model failed to detect as an anomaly. Recall evaluates the ability of the model and checks True Positives (TP) over the total number of actual Positives (TP + FN). A higher Recall score indicates that the model is better at detecting real positives. Precision calculates the proportion of True Positives (TP) over the total predicted Positives (TP + FP). A high TP rate and a low FP rate lead to high Precision.

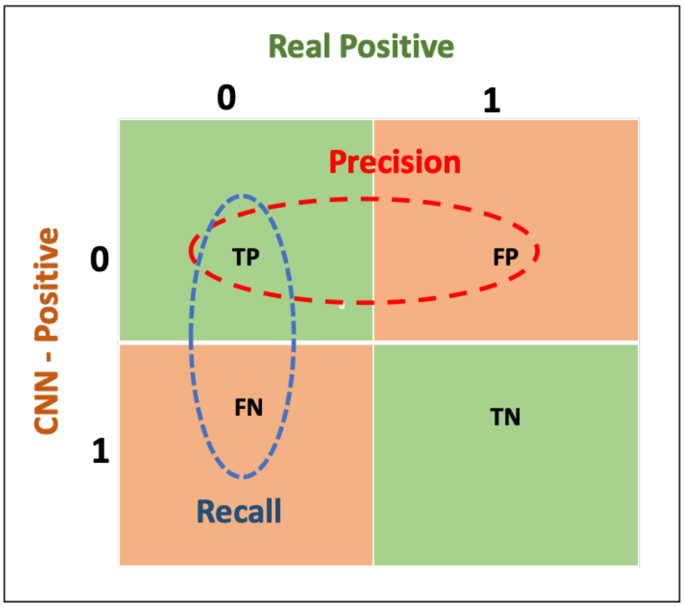


Figure 4.15: confusion matrix for the computation of recall and precision

4.2.5.2 Model Optimization

Model optimization is executed to minimize a *training loss*, which measures the difference between the reconstructed output and the original input. When training loss nearly approaches zero, it means the model is well trained. Therefore, plots representing the training loss are a way commonly used to assess the performance of model learning,

In this section, we compare the model optimization performance of the four learning approaches described in the previous Section 4.2.4. More specifically, in Figure 4.16, we report the training loss in the 4 different cases vs the percentage of the original dataset utilized to train the model in Case 1 and to refine the model after transfer learning in all other cases. Figure 4.16 contains 4 plots obtained using the settings summarized in Table 4.5.

Furthermore, note that plots 4.16(a), (b), and (c) have been obtained by setting the upper bound for the number of training epochs to 5, 10, and 15, respectively. Plot 4.16(d) has been obtained without setting the above upper bound. As obvious the loss decreases as the above upper bound increases and is minimal when it is not set at all. In each figure of the plots, we observe that the performance obtained in Cases 2-4 is similar, while performance is much worse in Case 1. Therefore, transfer learning is beneficial in the scenario of our interest.

Table 4.5: Training Model Parameters

S.No.	**Parameter**	**Value**
1	No. of Layers	3
2	Stride	4
3	Random Seed	120
4	Batch Size	128
5	No. of Epochs	5, 10, 15, 200 (maximum)
6	Learning Rate	0.001
7	Activation Function	Relu
8	Data size in percentage	1, 2.5, 5, 7.5, ...20

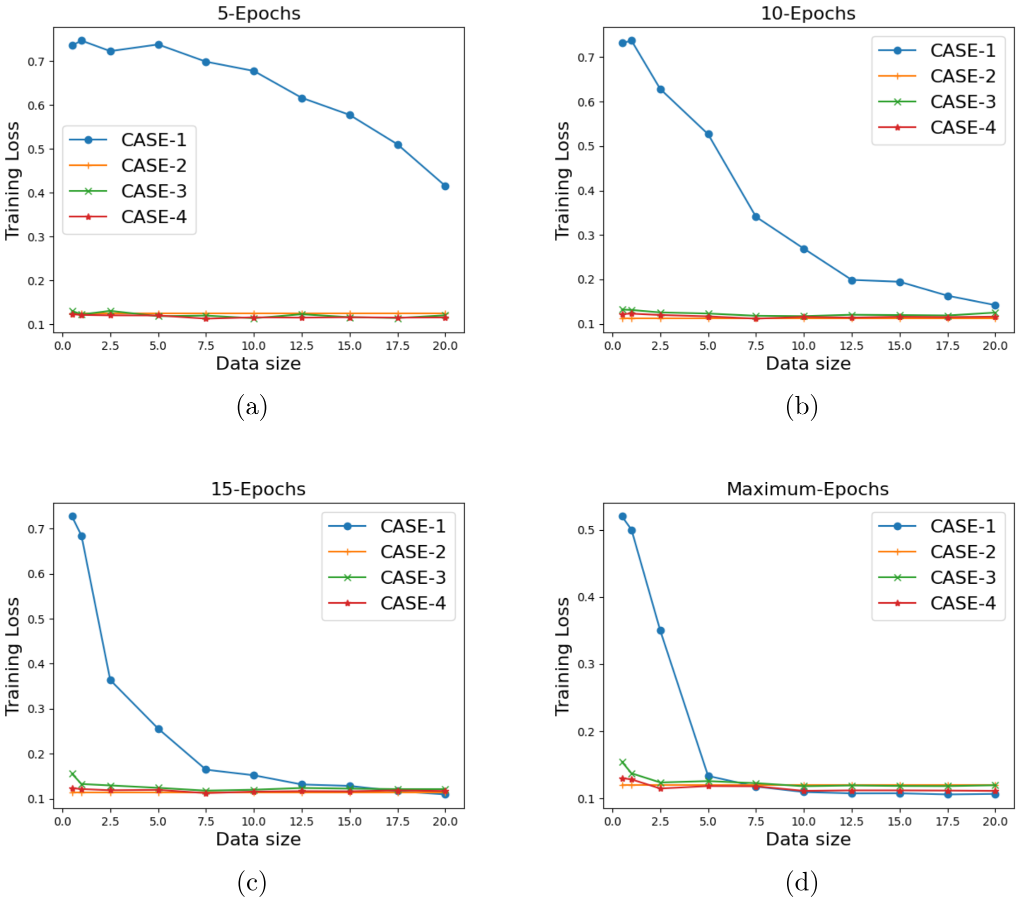


Figure 4.16: Training loss with respect to the number of epochs

In Figures 4.17 and 4.18, instead, we report the performance of the proposed anomaly detection schemes in terms of Recall and Precision, respectively.

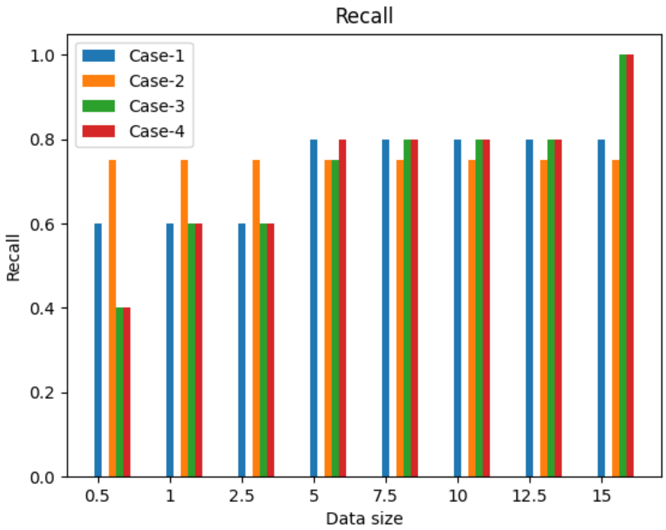


Figure 4.17: Recall

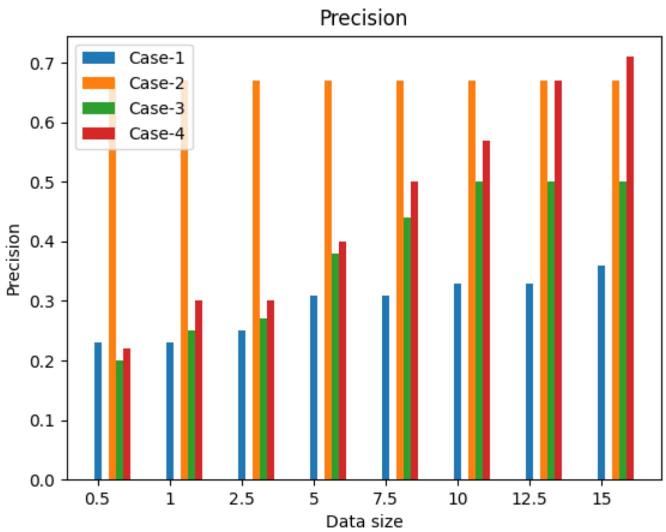


Figure 4.18: Precision

4.2.5.3 Validation

In this subsection, we are validating the results by implementing optimized models with the data set from various users. Moreover, the application of the proposed methodology is presented where we identified the dangerous point of the hotspot area of the city of Catania (Italy).

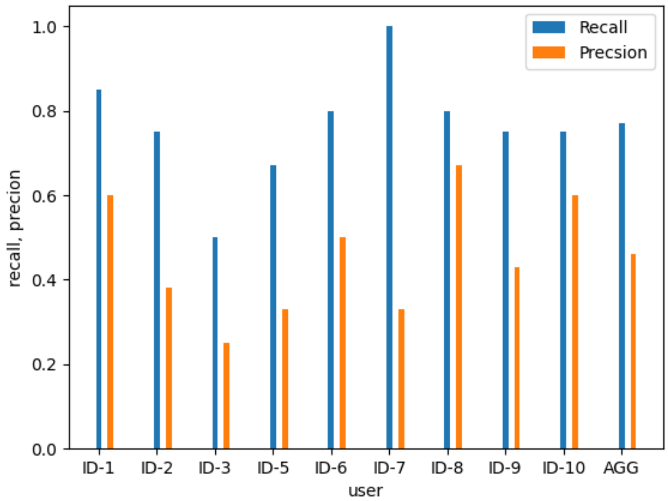


Figure 4.19: Recall and Precision for various users

4.2.5.4 Dataset from various users

Validation of the optimized model is performed by considering data from various bicycle users. The optimized model follows 15% of data and case 4. Recall and precision are calculated for various users as shown in Fig.4.19.

Upon examination of Figure 4.19, it is observed that Recall is consistently higher than Precision when the two are compared for the 10 users. It is worth mentioning that the lower classification performances are reported for users who experienced a very limited number of TCs (e.g. ID-3, ID-5) and ID-4 is not reported because for that user the number of TCs was equal to 0. In our application, it is important because False negatives have greater concern than False Positives. False Negative refers to missed TCs while False Positives are expected as anomalies in normal riding when braking at intersections or swerving to avoid pavement defects or fixed obstacles (e.g. illegal parking) and other issues not classified as TCs in our experiment. On average the total Recall was equal to 77% which can be considered an overall good performance for the proposed classifier.

4.2.6 Application of the proposed methodology

The procedure was applied to the trajectory data and anomaly detections located with the GPS coordinates in the different roadway sections composing a mixed cycling path selected for the experiment (i.e. cycle track, roundabout, cycle track termini, and bicycle/bus shared lane). The anomalies in cycling behavior detected by transfer learning methodology were compared with the TCs noticed by analyzing with the users the video recording during the experiment and further checked by road safety experts [92].

Figure 4.20: map

Where the TL anomalies overlapped with the Real Positives, a True Positive was identified (red dot in the map), while if the procedure did not identify the Real Positives, we had a False Negative (black dot in the map). False Positives are plotted as yellow dots. As an example, screenshots of the videos recorded with the speed profile in correspondence with three TP are shown. GIS mapping is useful to analyze the spatial distribution of TP (red dots) and FP (yellow dots). It is worth noting that there is clustering at the roundabout, along the bicycle/bus shared lane, particularly at the signalized intersections. As expected few anomalies are classified along the cycle track but in the correspondence two breaks in the lane separation and at the track termini where the path needs to cross the normal traffic lane.

4.2.7 Remarks

As the urban population grows and traffic becomes more congested, it is crucial for sustainable cities and healthy people to make cycling a desirable mode of transport. However, the perception of poor safety is one of the main hurdles to cycling and is still an open issue. Enhancing the safety of cycling by identifying high-risk locations and routes in the wide and complex urban road network is a major challenge for city managers in implementing safety measures. Crash data analysis can be used to assess risk, but the lack of data and its under-reporting make this normal approach less effective and also only reactive rather than proactive.

In this paper, we have presented an approach that exploits unsupervised machine learning for anomaly detection and transfer learning to reduce the amount of data needed to synthesize a fitting artificial neural network dramatically.

Results of our experimental campaign obtained in a specific case study demonstrate that the proposed approach has promising performance as it has 77 % of Recall on average while 100% for some individual users. Performance in terms of Precision is 46% which looks significantly lower than in more traditional classification models. However, it is northerly to mention that our False Positives (i.e. anomaly detection not correspondent to a pre-classified traffic conflict) are indicators of anomaly maneuvers along the user path even if not classified as traffic conflict because of the lack of an opponent road user. Such anomalies can be related to traffic regulations (e.g. sudden stopping at the red light) or other critical conditions (e.g. avoiding potholes or other pavement anomalies, maneuvers due to discontinuities in the cycling path) whose identification can be equally useful to analyze the cycling safety conditions in the road network [89]. Localization and spatial clustering of anomaly events can be used for network screening to select sites for more in-depth safety inspections and countermeasure selection[135]. Although our experiment was limited to only 10 users, plotting the CNN positives in GIS shows some clustering that is located where expected in the road network and already classified as high-risk locations in a previous study [92].

The study also gave the opportunity to identify some novel open issues for practical application. Given that evasive maneuvers can be a mix of speed and direction changes that take less than 2 seconds (0.8 - 1.5 sec. in our test), in our study we have collected data using extended GNSS NMEA contents (i.e., Location, Speed, Heading) acquired at 10 Hz frequency which are not commonly provided by standard smartphones and applications. Therefore, for practical applications, there is a need for more suitable data recording or for dedicated devices installed for example in the bike sharing fleets.

When the cyclist rides in different road infrastructures and traffic conditions she/he can adapt the cycling style to the actual and perceived risk conditions. In our study, in order to achieve a general cycling model of the user, the model was built by shuffling the training dataset collected along the different typologies of the test road path. Then some layers have been further customized to each user. Therefore, a future improvement can be a model that specializes in some of the layers of the user, whereas the others to the current road environment.

4.3 Spatial Analysis: Role of convolutional layers with respect to road environment and user

4.3.1 Objective

Machine Learning (ML) can be applied in many scenarios where one user interacts with the surrounding environment. In such cases, the environment will provide *stimuli* to the user that will react consequently with certain actions. The way in which the user will execute the above actions depends on the specific user and her current status.

In this paper, we focus on *Convolutional neural networks* (CNN)s and *Convolutional autoencoders* (CAE)s that are *deep* neural networks that have been successfully utilized in several application scenarios ranging from image recognition to anomaly detection and useful feature extraction [115] [136]. In both CNNs and CAEs, one or several *convolutional* layers are at the input of the neural network. Such convolutional layers exploit the properties of the convolution operation to detect patterns in the input data.

In such a context, the first major objective of this paper is to answer the following question:

Are the patterns recognizable by the convolutional layers more specific to the user or of the environment?

To understand the importance of the above question, observe that for a given *scenario*, defined here as the combination of a specific user and her current environment, a fitting model will be dependent on both the user and the environment.

For example, consider a smart road scenario. A traffic light along a given road implies that a driver, that is the user that can be identified in such a scenario, might need to stop the car in a certain area. Therefore, the traffic light will provide the stimulus. However, the way in which the driver will stop the car depends on the driver and her current (mental and health) conditions.

As a consequence, in principle, a different model must be trained for all possible pairs of user-environment composing the scenario.

This implies a way too large processing for the training of the neural network models when the number of users and environments increases[1]

[1]Consider that the conditions of the same environment might change over time. Therefore, a specific model might be needed for each user in each environment in each possible condition, which is even more impractical.

However, if an answer to the above fundamental question is given the processing can be significantly reduced. In fact,

- If the patterns in the input data are mostly user-dependent, then each user can have its convolutional layers which will be concatenated to other layers. Only the new layers of the resulting neural network will be trained, instead of the entire neural network (see Figure 4.23).

- If the patterns in the input data are mostly environment-dependent, then each scenario will have its own convolutional layers. When a new user enters the environment the environment-dependent convolutional layers are concatenated to other layers. Only such new layers of the resulting neural network will be trained (see Figure 4.24).

In both cases, the number of model parameters to be trained can be reduced dramatically.

In this paper, we have demonstrated that in a specific case considering bicyclists (the users) riding a few road segments (the environments), the input patterns are specific to the user rather than to the environment.

Starting from such a result we define a few techniques that help increase the efficiency in the use of network resources and the level of privacy in a smart road scenario.

Accordingly, the following of this paper is organized as follows. In Section 4.3.2 we provide some background regarding deep neural networks with a specific focus on Convolutional Neural Networks (CNN)s and Convolutional Autoencoders (CAe)s. In Section 4.3.5 we formulate the problem we want to tackle and sketch our methodology. In Section 4.3.6 we apply the above methodology to a specific use case. Finally, in Section 4.3.10 we draw our conclusions.

4.3.2 Background

In this section, we provide some background regarding convolutional neural networks and convolutional autoencoders respectively. More specifically, in Section 4.3.3 we will provide the basic concepts regarding convolutional neural networks, then, in Section 4.3.4 we will review the convolutional autoencoder basics.

4.3.3 Convolutional Neural Network

Convolutional Neural Networks (CNN) are a well renowned deep learning artificial neural networks used to process data patterns [115][137][138]. Applications of

CNN are image processing, anomaly detection, and feature extraction [137] [136].

CNNs' architecture consists of three basic layers named input, hidden, and output layers [115], [136], [139]. Typically, hidden layers include convolutional, pooling, and connected layers. *Convolutional layers* are responsible to extract features and recognize patterns efficiently. *Pooling layers* are used to minimize data dimensions by integrating the output of the previous layer into a single neuron of the next layer. Finally, the job of a connected layer is to connect each neuron of one layer to another. Hence, CNN is a layered network made of neurons [137], [139], [140].

In CNN, each neuron is responsible for a task t with input X and output Y. The task t is expressed as given in eq. (4.17):

$$Y^{(k)} = t(X^{(k-1)} * V^{(k-1)} + c^{(k-1)}) \tag{4.17}$$

where V and c are the weight vector and bias of the neuron, respectively. Weights V represent the connection strength of neurons and improve the effectiveness of the activation function. Basically, weight parameters decide how activation will be triggered. The bias c of the neuron is a constant used to introduce the delay in the activation function. In eq. (4.17) function $t(\cdot)$ is considered an activation function, which is responsible for transforming weighted input into an output. Several activation functions have been considered in the literature; examples include *Rectified Linear Unit* (ReLU), sigmoid, and hyperbolic tangent functions.

Neural network training sets the parameters (i.e weights, biases) to build relationships between inputs and outputs. The training of neural networks is based on the backpropagation algorithm and exploits an appropriate loss function. Mostly, the loss function on the data item (X^k, P^k) in which X^k represents a specific input and P^k the corresponding expected output can be evaluated as shown in eq. (4.18).

$$L(V, c, X^k, P^k) = \frac{1}{2} \parallel l_{v,c}X^k - P^k \parallel^2 \tag{4.18}$$

where $l_{v,c}X^k$ is the result given by the neural network when the input is X^k, while P^k is the expected outcome. Let T represent the number of data packets, the overall loss function can be calculated as in eq. (4.19):

$$L(V, c) = \frac{1}{k} \sum_{k=1}^{T} L(V, c, X^k, P^k) \tag{4.19}$$

4.3.4 Convolutional Autoencoder

Convolutional autoencoders (CAE) represent variants of Autoencoder (AE)s [141], [142]. CAEs are capable of learning the most useful feature patterns in the input data [143]. Applications of CAEs as well as AEs include dimension reduction, information retrieval, and anomaly detection [81]. A CAE consists of two components: *Encoder* and the *Decoder*.

The Encoder, specified by a set of parameters, U_E, is responsible for feature extraction by mapping the input X into latent vector T as shown in eq. (4.20).

$$T = f_{en}(U_E, X) \tag{4.20}$$

The Decoder is specified by the parameters U_D which takes the latent vector T as input and gives output Y as shown in eq. (4.21). The AE training minimizes the difference between the input X and the output Y as shown in eq. (4.22).

$$Y = f_{Dc}(U_D, T) = f_{Dc}(U_D, f_{en}(U_E, X)) \tag{4.21}$$

$$\cong X \implies f_{Dc}(U_D, f_{En}(U_E, X)) \cong X \tag{4.22}$$

Therefore, given a dataset Ω, the training phase consists of outcomes that are extracted from encoder and decoder parameters i-e U_E and U_D that minimize the loss function as defined in eq. (4.23)

$$L_\Omega(U_E, U_D) = \Sigma_{X\in\Omega} \parallel f_{Dc}(U_D, f_{En}(U_E, X)), X \parallel \tag{4.23}$$

There are several ways to measure the loss such as the Mean Absolute Error (MAE), the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE). We opt MAE over Mean Square Error and Root Mean Square Error (RMSE) due to data distribution and error size suitability. Therefore MAE is expressed as shown in eq. (4.24).

$$MAE = \parallel Y, X \parallel = \frac{1}{k}\sum_{k=1}^{k} |P^k - X^k| \tag{4.24}$$

On the basis of encoder and decoder parameters $(\widetilde{U_E}, \widetilde{U_D})$, the autoencoder is evaluated as shown in eq. (4.25).

$$(\widetilde{U_E}, \widetilde{U_D}) = \underset{(U_E,U_D)}{\mathrm{ArgMin}} \sum_{X\in\Omega} \| f_{Dc}(U_D, f_{En}(U_E, X), X) \| \tag{4.25}$$

Generally, the significant symbols and their definitions in this section are summarized in Table 4.6.

Table 4.6: Important notations and their definition

S.No.	Symbol	Definition
1	$t(X^{(n-1)})$	task of neurons with input X
1	$V^{(n-1)} and c^{(n-1)}$	vector of weights and bias values
2	$L(V, c, X^k, P^k)$	loss function for unique training data packet
3	Ψ	learning rate
4	U_E and U_D	encoder and decoder
5	T	latent vector
6	$\frac{1}{n}\sum_{i=1}^{n} \lvert X_i' - X_i \rvert$	mean absolute error
7	$M_{i,j} = [O_{i,j}, I_{i,j}]$	model with input and output layers
8	$S_{i,j}$	scenario
9	$D_{i,j}$	Dataset
10	U_i	*user* with specific layers
11	E_j	environment with specific layers

4.3.5 Problem formulation and methodology

Let us consider a scenario in which users interact with the environment. More specifically, we focus on the i-th user interacting with the j-th environment and we refer to the resulting scenario and the dataset collected in that specific scenario as $S_{i,j}$ and $D_{i,j}$, respectively. Let us call $M_{i,j}$ the Convolutional Autoencoder (CAE) utilized for such a scenario. The structure of such a CAE is depicted in Figure 4.21. In the above Figure 4.21, we distinguish *outer* and *inner* layers. Therefore, the model $M_{i,j}$ will be the concatenation of outer and inner layers which we denote as

$$M_{i,j} = [O_{i,j}, I_{i,j}] \tag{4.26}$$

where $O_{i,j}$ and $I_{i,j}$ represent the outer and inner layers respectively for the scenario $S_{i,j}$.

In CAEs, the outer layers are utilized to recognize and then reconstruct the patterns in data, whereas the inner layers are utilized to interpret such patterns into a bigger application-dependent picture.

As we already stated, the objective of our study is to analyze whether the patterns recognized by the outer layers are more specific to the user or to the environment. Responding to such a question is extremely important. In fact, We examine two scenarios in the following manner, and the information is also summarized in Table 4.7.

- Case 1) If the patterns are more specific of the user, we can assume that user i has its own outer layers, U_i, which are not trained further when the user visits a new environment, say the j-th, and the model $M_{i,j}$ must be synthesized. Such user-specific layers, U_i can be carried by the user in her smartphone or stored in some reserved area in the cloud. In this case the layer parameters, U_i, will be *transferred* to the model, i.e.,

$$M_{i,j} = [U_i, I_{i,j}] \tag{4.27}$$

- Case 2) If the patterns are, instead, more specific of the environment, we can assume that such environment, say the j-th, has its own outer layers E_j which are not trained further when a new user, say the i-th visits it and thus, a new scenario must be considered. In this case, the outer layers E_j will be stored in some server and will be transferred to the scenario model, $M_{i,j}$, that is,

$$M_{i,j} = [E_j, I_{i,j}] \tag{4.28}$$

Note that in both cases, the outer layers will not be further trained, i.e., the `trainable` parameter is set to `false`, and thus the number of parameters to be evaluated and the amount of data needed for the training, are significantly reduced.

In order to determine which of the two cases applies, we will compare the fitting loss obtained by synthesizing the scenario model as given in eqs. (4.27) and (4.28).

To this purpose we select a target scenario S_{i^*,j^*} in which user i^* interacts with environment j^* and data D_{i^*,j^*} is collected. Then we consider another user $i \neq i^*$ and another scenario $j \neq j^*$ and we train models $M_{i^*,j}$ and M_{i,j^*} using data $D_{i^*,j}$ and D_{i,j^*}, respectively. According to the notation given in eq. (4.26), we can write

$$M_{i^*,j} = [O_{i^*,j}, I_{i^*,j}] \text{ and } M_{i,j^*} = [O_{i,j^*}, I_{i,j^*}] \tag{4.29}$$

Then we train the inner layers, $I^{(1)}_{i^*,j^*}$ and $I^{(2)}_{i^*,j^*}$, of the models $[O_{i^*,j}, I^{(1)}_{i^*,j^*}]$ and

$[O_{i,j^*}, I^{(2)}_{i^*,j^*}]$, respectively exploiting part of the dataset D_{i^*,j^*} and we calculate the corresponding fitting loss values using the rest of the dataset D_{i^*,j^*}.

Let $L(O, I, D)$ represent the fitting loss of the CAE when the model is constructed as the concatenation of the outer layers O and the inner layers I and the dataset utilized is D. Observe that if the fitting loss $L(O_{i^*,j}, I^{(1)}_{i^*,j^*}, D^{i^*,j^*})$ achieves lower values more quickly than the fitting loss $L(O_{i,j^*}, I^{(1)}_{i^*,j^*}, D^{i^*,j^*})$ we are in Case 1. In fact, low loss values and thus good fitting can be achieved by exploiting the outer layers, $O_{i^*,j}$ obtained for the same user, i^*, in another environment, j.

Analogously, we are in Case 2 if the fitting loss $L(O_{i,j^*}, I^{(1)}_{i^*,j^*}, D^{i^*,j^*})$ achieves lower values more quickly than the fitting loss $L(O_{i^*,j}, I^{(1)}_{i^*,j^*}, D^{i^*,j^*})$.

It follows that in order to determine which between Case 1 and Case 2 apply, we will compare the fitting losses $L(O_{i^*,j}, I^{(1)}_{i^*,j^*}, D_{i^*,j^*})$ and $L(O_{i,j^*}, I^{(1)}_{i^*,j^*}, D_{i^*,j^*})$.

Observe that by identifying whether we are in Case 1 or Case 2, it is possible to significantly reduce the amount of data exchanged in the communication network for training purposes. In particular, the parameters of our model are represented in Table 4.8.

For example, later in this paper, we will show that in a specific setting, i.e., bicyclists riding through a few streets of Catania in Italy, we are in the first case, that is, the outer layers are specific to the user, and thus $M_{i^*,j^*} = [U_{i^*}, I^{(1)}_{i^*,j^*}]$ as given in eq. (4.27). In this case, during the inference phase the data which is generated locally, let us call it X_{i^*,j^*}, can be given as input to the outer encoder layers $U^{(E)}_{i^*}$ which can be executed on some user devices. Let us call Z_{i^*,j^*} the resulting output. This is given as input to the inner layers running in some remote server. Let Z'_{i^*,j^*} represent the output of the inner layers. This is sent back to the user where it is given as input to the outer decoder layers $U^{(D)}_{i^*}$ to obtain the reconstruction X'_{i^*,j^*} of the initial data, X_{i^*,j^*}.

Observe that in this case we obtain the following advantages:

- The data collected regarding the user i^* is kept local, which improves privacy.
- Since the dimensions of the data samples Z_{i^*,j^*} and Z'_{i^*,j^*} are significantly smaller than those of the input data X_{i^*,j^*} the amount of transmitted data is reduced and thus the need for communications resources is reduced as well.

Table 4.7: Learning strategies considered in our analysis

Use case	Learning strategy	More details
Case 1: Data patterns assumed to be dependent on user	Transfer learning with the refinement of all layers	Each user has her own user-dependent outer layers. When she visits a new environment the entire neural network will be retrained, using the user-specific outer layers as the starting point
	Transfer learning with a refinement of the inner layers only	Each user has her own user-dependent outer layers that will not be trained further. When she visits a new environment the inner layers will be retrained only.
Case 2: Data patterns assumed to be dependent on environment	Transfer learning with the refinement of all layers	Each environment has its own environment-dependent outer layers. When a new user visits the environment the entire neural network will be retrained, using the environment-specific outer layers as the starting point
	Transfer learning with refinement of inner layers only	Each environment has its own environment-dependent outer layers that will not be trained further. When a new user visits the environment the inner layers will be retrained only.

Table 4.8: Model parameters

S.No.	Parameter	Value
1	Model Input	User (Bicyclist), Road-Environment (Bike-lane, Normal-lane, RoundAbout, Shared-lane)
1	No. of Layers	4
2	Stride	4
3	Random Seed	120
4	Batch Size	128
5	No. of Epochs	200 (maximum)
6	Filters	16
7	Kernel size	8
8	Learning Rate	0.001
9	Activation Function	Relu

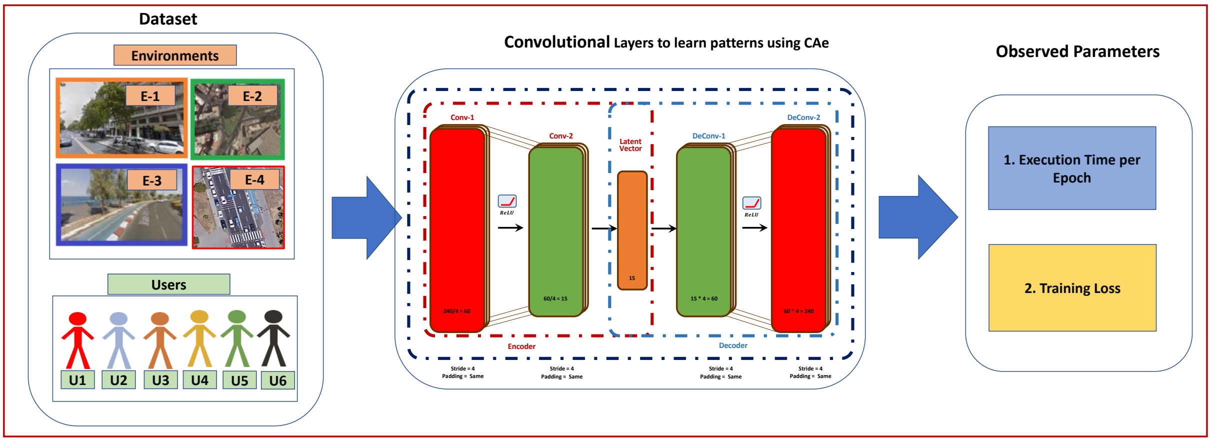


Figure 4.21: Flow diagram of proposed approach

4.3.6 Use case

In this section, we focus on a specific use case. More specifically, in Section 4.3.7 we present the scenario and the dataset. In Section 4.3.8 we give an overview of the experiments carried out and explain their objectives. Finally, in Section 4.3.9 we present the numerical results.

4.3.7 Scenario and dataset

In this study, we focus on collecting data in an actual, real-world setting. Specifically, we gather data from specific roads located in Catania, Italy.

The data is obtained from a specialized bicycle equipped with a GPS (Global Positioning System) and an HD video system called Video Vbox Lite (VVL). The VVL captures a comprehensive GPS NMEA dataset, which goes beyond just latitude and longitude by including speed and heading information at a frequency of 10 HZ. This dataset is synchronized with a video recording from two HD cameras. The data collected demonstrated excellent accuracy and resolution. It is important to highlight that although GPS alone may have limitations in accurately determining positions, the speed and heading data derived from the Doppler method and Carrier Phase observations exhibited high-quality information [109][144].

Data was obtained from a group of six cyclists, identified as ID-1 to ID-6, who took part in controlled test rides. Among the participants, there were five males and one female. Their ages ranged from 27 to 65, with 40 percent of them being over 40 years old. The participants were instructed to ride the instrumented bicycle according to their normal behavior. The tests took place under typical weather conditions, during daylight hours, and regular traffic flow. The route covered less than 4 km, encompassing various road infrastructures to represent different traffic and road environments, including bike lane, normal lane, roundabout, and shared lane. By traversing diverse road infrastructures, the travel experience becomes more dynamic as it allows for the assessment of different road sections and traffic conditions [88] [145]. For each rider, the collected dataset consisted of around 9,000 samples recorded at a frequency of 10 Hz. Despite the relatively small number of participants, this dataset is deemed suitable for training purposes.

4.3.8 Overview of the experiments

We consider 4 different road environments E_1, E_2, E_3, and E_4 and six users, U_1, U_2, U_3, U_4, U_5 and U_6. For what concerns the environments, as shown in Figure 4.22,

- E_1: It is a lane reserved for bicycles, its length is approximately 2300 m;
- E_2: It is a normal lane, and its length is approximately 130 m;
- E_3: It is a roundabout, its length is approximately 180 m;

- E_4: It is a shared lane, and its length is approximately 1000 m.

Figure 4.22: Four different road environments

It is obvious that the four environments have characteristics and lengths totally different from each other and the riding style will change accordingly.

Since we can distinguish 6 users and 4 environments, 24 scenarios can be defined, i.e., $S_{1,1}$, $S_{1,2}$, ..., $S_{1,4}$, $S_{2,1}$, ..., $S_{6,4}$.

We will compare the loss functions as explained in Section 4.3.5 to identify whether the conditions of Case 1 or Case 2 hold. Observe, however, that such a comparison might be influenced by the impact of the specificity of the user or of the environment. Accordingly, we will also consider the cases in which we start from the *wrong* user and/or environment. For example, we will evaluate the losses of the type $L(O_{i,j}, I_{i^*,j^*}, D_{i^*,j^*})$ which implies that the outer layers are created on user i and environment j and left unchanged, whereas the inner layers are retrained in a different scenario, S_{i^*,j^*}, and tested with the corresponding dataset D_{i^*,j^*}.

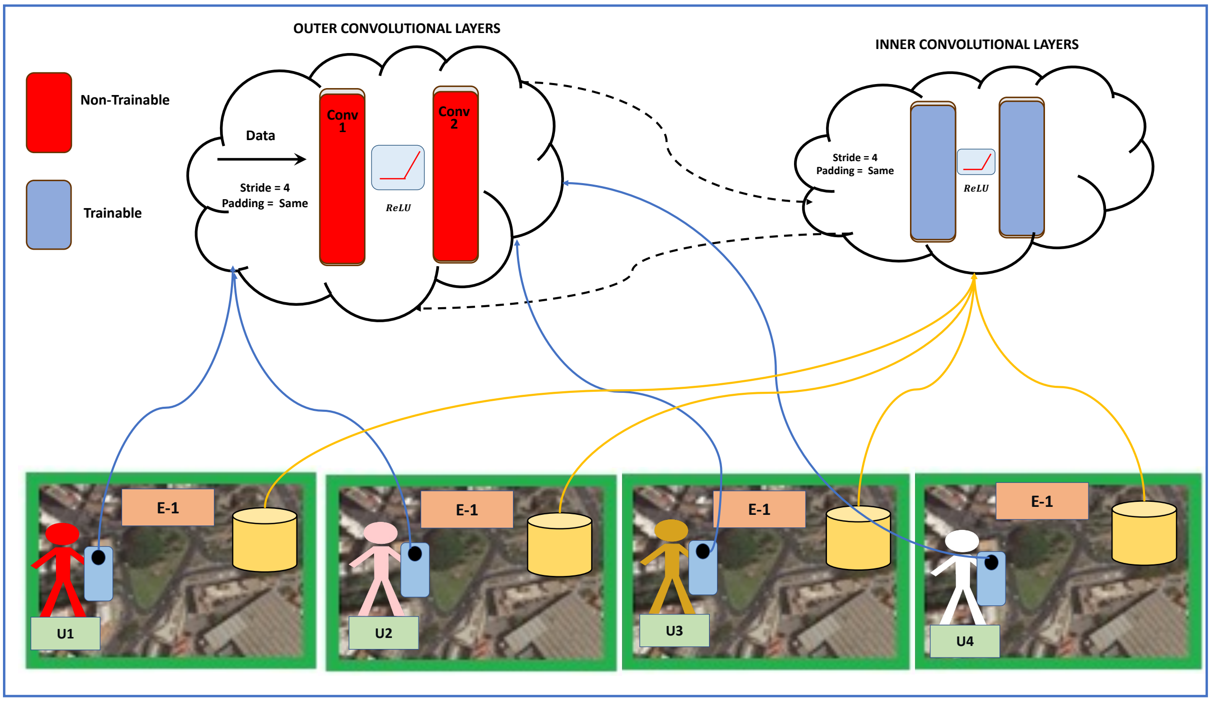


Figure 4.23: Behavior of convolutional layers with respect to user

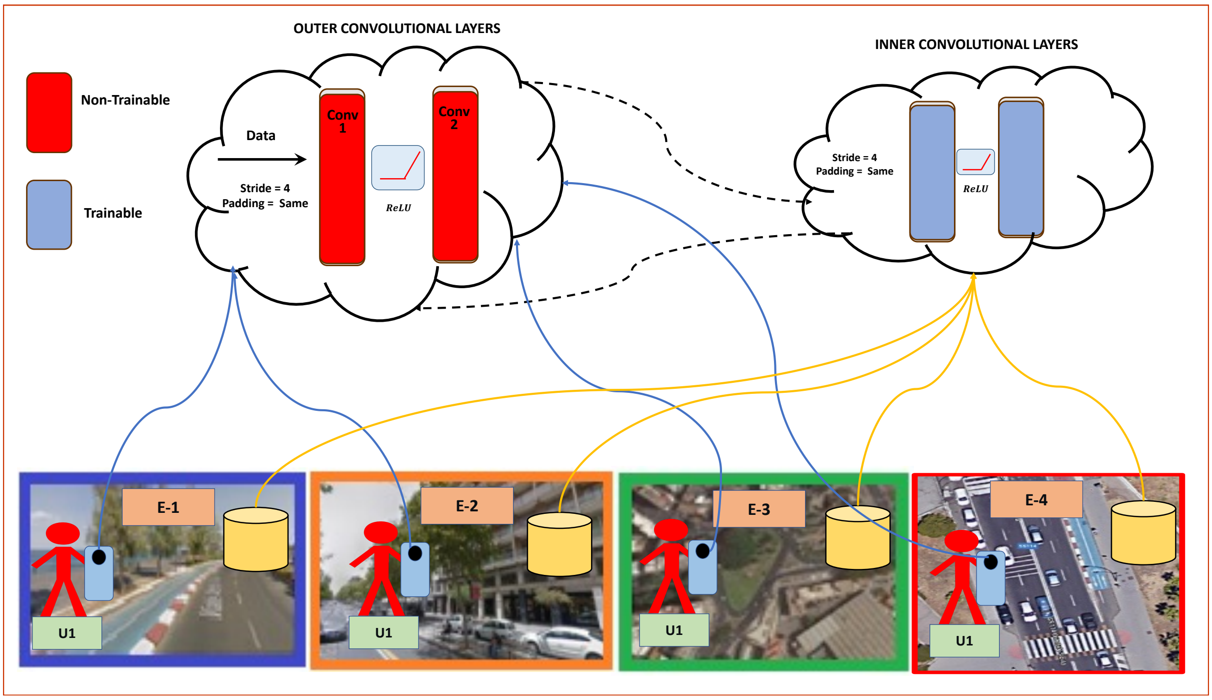
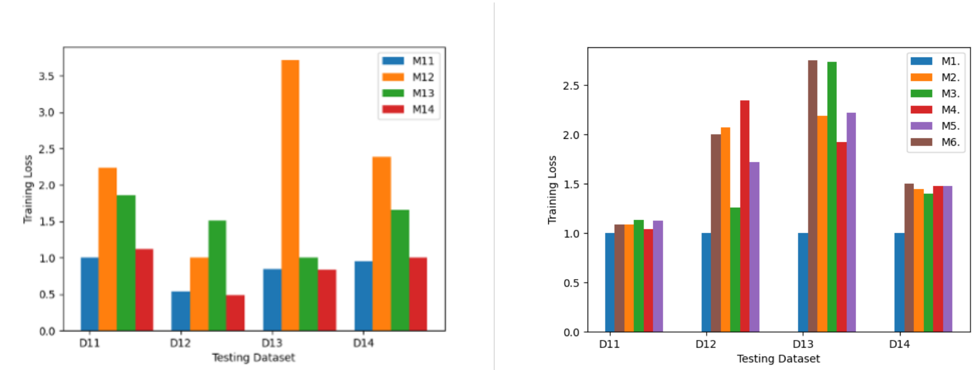


Figure 4.24: Behavior of convolutional layers with respect to Environment

4.3.9 Results

In this section, we present the experimental results obtained as explained in the previous section.

We start by observing that, as discussed in the previous Section 4.3.8, the amount of data available for environments E_1 and E_4 is much larger than for environments E_2 and E_3. Accordingly, the results corresponding to E_2 and E_3, which we report for the sake of completeness, cannot be considered reliable enough to draw any type of consideration. Therefore, in the following, we will focus on the results regarding the environments E_1 and E_4.



(a) Case 1 (b) Case 2

Figure 4.25: Training loss in Cases 1 and 2 with a refinement of the inner layers only.

In Figures 4.25 we show the validation loss obtained in Cases 1 and 2 in case only the inner layers are retrained. More specifically, in Figure 4.25(a) we show the validation loss obtained in Case 1. Results show what happens when a model for user U_1 is trained using the dataset collected in a specific environment, say the j-th, and its inner layers are retained using the data collected in a different environment, say the k-th.

Therefore, there are four different initial models $M11$, $M12$, $M13$, and $M14$. The inner layers of such models will be re-trained in a different environment using the corresponding collected data, i.e., $D11$, $D12$, $D13$, and $D14$. In Figure 4.25(a), we show four sets of bars, each for a different dataset utilized for the retraining of the inner layers. In each set there are four colored bars, each bar corresponds to a specific model which was used as the starting point to train the inner layers.

By comparing the blue and red bars, i.e., the first and the fourth corresponding to environments E_1 and E_4, for the dataset $D11$, we observe that the average loss obtained by using $M14$ as the starting point (red bar) is slightly higher, the difference is less than 12%, than the average loss obtained when $M11$ is used as the starting point (blue bar). This means that blocking the outer layers does not have a significant negative impact on the performance of the model as long as the user considered when training the starting model (in our case, user U_1) does not change.

By comparing the blue and red bars for the dataset $D14$, we observe that the average loss obtained by using $M11$ as the starting point (red bar) is even lower than the average loss obtained when $M14$ is used as the starting point (blue bar), the difference is 5% approximately. This means that blocking the outer layers does not have a negative impact on the performance of the model as long as the user considered when training the starting model (in our case, user U_1) does not change. Actually, in this case, we have an improvement in performance when using $M11$ as the starting point. This improvement is due to the fact that by training the neural network model with $D11$ first and $D14$ later, the total dataset utilized is larger and therefore, more fitting.

In Figure 4.25(b) we show the validation loss obtained in Case 2. More specifically, results show what happens when a model for any user, say U_i, is initially trained using the dataset collected in a specific environment, say the j-th, and its inner layers are retrained using the data collected for the 1-st user, U_1, in the same environment. Therefore, inner layers will be retrained using the datasets $D11$, $D12$, $D13$, and $D14$.

More specifically, in the plot we show several values of the average validation losses represented as colored bars. Such bars are grouped in 4 sets (one for each dataset utilized for retraining), consisting of six elements, one for each user considered when training the outer layers. For example, the green bar in the first group represents the average validation loss obtained in the case in which the initial model is trained considering the data collected for user U_3 in environment E_1, and the inner layers are then retrained using the dataset $D11$. Analogously, the orange bar in the fourth group represents the average validation loss obtained in the case in which the initial model is trained considering the data collected for user U_2 in environment E_4, and the inner layers are retrained using the dataset $D14$.

By comparing the bars in the first group, we observe that the average loss obtained exploiting the outer layers trained considering a user different from user U_1, i.e., all bars with the exclusion of the blue one, are slightly higher than the average loss obtained applying the outer layers trained considering user U_1, on the average the difference is around 10%.

This means that blocking the outer layers does not have a significant negative impact on the performance of the model as long as the environment considered when training the starting model (in our case, user E_1) does not change.

By comparing the bars in the fourth group, we observe that the average loss obtained applying the outer layers trained considering a user different from user U_1, i.e., all bars with the exclusion of the blue one, are significantly higher than the average loss obtained applying the outer layers trained considering user U_1, on the average the difference is higher than 45%.

This means that, in this case, blocking the outer layers has a significant negative impact on the performance of the model even if the environment considered when training the starting model (in our case, user E_4) does not change. Conversely, recall that in Case 1 there was even an improvement in performance in the same environment.

In summary, while in Case 1 we obtain an increase in the reconstruction loss of 3.5% on average, in Case 2 we obtain an increase in the reconstruction error which is approximately 28% on average.

Therefore, from the analysis of the results we can conclude that the data patterns recognized by the outer convolutional layers are more specific to the user rather than of the environment, that is the considered scenario falls mostly in Case 1.

4.3.10 Remarks

In this paper, we have defined a methodology to answer the question of whether the patterns recognizable by the convolutional layers of CNNs and CAEs are more specific to the user or to the environment.

The importance of resolving the above issues has been discussed.

We have applied the approach in a specific case involving several bicyclists running the streets in the city of Catania in Italy. The results have been discussed and techniques have been proposed to exploit the results of our investigation from a practical point of view.

Chapter 5

Use Case 2: Car

5.1 Abnormal driver behavior detection through deep learning

5.1.1 Orientation

Driver behavior is a major concern for societies and safe driving plays a crucial role in ensuring road safety [21][146]. Road accidents cause significant human and material losses every year, affecting both developed and developing countries. Therefore, it is crucial to monitor abnormal driver behavior to prevent potential accidents and minimize losses. The aim of this research is to use machine learning techniques to monitor and detect abnormal driving behaviors, such as drowsy, aggressive, and distracted driving, which are the leading causes of traffic accidents and deaths [2]. In our case, the abnormal driving behavior is referred to as an "anomaly" and several machine learning techniques have been studied and explored to identify these anomalies both considering vehicular and medical parameters.

5.1.2 Machine Learning for Detecting Vehicle Anomalies

Anomaly detection is the process of identifying rare events or patterns that deviate significantly from expected or normal behavior. Machine learning techniques have proven very effective in detecting anomalies in various industries such as finance, transportation, healthcare, and cybersecurity [140]. Machine learning algorithms can analyze large volumes of data and identify complex patterns that would be difficult or impossible for humans to detect[66]. Machine learning for anomaly detection has become increasingly important in today's data-driven world, as it enables organizations to proactively identify and address potential problems before they become serious problems. Machine learning offers a range of powerful methods for anomaly detection across domains [81]. Some of the most widely used machine learning methods for anomaly detection include:

1. **Unsupervised learning methods:** These methods do not require labeled data and are used to identify anomalies based on deviations from expected patterns. Clustering methods, such as k-means, are unsupervised learning methods commonly used for anomaly detection.

2. **Supervised learning methods:** These methods use labeled data to train a model to classify data as normal or abnormal. Some common supervised learning methods for anomaly detection include decision trees, supporting vector machines, and neural networks.

3. **Semi-supervised learning methods:** These methods use a combination of labeled and unlabeled data to train a model. Semi-supervised learning methods are useful when labeled data is scarce and the focus is on detecting anomalies in a specific subset of the data.

4. **Deep learning methods:** Deep learning methods, such as autoencoders and neural networks, are powerful techniques for anomaly detection. These methods can learn complex patterns and identify anomalies in large datasets.

In summary, machine learning offers a range of methods for anomaly detection. In our project, we used several machine learning techniques for anomaly detection, including Principal Component Analysis (PCA), One-Class Support Vector Machine (OSVM), and Convolutional Neural Networks (CNN). After evaluating the performance of each method, we determined that our most effective approach was the use of a convolutional autoencoder. This method uses neural networks to encode and decode input data, allowing anomalies to be identified based on deviations from expected patterns. By training the autoencoder to minimize the reconstruction error between input and output data, we can mark anomalies that result in a higher-than-expected reconstruction error. Overall, the convolutional autoencoder is a powerful technique for detecting anomalies in large datasets that can learn complex patterns and identify deviations from the norm. ML or DL-based models can be applied to the collected dataset which can be in the form of images or signals.

5.1.3 Dataset

5.1.3.1 Dataset Aquisition

The data source is a car instrumented with various types of sensors such as GPS (Global Positioning System), IMU, OBD, and ADAS as shown in Figure. 5.1 and Table. 5.1. Data is collected at 20Hz. More specifically, the dataset consists of the following specifications.

Dataset specifications:

- Type: Real-world (time series signals)
- Sensors: IMU, Mobileye, RTK-GNSS, OBD
- Location: Catania-Ispica, Ispica - Catania
- Duration: 5 hours
- Driver: Various
- Standard features: speed, direction
- Derived Parameters: acceleration, Heading rate
- Anomaly labeling: abnormal driving behavior

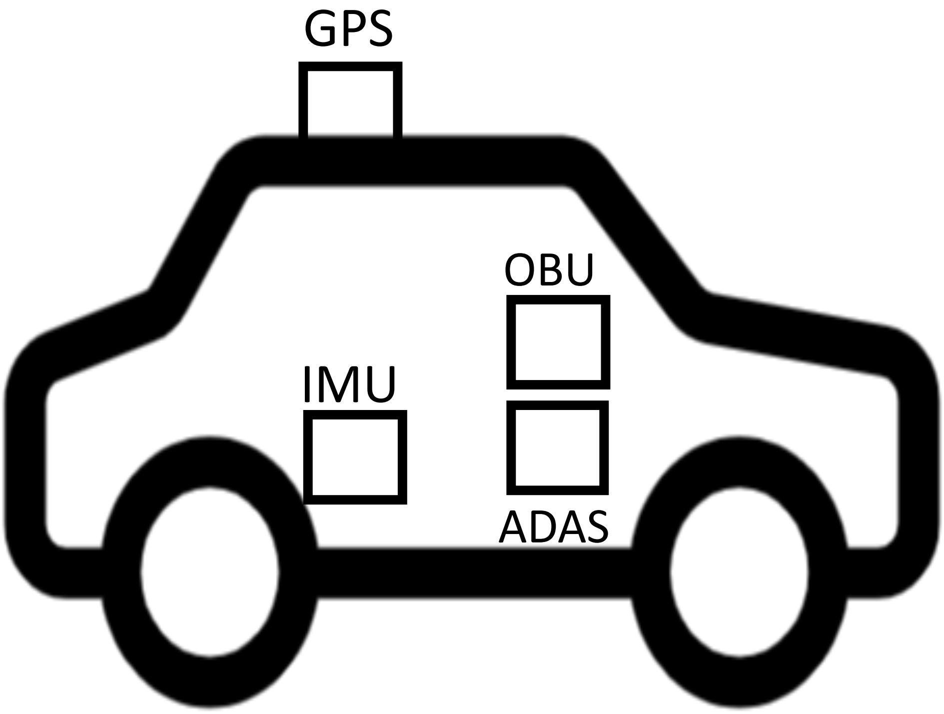


Figure 5.1: Instrumented Car with sensors

Table 5.1: Original collected data parameters from various sensors

hline No.	OBD	(GPS)	map	Mobile eye
1	Timestamp	GPS_Week()	GPS_Week()	timestamp
2	rpm	GPS_TimeWeek (s)	GPS_TimeOfWeek (s)	SoundType
3	speed	ins_status()	Timestamp	timeIndicator
4	coolant	ins position_type()	latitude(deg)	zeroSpeed
5	load	latitude(deg)	longitude(deg)	Headway Measurement
6	intake	longitude(deg)	north_velocity(m/s)	failSafe
7	maf	height(m)	east_velocity(m/s)	Maintenance
8	dtc	north_velocity (m/s)	up_velocity (m/s)	FCWon
9	throttle	east_velocity (m/s)	shape_distance	RightLDWON

Table 5.1: Original collected data parameters from various sensors

hline No.	OBD	(GPS)	map	Mobile eye
10	connected	up_velocity(m/s)	shape_radius	LeftLDWON
11	roll(deg)	shape_id	LDWOFF	wipersAvailable
12	pitch(deg)	shape_length	TSRenabled	speed
13	heading(deg)	osm_id	TamperAlert	left signal
14	latitude_std(m)	osm_highway	PedsinDZ	right signal
15	longitude std(m)	osm_int_ref	PedsFCW	brakes
16	height_std(m)	osm_ref	TSRWarning Level	
17	north_velocity std (m/s)	osm_name	HeadwayWarning Level	
18	east_velocity std (m/s)	osm_lanes		
19	up_velocity std (m/s)	osm_lit		
20	roll_std(deg)	osm_maxspeed		
21	pitch_std(deg)	osm_maxheight		
22	heading_std (deg)	osm_overtaking		
23	Timestamp	osm_incline		
24		osm_oneway		
25		osm_surface		
26		osm_tunnel		
27		osm_tunnel name		
28		osm_bridge		
29		osm_bridge name		
30		next_shape radius		
31		next_shape distance		
32		next_shape_id		

Table 5.1: Original collected data parameters from various sensors

hline No.	OBD	(GPS)	map	Mobile eye
33		next_shape length		
34		next_osm bridge distance		
35		next_osm tunnel distance		
36		next_osm intersection type		
37		next_osm intersection distance		
38		next_osm intersection id		

5.1.3.2 Data-Preprocessing

Once the data was recorded, several Python routines were applied for:

- Interpolation to improve data.
- Feature extraction.

5.1.3.2.1 Data interpolation Time series data interpolation refers to the process of estimating missing or incomplete data point values in a time series based on existing data point values [6]. This is often necessary in situations where some of the data points are missing due to various reasons, e.g. measurement errors, data loss during transmission, or incomplete data storage or acquisition [147][148]. Time series data interpolation is a common pre-processing step in many applications such as finance, weather forecasting, and signal processing, where comprehensive time series data is needed for accurate analysis and forecasting. We use this for time series signal data. We used the following interpolation techniques for our data:

a. Linear interpolation

A common technique for interpolating time series data is linear interpolation. In linear interpolation, missing data points are estimated based on the straight line connecting the two closest data points on either side of the missing data point

[148]. This method assumes that the data points between two known points follow a linear trend.

b. Nearest-Neighbour interpolation

"Nearest-neighbour interpolation" is a method of interpolating missing data points in a time series by replacing them with the nearest available data point value [149]. This method is also known as the nearest neighbor interpolation. In this method, the missing data point is replaced with the value of the available data point closest in time. This method assumes that the time series has a constant value between two consecutive data points and is useful when the time series has a relatively low level of variability. It is also computationally efficient and easy to implement. It can be used in signal processing and other applications where data is sampled at discrete time intervals.

5.1.3.3 Feature Derivation

Feature Derivation refers to the process of extracting new features from existing data by performing mathematical operations or transformations on raw data. In other words, deriving features involves creating new variables or features that capture important features or patterns in the data [150].

Feature derivation is a crucial step in data preprocessing and is widely used in various machine learning and statistical modeling applications. Derived features are often more informative and relevant to the problem at hand than the original raw data. Common methods used for deriving features include mathematical operations such as addition, subtraction, multiplication, division, and exponential.

In the present application, we opt for more related characteristics such as north velocity (NV), east velocity (EV), direction (H), direction measurement, shape distance, and shape radius. Then, we derived velocity (S) from NV and EV, direction velocity (HR) from direction (H), and transverse acceleration (TA) using velocity (S) and direction velocity (HR) as shown in Table 5.2.

5.1.3.4 Data Labelling

Data labeling is the process of assigning predefined tags or labels to data points or instances for classification and analysis. It is an essential step in supervised learning, where algorithms learn from labeled data to make predictions about new invisible data [151]. Data labeling can be done manually by human annotators or automatically using algorithms. Data labeling can be a time-consuming and costly process, especially when large amounts of data need to be labeled [152]. There are several challenges associated with data labeling, such as label noise,

Table 5.2: Feature Derivation

No.	Original parameters	Derived parameters	Formula
1.	north_velocity (m/s) east_velocity (m/s)	speed	$\sqrt{\text{northVelocity}^2 + \text{eastVelocity}^2}$
2.	speed Timestamp	longitudinal acceleration (L-Acc)	$\frac{\text{speed}_{i+1} - \text{speed}_i}{\text{timestamp}_{i+1} - \text{timestamp}_i} \times 1000$
3.	heading (deg) Timestamp	HeadingRate	$\frac{\text{heading}_{i+1} - \text{heading}_i}{\text{timestamp}_{i+1} - \text{timestamp}_i} \times \left(\frac{1000 \times 3.14}{180}\right)$
4.	HeadingRate, Speed	Transversal acceleration (T-Acc)	$\frac{\text{speed}_i + \text{speed}_{i+1}}{2} \times \text{headingRate}_{i+1}$

where incorrect labels are assigned to data points, and label bias, where assigned labels are not representative of data distribution. To mitigate these issues, it is essential to carefully design the labeling process and have quality control measures in place to ensure the accuracy and consistency of the labels assigned. In our case, we label the data manually under the supervision of drivers and experts. We label normal data as '0' and anomalous data as '1'.

5.1.3.5 Data Filtering

Environmental factors such as satellite view, signal blocking, and weather conditions can affect accuracy [153]. To remove signal noise, we apply two data filtering techniques:

a. Svitzky-Golay filter (SGF):

We applied a Savitzky-Golay (SGF) leveling filter to selected characteristics, applied specifically to velocity and direction profiles, before calculating their derivatives (e.g., HR, TA). SGF is a digital filter [133], well applied in GPS trajectory data, which we have adapted to our speed and direction time series dataset to increase data accuracy without distorting the frequencies and shape of the actual signal, reducing noise and determining a blunt trend line to derive the other parameters. It is a type of linear filter that uses a sliding window to perform local regression on data points within the window [154]. The Savitzky-Golay filter works by adapting a polynomial function to the data points within the sliding window and then using polynomial coefficients to estimate rounded or differentiated values. The degree of the polynomial and the size of the sliding window can be adjusted to control the leveling or differentiation effect. In our case, we opt for the window size of 160 and the polynomial order of 2 for the sake of the uniform spline.

b. Denoising autoencoder

Denoising autoencoders are a type of neural network architecture that can be used for noise reduction in time series data [155]. They consist of a network of encoders that maps the input data to a smaller size representation and a network of decoders that maps the smaller dimensional representation to the original input data. The denoising autoencoder is trained to reconstruct the original data from the smaller dimensional representation, while being noise-resistant in the input data [156]. We applied the denoising autoencoder to selected features by including 3 convolutional levels in the encoder and 3 deconvolutional levels in the decoder to reconstruct the latent vector in the original input.

After initial data preprocessing, we perform exploratory data analysis as shown in Figure 5.2 , to observe the role of each parameter with respect to the anomaly. Exploratory data analysis helps to improve the performance of the model by eliminating the useless parameters.

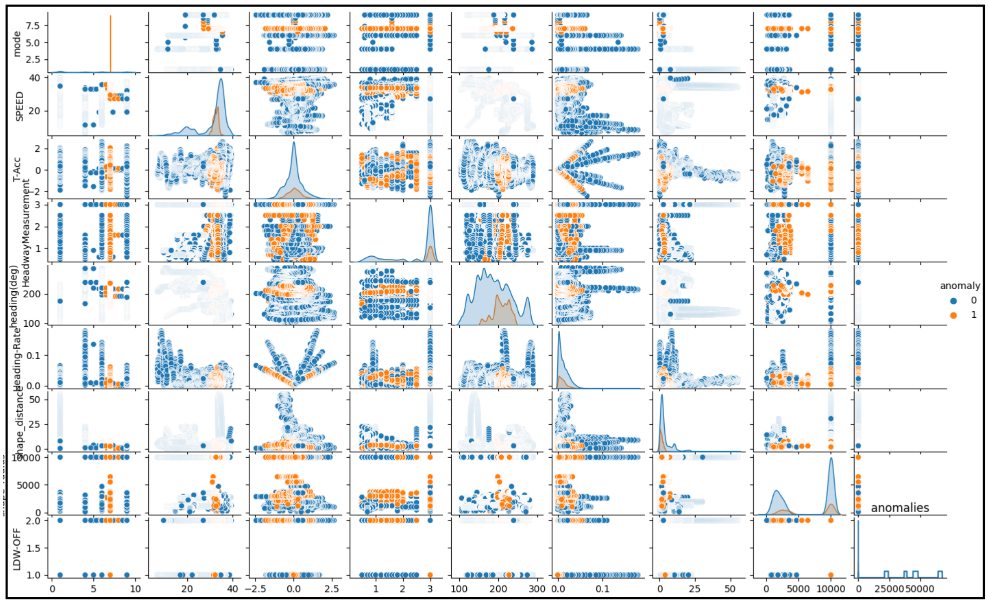


Figure 5.2: Exploratory data analysis on the basis of anomaly

5.1.4 Methodology

This section will detail the different machine-learning techniques used for anomaly detection. We will start by providing an explanation of each method, outlining how they work. Next, we will discuss the results obtained from our experiments and a comparison is carried out between these standard anomaly detection schemes and the proposed method in the following sections.

5.1.4.1 Principal component analysis (PCA)

Major component analysis (PCA) is a useful machine-learning technique for anomaly detection [157]. PCA works by transforming high-dimensional data into a smaller dimensional space defined by major components [157]. These key components capture the most significant variation in the data and can help simplify and identify patterns in the dataset. In anomaly detection, PCA can be used to identify outliers and deviations from predicted patterns in the dataset [158]. By identifying data points that fall outside the normal range defined by the major components, anomalies can be reported and analyzed further. In detail, PCA components are to identify anomalous elements in a data set by analyzing the reconstruction error. Essentially, this involves decomposing the matrix of source data into its main components and then reconstructing the original data using only some of the most significant major components. By comparing the reconstructed data with the original data, any anomalies can be identified as leading to a higher reconstruction error. However, it is important to note that PCA is an unsupervised technique and may not be able to capture all kinds of abnormalities. In cases where abnormalities are more complex or subtle, other machine learning techniques such as neural networks may be more effective. However, PCA can still be a useful tool in a broader suite of anomaly detection methods

5.1.4.2 One-Class Support Vector Machines (OSVM)

OSVM (One-Class Support Vector Machines) is a machine learning method often used for anomaly detection. Unlike other SVMs that are trained on data labeled for classification, OSVMs are designed to work on unlabeled data and identify normal patterns in the data [159]. The purpose of OSVM is to find a decision boundary that separates the normal given points from the anomalous points. This limit is represented by a hyperplane that maximizes the margin between the normal data points and the hyperplane. The hyperplane is found by solving a constrained optimization problem, where constraints ensure that all normal data points are on one side of the hyperplane while anomalous points are on the other side. Once the hyperplane is determined, new data points can be classified as normal or abnormal based on the side of the hyperplane on which they fall. If a data point falls on the same side as normal data points, it is considered normal.

If it falls on the other side, it is considered abnormal.

However, OSVM can have difficulty identifying anomalies in datasets with high noise levels or complex patterns. Therefore, it may be useful to integrate OSVM with other machine learning methods for greater accuracy in anomaly detection.

5.1.4.3 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a type of artificial neural network (ANN) designed to process data with a grid-like topology, such as images or time series. CNNs were originally developed for computer vision tasks, such as image classification, object detection, and segmentation. CNNs are a type of deep learning algorithm originally designed to process images efficiently, but have also been used for anomaly detection [115]. The architecture of a CNN includes convolutional layers, pooling layers, and fully connected layers. Convolutional layers contain filters in the form of a weighted matrix (C1) that efficiently recognize patterns by reducing the dimensionality of variables. Pooling levels (S) summarize features and can be repeated multiple times [160]. The final layer is the fully connected layer, in which neurons (NN) take the extracted features as input. Fully connected layers use the extracted features to make an estimate. They automatically extract features from data, which can be used for classification.

Additional information regarding CNN is explored in the preceding chapters from a technical perspective.

5.1.4.4 Proposed Scheme

In the proposed methodology, we use a convolutional autoencoder to detect anomalies in time series data. In our work we use Convolutional Autoencoders and, therefore, we will provide the initial information of Autoencoder (AE). An AE is a type of artificial neural network popular for anomaly detection, AE consists of two main modules: the encoder and the decoder (Figure. 4.7). The encoder maps the input data into a latent vector while the decoder attempts to reconstruct the input from the latent vector [161][81].

Note that convolutional autoencoders (CAEs) can learn the most useful feature patterns in input data. To detect anomalies in the time series data, the auto-encoder is first trained on a normal time series dataset. During the test phase, the input data is passed through the trained autoencoder, and the reconstruction error is calculated. Data anomalies can be detected by comparing the reconstruction error with a threshold value. The use of convolutional layers in the autoencoder architecture enables efficient feature extraction and reconstruction of time series data, making it a popular choice for anomaly detection in time series data.

We call the proposed comprehensive scheme for anomaly detection in the scenario of interest: "Road safety through deep learning-based detection of abnormal driver behavior" (ROAD-DAD). Road-DAD uses a 1D CAE where the input consists of a sequence of time data samples X 1, X 2, . . . , X7 generated at a frequency of 20 Hz. The generic Xi is a 7-tuple of values, i.e. representing speed, direction, direction velocity, forward measurement, transverse acceleration, distance, and radius, calculated as discussed in section 5.1.3.3 above. Therefore, the input data is of a 2-dimensional matrix nature that we flatten as a sequence of 1-dimensional input. The number of j-samples that make up the input sequence as the window size. Experiments show that a good value for j is j = 64. Therefore, the input size of the encoder is n = 64×7 = 448. The overall architecture of Road-DAD in which the encoder consists of 4 convolutional levels. We applied Stride as an advanced convolutional parameter that is able to replace the maximum pool with less computing. Padding is used to maintain the output size as input while the activation function is responsible for activating neurons. In our case, each convolutional layer reduces the input size by a factor equal to the step, that is, two. Consequently, the output of the first convolutional layer has a dimension equal to 224, while the output of the 4th layer, i.e. the latent vector Z, has a dimension equal to m=56. The decoder consists of two levels of deconvolution and a dropout layer that prevents the model from being overfitted. The output of the decoder will again have a size equal to 448 and compared with the input by calculating the mean absolute error (MAE). If this error is above a certain level of threshold, an anomaly warning is issued such as '1' otherwise '0'. So, we give high weights to "1" and low weights to "0". After setting weights against anomalous and normal data points, we set the 2nd level threshold to identify and count anomalies for evaluating results as illustrated in Figure 5.3.

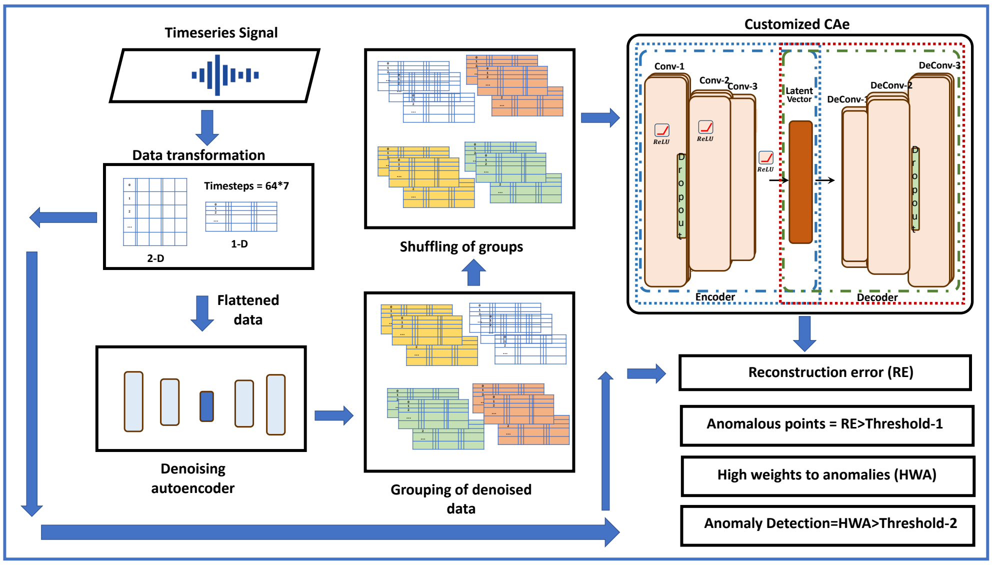


Figure 5.3: Table 3. F-Score based performance evaluation for proposed scenarios.

5.1.5 Result

The proposed scheme will be validated by comparing the actual CSE detected, which we call "real positives", with the anomalies, which we call "CNN positives". For comparison, similar validations were performed using principal component analysis as a robust standard statistical approach for feature reduction and anomaly detection, and the most widely used method compared to One-Class Support Vector Machines (OSVM) which is well known for anomaly classification and detection. In addition, we also compared the results of ROAD-DAD with the standard convolutional autoencoder.

We utilize recall and precision as performance metrics. The preceding chapters have already elaborated on how these metrics function, how they calculate results, and how performance is assessed.

5.1.5.1 Model testing and validation

We perform different experiments using various ML and statistical methods for anomaly detection such as PCA, OSVM, CAe, etc. Initially, we use the standard convolutional autoencoder and obtain the following recall and precision for the two scenarios "test" and "validation" as shown in Figure 5.4, 5.5 and 5.6 for PCA, OSVM and CAe respectively.

Testing means that we use the dataset collected from a road section for training and testing, while in validation we test the model with the dataset of different road sections.

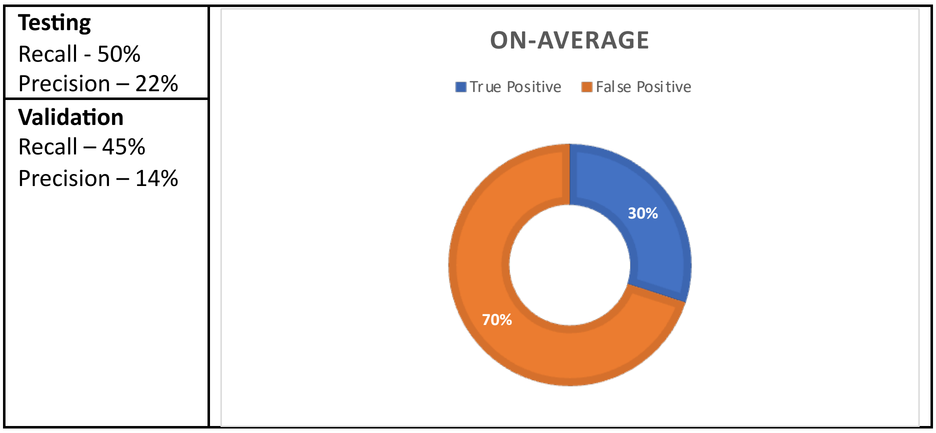


Figure 5.4: True positive and False positive by exploiting PCA

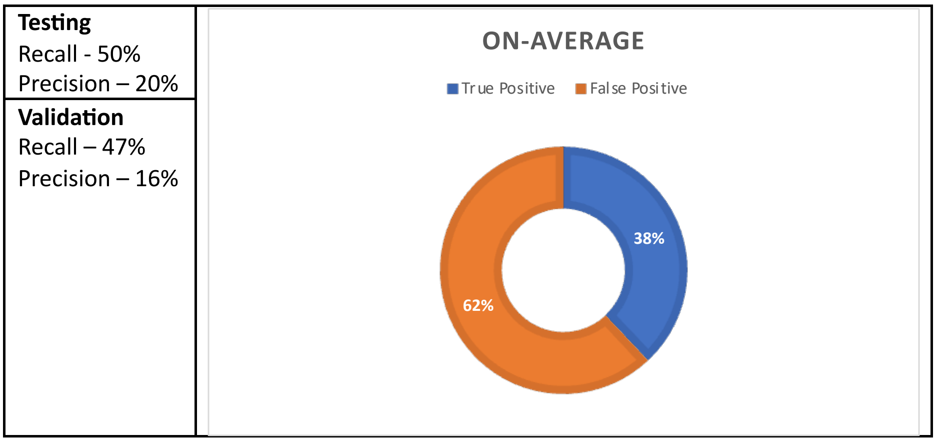


Figure 5.5: True positive and False positive by exploiting OSVM

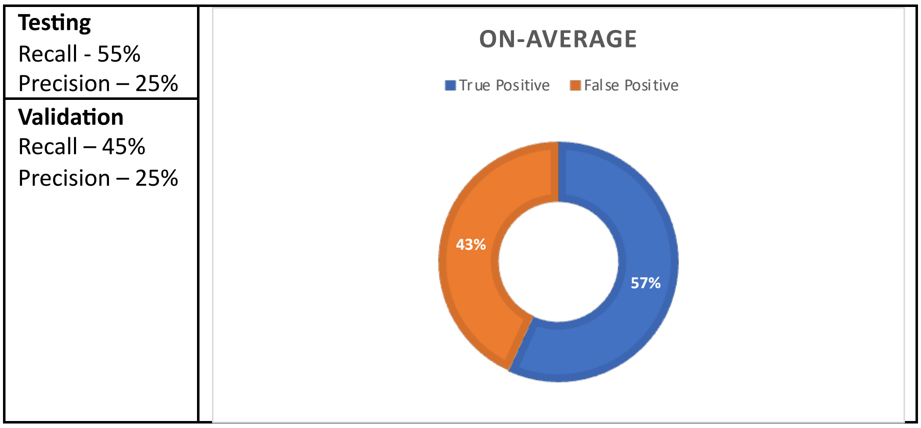


Figure 5.6: True positive and False positive by exploiting standard CAe

Observation and next idea:

After taking advantage of the standard convolutional autoencoder, we observe that the accuracy of anomaly detection is limited as we have reached 57% of the real detection while the false alarm rate is very high at 43%. Therefore, we plan to integrate another algorithm to create data groups and shuffle for different road sections. We also optimize model parameters such as trigger function, and latent vector size, and add some dropout levels to avoid model overfitting issues. In addition, we plan to filter the dataset as noise causes a high rate of false alarms. After implementing these ideas, we achieved results as shown in Figure 5.7.

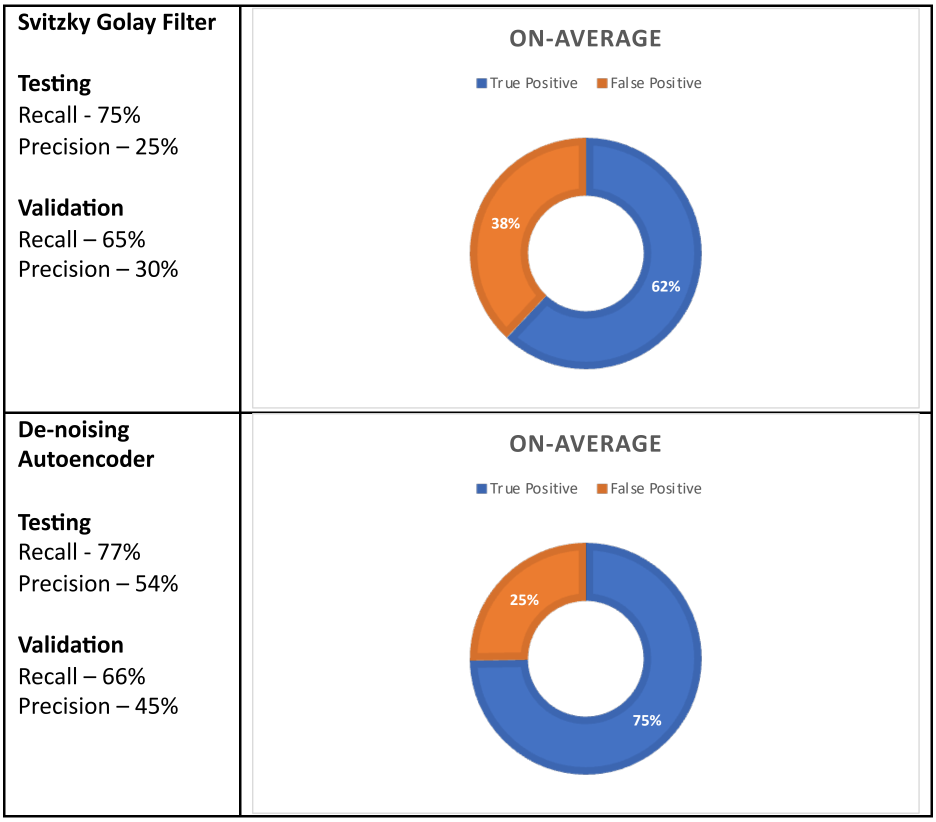


Figure 5.7: True positive and False positive by using filters and customized CAe

Observation and next idea:

Although the Denoising-autoencoder significantly improves false positives, but for more promising results we deepen the exploratory analysis of the data as shown in Fig. 3. We have observed that anomalies are always surrounded by normal points, so we plan to weigh the anomalies at the top. Then we reach the results as indicated in the Figure 5.8.

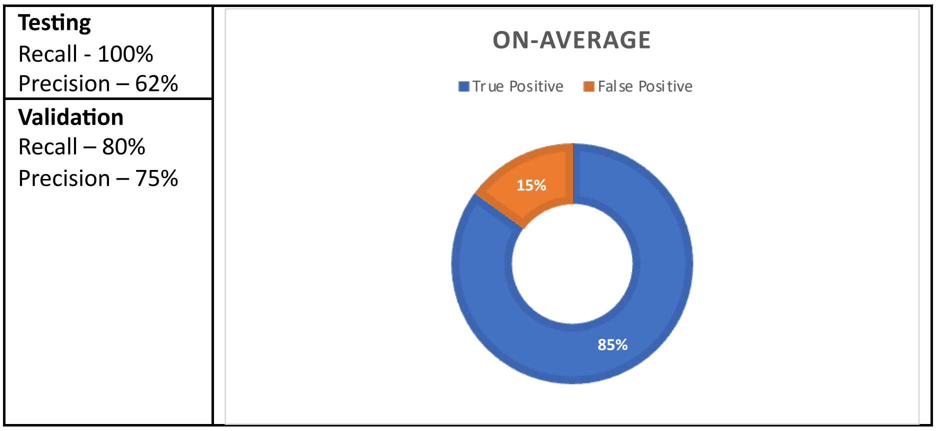


Figure 5.8: True positive and False positive of proposed model

In the end, we concluded our results based on several scenarios, as shown in Figure 5.9.

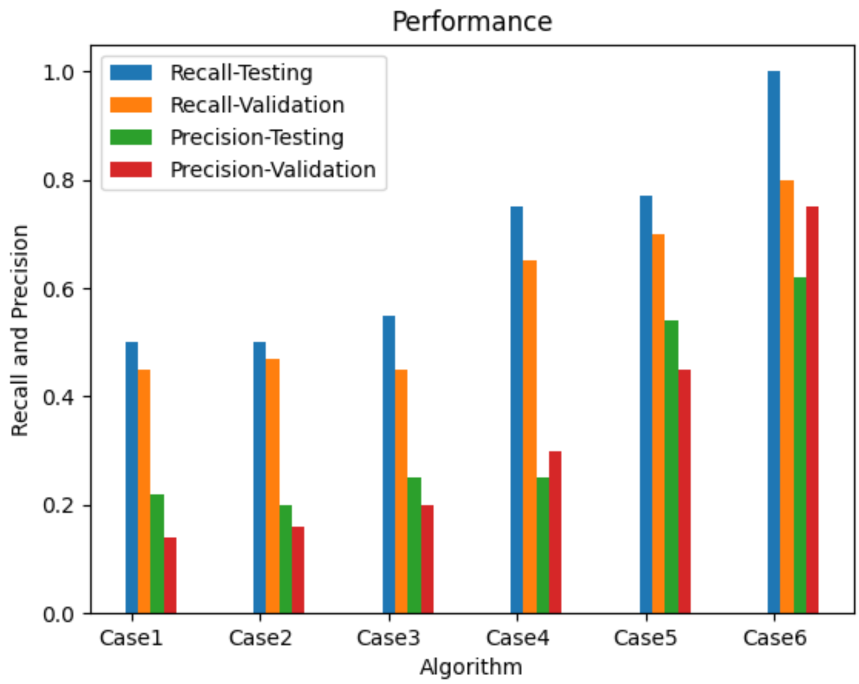


Figure 5.9: Recall and Precision for all cases

5.1.5.2 Real-time anomaly detection by exploiting Long-short Term Memory Autoencoder

Using Long Short-Term Memory (LSTM) neural networks for real-time anomaly detection is a powerful and effective approach, particularly in time-series data analysis. LSTMs are a type of recurrent neural network (RNN) that is well-suited for sequential data, making them a valuable tool for detecting anomalies in streaming or time-series data.

LSTM autoencoder for real-time anomaly detection is a powerful technique for identifying abnormal patterns or outliers in sequential data. LSTM autoencoders leverage the ability of Long Short-Term Memory (LSTM) networks to model sequential dependencies and learn compact representations of input data. To Build an LSTM autoencoder, it consists of two parts: an encoder and a decoder.The encoder LSTM layer compresses the input data into a lower-dimensional representation, which is known as the bottleneck or latent space. The decoder LSTM layer attempts to reconstruct the original input data from the compressed representation. The model aims to minimize the reconstruction error during training.

For real-time anomaly detection, we continuously feed new data into the trained LSTM autoencoder model in real-time. For each new input sequence, encode it using the trained encoder and then attempt to reconstruct it with the decoder. Calculate the reconstruction error between the original input and the reconstructed output and then find anomaly by fixing threshold.

We achieved the following outcomes by leveraging an LSTM autoencoder and utilizing data from numerous users, similar to the approach outlined previously but on a larger scale compartively. In Figure 5.10, the results show that successfully detected anomalies are categorized as true positives, while those that are erroneously flagged are considered false alarms or false positives. Figure 5.11 illustrates the rate of successful anomaly detection as true positives, contrasting with cases where the model fails to detect anomalies, which are labeled as false negatives.

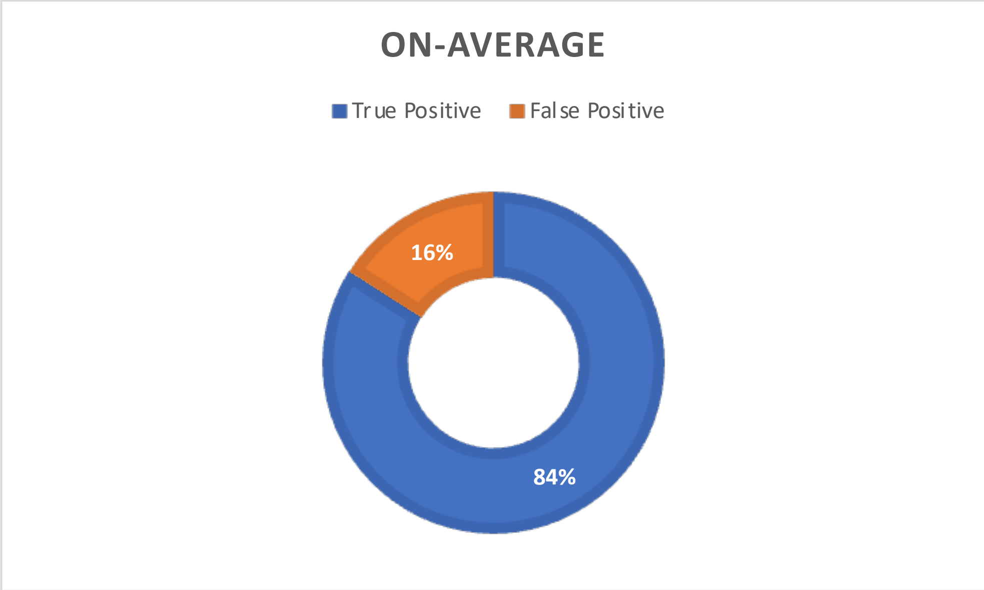


Figure 5.10: Rate of true positive and false positive

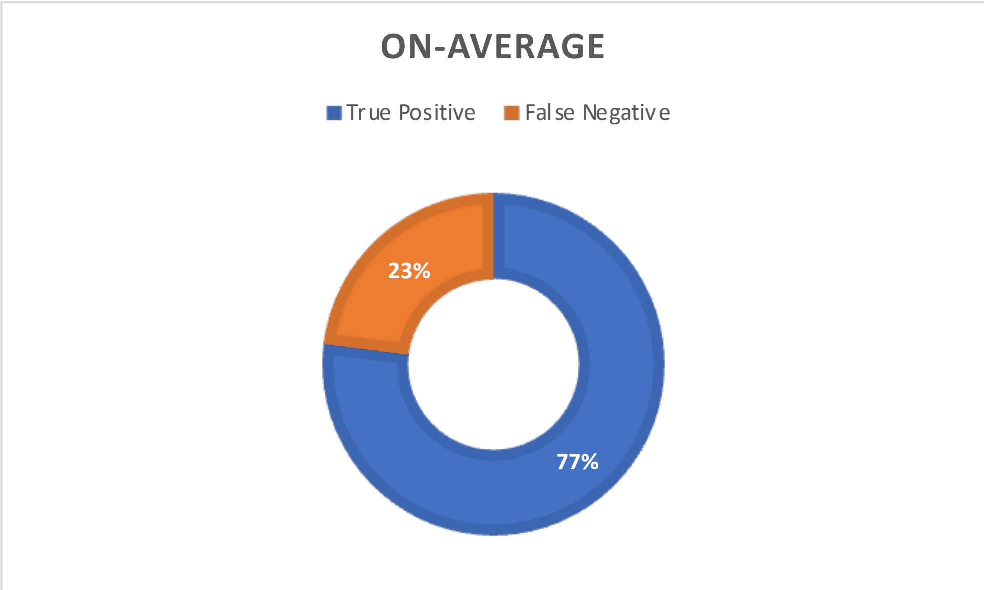


Figure 5.11: Rate of true positive and false positive

5.1.5.3 Remarks and Future Directions

In this chapter, we employ a vehicle as a data source, equipped with a range of sensors including GPS, OBD, Mobileye, etc. Data is collected from these sensor-equipped vehicles in the form of time series signals. The collected time series data undergoes several processing techniques such as interpolation, feature extraction, and data filtering to obtain a clean and meaningful dataset. Following data preparation, we feed the dataset into our proposed system named "ROAD-DAD" for the purpose of identifying abnormal or anomalous driver behavior. ROAD-DAD leverages a Convolutional Autoencoder, complemented by our own algorithms for efficient anomaly detection.

We conduct a comparative analysis of our proposed scheme with well-established machine learning-based anomaly detection algorithms, including a standard Convolutional Autoencoder. The evaluation of results is based on the recall and precision evaluation metrics. We have achieved an 85% recall rate with a 15% precision rate, which is considered a favorable outcome. In the future, we are intended to explore more the same dataset parameters on larger scale using Long Short-Term Memory (LSTM) autoencoder and deep transfer learning for various users.

5.2 Multi-sensor-based analysis for anomaly detection

5.2.1 Initial Information

Abnormal driver behavior detection refers to identifying and analyzing actions, movements, or patterns exhibited by drivers that deviate from standard or safe driving practices [146]. This concept is commonly applied in the context of advanced driver assistance systems (ADAS), vehicle safety, and driver monitoring technologies[124].

Various sensors and data sources can be used to detect abnormal driver behavior, including:

- **In-vehicle Cameras:** Cameras installed inside the vehicle can monitor the driver's face, eyes, and body movements to detect signs of fatigue, distraction, or other abnormal behaviors.

- **Steering Wheel and Pedal Sensors:** These sensors can capture the driver's input, allowing the system to detect aggressive or erratic driving behavior, such as sudden braking or sharp turns.

- **GPS and Location Data:** Analyzing the vehicle's location and trajectory can help to identify abnormal driving patterns, like excessive speeding or sudden lane changes.

- **Accelerometers and Gyroscopes:** These sensors can detect sudden changes in vehicle acceleration or orientation, which may indicate reckless driving or potential collisions.

- **OBD-II:** Data from the vehicle's onboard diagnostic (OBD-II) system can be analyzed to assess driving behaviors like speeding, hard braking, and rapid acceleration.

Machine learning algorithms are often used to process the collected data and detect abnormal behaviors. These algorithms are trained on labeled datasets that contain examples of both normal and abnormal driving behaviors. The model then learns to recognize patterns and anomalies in real-time data to make accurate predictions.

Some examples of abnormal driver behaviors that can be detected include: Drowsy or fatigued driving, Distraction (e.g., using a mobile phone while driving), Aggressive driving (e.g., tailgating, sudden lane changes), Drunk or impaired driving, Failure to obey traffic rules (e.g., running red lights). By detecting abnormal driver behavior, ADAS and other safety systems can intervene and provide warnings to the driver or take control of the vehicle if necessary to prevent accidents and enhance road safety [2]. Additionally, this data can be valuable for fleet management and driver training purposes.

In this chapter, we present deep learning-based anomaly detection scheme for various sensor-based data. Considered sensors are GNSS, Map, OBD and mobileye.

5.2.2 Related Work

Anomaly detection and classification are two related concepts used in various fields to identify unusual patterns, behaviors, or events in data . Both are techniques used to spot deviations from the norm, but they serve different purposes and are applied in distinct scenarios. Let's explore each of these concepts:

5.2.2.1 Anomaly Detection:

Anomaly detection, also known as outlier detection, is the process of identifying data points or instances that deviate significantly from the expected or normal behavior of the dataset [81]. These anomalies can represent errors, fraud, unusual events, or any other observations that are rare and distinct from the majority of the data. Anomaly detection techniques can be applied in various domains, including cybersecurity, finance, healthcare, industrial monitoring, and more. Some popular anomaly detection methods include:

a. Statistical methods: These techniques use statistical distributions and models to identify outliers based on the assumption that anomalies are infrequent and far from the mean.

b. Machine learning approaches: Machine learning algorithms, such as clustering, one-class SVM (Support Vector Machines), or isolation forests, can be used to learn the normal behavior of the data and then identify instances that do not fit the learned patterns as anomalies.

5.2.2.2 Classification:

Classification, on the other hand, is a supervised learning technique where the objective is to assign data points to predefined classes or categories based on their features. In this task, the algorithm is trained on a labeled dataset, where each instance has a known class label. The primary goal of classification is to learn a model that can accurately predict the class of unseen data points based on their features. Common classification algorithms include decision trees, random forests, logistic regression, support vector machines (SVM), and neural networks.

The main difference between anomaly detection and classification lies in their objectives and the nature of the datasets they work with. In summary, anomaly detection aims to identify rare and abnormal instances without using labeled data, while classification focuses on assigning data points to predefined classes using labeled data. Both concepts play crucial roles in data analysis and machine learning applications, and their effective use depends on the specific problem and data available.

5.2.2.3 Anomaly Detection Methods:

Anomalies in sensor data can indicate critical events or errors. Well-renowned anomaly detection methods include:

- Statistical approaches (e.g., z-score, box plot, Grubbs' test)
- Machine learning-based methods (e.g., Isolation Forest, One-Class SVM)
- Deep learning techniques (e.g., autoencoders, LSTM networks)

Emphasis is placed on adapting these methods to suit the characteristics of different sensors and application domains.

5.2.3 Problem Statement

With the increasing proliferation of sensors in various fields, there arises a need for comprehensive sensor-fusion analysis and efficient anomaly detection methods. This research aims to address this need by investigating the utilization of different types of sensors in developing robust algorithms for anomaly or abnormal driver behavior detection. The study seeks to contribute to the enhancement of data-driven decision-making processes and the reliability of sensor-based systems.

Modern industries and systems heavily rely on sensor data for monitoring, control, and optimization. From environmental monitoring to industrial automation and healthcare, sensors provide valuable insights into various processes [17]. However, anomalies, unexpected events, or sensor malfunctions can compromise system performance and data quality[19]. This research focuses on exploring sensor-based analysis techniques and developing anomaly detection methods to ensure the accuracy and dependability of sensor-generated data.

5.2.4 Proposed Method

There are Different approaches for Sensor-Based Analysis such as :

- Signal processing and feature extraction methods
- Time-series analysis and trend identification
- Data fusion and sensor integration
- Pattern recognition and machine learning techniques
- Visualization and interpretation of sensor data

"We employ a deep learning-based pattern recognition approach for detecting anomalies, utilizing data from different sensors individually. We apply the method we propose, referred to as ROAD-DAD, which is extensively described in Chapter 5 (I)."

5.2.5 Results

We assess performance through two types of comparisons. The initial comparison involves employing the "ROAD-DAD" method with various sensor combinations for anomaly detection. The second comparison entails the use of the classification technique SVM.

5.2.5.1 Anomaly detection by exploiting "ROAD-DAD" against various sensor combinations

In this part, we exploit "ROAD-DAD" for anomaly detection using various combinations of sensors and observe which work well. In Figure 5.12, model testing is carried out by using the dataset of the same user and road section for training and testing.

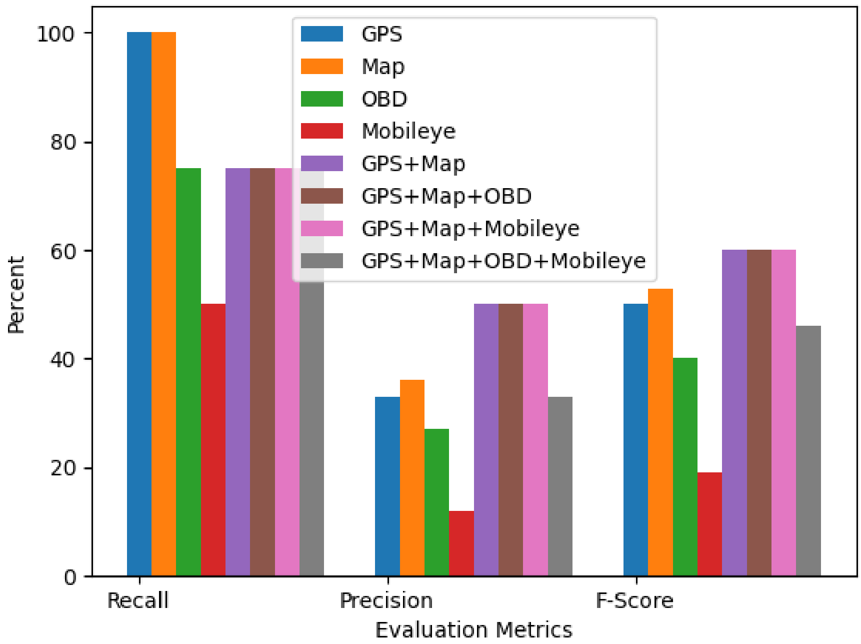


Figure 5.12: Anomaly detection using same road section and different sensors

In Figure 5.13, model validation is carried out by using the dataset of different user and road sections for the training and testing phase.

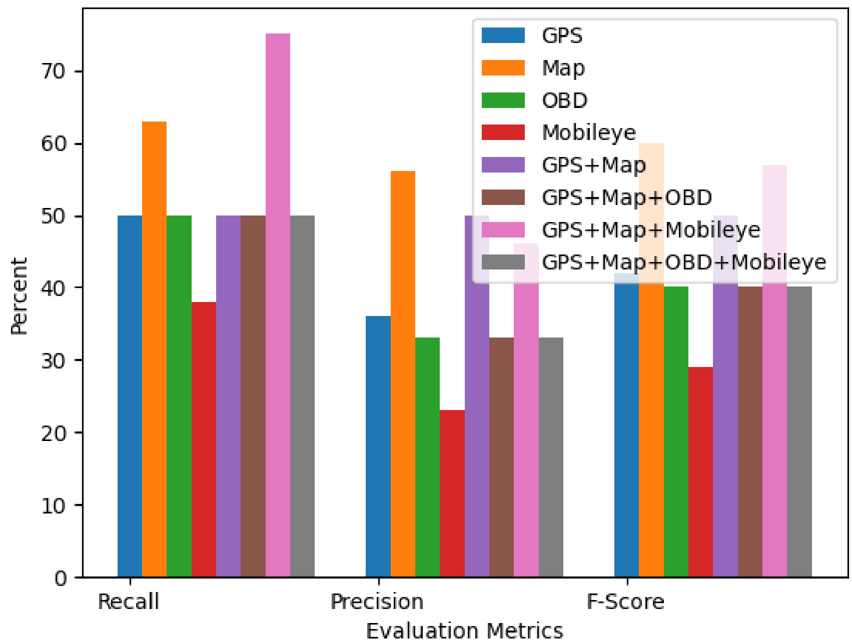


Figure 5.13: Anomaly detection using different road sections and different sensors

From Figure 5.12 and 5.13, we observe that data collected from GPS, map and mobileye work well while data from OBD causes less true positive and high false positive.

5.2.5.2 Classification by exploiting "SVM" against various sensor combinations

In Figure 5.14, the orange lines represent true positives, while the blue lines represent positives detected by CNN. Notably, we observe an increase in false positives between 20,000 and 40,000 sample points. Consequently, we decided to employ a classification algorithm to perform multi-class classification, distinguishing between normal, abnormal, and other maneuver points, as depicted in Figure 5.15. These 'other maneuvers' could encompass actions such as overtaking vehicles, traffic lights, or roadside work. We found that the points between 20,000 and 40,000 were not actually anomalous but rather fell into the category of other road maneuvers, a classification achieved successfully using the SVM classification algorithm

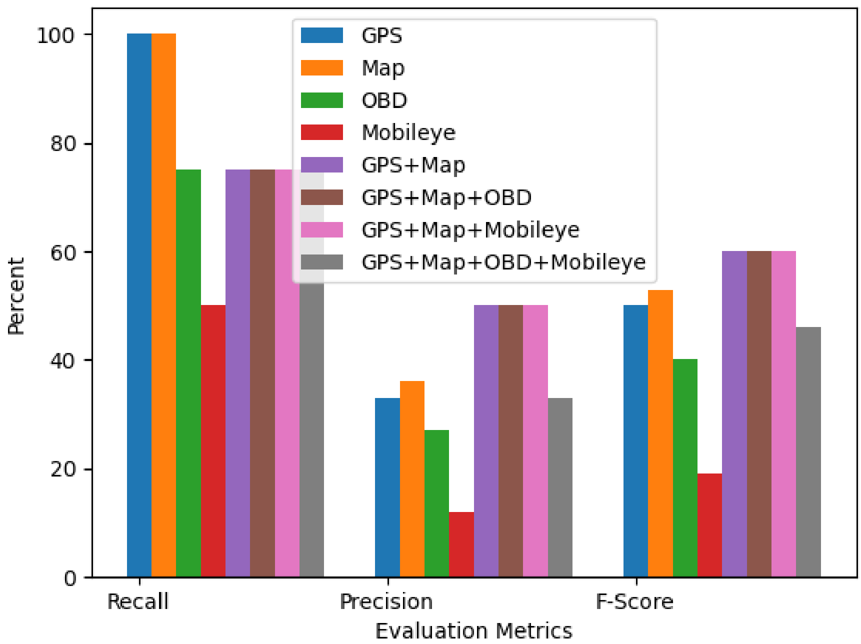


Figure 5.14: Anomaly detection using different road section and different sensors

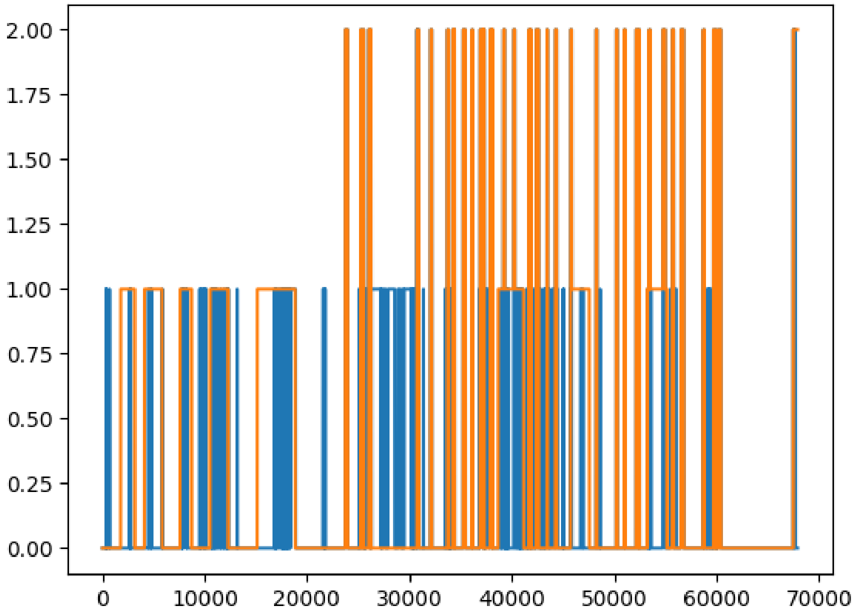


Figure 5.15: Classification of normal, abnormal and other road maneuvers

5.2.6 Limitations

The research contributes to the advancement of sensor-based analysis for anomaly detection methods, fostering reliable and efficient data-driven decision-making in diverse applications. By understanding the unique characteristics of different sensors and addressing the challenges of anomaly detection, this study aims to pave the way for more resilient and accurate sensor-based systems in the future. Furthermore, it explores the use of edge computing for real-time anomaly detection, enhancing the interpretability of deep learning models, and addressing security concerns related to sensor data.

Chapter 6

Conclusion and Future Directions

The thesis concludes by summarizing its significant contributions and findings. Initially, it categorizes driver behavior detection solutions into three primary types: those dependent on driver health parameters, those utilizing vehicle parameters, and those considering both. It explores various methods for data collection, including cameras and sensors, and examines their applications in real-time and simulation-based scenarios. A comprehensive taxonomy is introduced to enhance comprehension of driver behavior detection schemes, followed by an analysis section pinpointing research gaps and future directions.

During this investigation, it was observed that data collection for driver behavior detection primarily revolves around two key methods: the utilization of cameras and sensors. These data sources encompass smartphone cameras, capable of capturing valuable image-based information, and vehicle-mounted cameras, which provide visual data. Sensors play a critical role by gathering diverse data from vehicles and drivers in signal form. The research also distinguishes between test environments, differentiating between real-time situations grounded in the actual world and non-real-time scenarios involving simulation-based driver behavior detection. Real-time techniques involve the assessment of results using real-time data, whereas non-real techniques rely on simulated data for analysis.

A fundamental contribution of this study is the creation of a comprehensive taxonomy, designed to provide a clear understanding of the diverse array of driver behavior detection schemes found in the literature. This taxonomy serves as a valuable reference for researchers and practitioners interested in this field. Subsequently, the analysis, stemming from this extensive literature review and taxonomy construction, identified significant gaps in the current body of knowledge.

The thesis then shifts its focus to cyclist safety and the need for innovative observational studies in smart cities. It introduces the use of CNN and autoencoders for anomaly detection in cycling behavior, specifically the combination of speed and direction GPS data. The performance of the model is assessed, showing promising results in terms of recall and precision. The study suggests the potential for using such models to improve cycling safety in urban areas.

Then, the thesis incorporates transfer learning to significantly reduce the amount of data required to build an effective artificial neural network. The experimental results, conducted within a specific case study, demonstrate the promising performance of the proposed approach. It achieves an average recall of 77%, with some individual users achieving a perfect 100% recall rate. Moreover, the potential for transferring a pre-trained model to different cyclists with satisfactory results suggests scalability and efficiency in large-scale applications. Incorporating direction information, such as heading and transversal acceleration, significantly enhances the model's ability to detect anomalies in cycling. The use of data filtering and Convolutional Neural Networks (CNN) further reinforces the model's robustness, outperforming traditional statistical techniques like PCA and heuristic threshold-based methods. A case study illustrates the practical applicability and consistency of risk assessment and ranking.

The next activity of the thesis is to introduce a methodology aimed at addressing the fundamental question of whether patterns identified by convolutional layers in CNNs and CAEs are more user-specific or environment-specific. The significance of resolving these questions has been emphasized.

Furthermore, within the proposed methodology, the ROAD-DAD phase emerged as a significant component. This phase introduced a deep learning model grounded in the principles of convolutional autoencoders, accompanied by a self-directed algorithm tailored for anomaly detection. Leveraging data from vehicles, including speed, acceleration, heading, and more, this model was devised to identify irregular behaviors or anomalies. The model's efficacy was confirmed through rigorous benchmarking against established machine learning methods widely employed in anomaly detection.

In summary, this aspect of our proposed methodology delved into the fusion of data derived from diverse sensors, all originating from the vehicle itself. These sensors encompassed GPS, OBD, and Mobileye, collectively contributing to a comprehensive data pool. Our overarching goal was to harness this amalgamated sensor data in conjunction with the ROAD-DAD model for the explicit purpose of detecting anomalies in driver behavior.

Finally, the thesis concludes by discussing the practical application of the proposed methodology in the city of Catania, Italy, and suggests techniques for leveraging the research findings for practical purposes. It underscores the significance of addressing issues related to user-specific and environment-specific patterns within deep learning models.

While this study provides valuable insights and promising results, several avenues for future research and practical application emerge. One crucial consideration is the potential for larger deviations in model performance with an increased number of users. Recognizing that the CNN model depends on both the user and specific road environments, applying transfer learning and cooperative learning in real time to adapt the model to individual users and road conditions is a promising area of exploration.

To improve the practical application of the model, there is a need for more suitable data recording methods or dedicated devices installed in bike-sharing fleets. Standard smartphones and applications currently do not offer the high-frequency data acquisition required for capturing evasive maneuvers, which can involve rapid speed and direction changes within seconds.

Furthermore, considering that cyclists adapt their behavior to various road infrastructures and traffic conditions, enhancing the model's adaptability remains a priority. Future research may focus on tailoring specific model layers to individual users while maintaining flexibility in adapting other layers to the current road environment.

Lastly, the study introduces an intriguing question about whether the patterns recognizable by convolutional layers in neural networks are more specific to the user or the environment. Resolving this question could provide valuable insights into refining anomaly detection models.

In summary, this research represents a significant step forward in enhancing road safety through advanced anomaly detection techniques. The promising results and identified areas for future research underscore the importance of continued efforts to protect vulnerable road users and promote safer urban environments.

List of publications

1. Yaqoob, S., Hussain, A., Subhan, F., Pappalardo, G., and Awais, M. (2023). Deep Learning Based Anomaly Detection for Fog-Assisted IoVs Network. IEEE Access, 11, 19024-19038.Digital Object Identifier 10.1109/ACCESS.2023.3246660

2. Yaqoob, S., Cafiso, S., Morabito, G., and Pappalardo, G. (2023). Detection of anomalies in cycling behavior with convolutional neural network and deep learning. European Transport Research Review, 15(1), 9.

3. Yaqoob, S., Cafiso, S., Morabito, G., and Pappalardo, G. (2022). Monitoring Bicycle Safety through GPS data and Deep Learning Anomaly Detection.

4. Yaqoob, S., Cafiso, S., Morabito, G., and Pappalardo, "Road Safety: Deep Transfer Learning based anomaly detection", Journal of Safety Research, 2023.

5. Yaqoob, S., Cafiso, S., Morabito, G., and Pappalardo, "Driver behavior assessment through E-health and vehicle monitoring", IEEE transaction on Intelligent Transportation System, 2023. (submitted)

6. Yaqoob, S., Morabito, G., Cafiso, S.,and Pappalardo, "On the role of convolutional layers in ML applications involving driver-road interactions: analysis and practical implications", Pattern Recognition, 2023. (submitted)

7. Yaqoob, S., Cafiso, S., Morabito, G., and Pappalardo, G. (2022). Monitoring Bicycle Safety through GPS data for Anomaly Detection, ICTCT 34 conference, Hungary.

8. Yaqoob, S., Morabito, G., Cafiso, S.,and Pappalardo, "RoadSafetyAI: Multisensor fusion for abnormal driver behavior identification".

9. S. Yaqoob, A. Ullah, M. Awais, I. Katib, A. Albeshri, R. Mehmood, M. Raza, S. Ul Islam, and J. J. P. C. Rodrigues, "Novel congestion avoidance

scheme for Internet of Drones," Comput. Commun., vol. 169, pp. 202–210, Mar. 2021, doi: 10.1016/j.comcom.2021.01.008.

References

[1] Sinan Kaplan et al. “Driver behavior analysis for safe driving: A survey”. In: *IEEE Transactions on Intelligent Transportation Systems* 16.6 (2015), pp. 3017–3032.

[2] Yunyun Zhao et al. “An abnormal driving behavior recognition algorithm based on the temporal convolutional network and soft thresholding”. In: *International Journal of Intelligent Systems* 37.9 (2022), pp. 6244–6261.

[3] Paul M Salmon et al. “Bicycle crash contributory factors: A systematic review”. In: *Safety science* 145 (2022), p. 105511.

[4] Francesco Barbato et al. “Road Scenes Segmentation Across Different Domains by Disentangling Latent Representations”. In: *arXiv preprint arXiv:2108.03021* (2021).

[5] Eunji Kim et al. “Safety monitoring system of personal mobility driving using deep learning”. In: *Journal of Computational Design and Engineering* 9.4 (2022), pp. 1397–1409.

[6] Najah AbuAli and Hatem Abou-Zeid. “Driver behavior modeling: Developments and future directions”. In: *International journal of vehicular technology* 2016 (2016).

[7] Salar Sadeghi Gilandeh, Mansour Hadji Hosseinlou, and Alireza Jafari Anarkooli. “Examining bus driver behavior as a function of roadway features under daytime and nighttime lighting conditions: Driving simulator study”. In: *Safety science* 110 (2018), pp. 142–151.

[8] Muhammad Qasim Khan and Sukhan Lee. “A comprehensive survey of driving monitoring and assistance systems”. In: *Sensors* 19.11 (2019), p. 2574.

[9] Danish Farooq, Sarbast Moslem, and Szabolcs Duleba. “Evaluation of driver behavior criteria for evolution of sustainable traffic safety”. In: *Sustainability* 11.11 (2019), p. 3142.

[10] Zhe Peng et al. “Vehicle safety improvement through deep learning and mobile sensing”. In: *IEEE network* 32.4 (2018), pp. 28–33.

[11] Apostolos Ziakopoulos and George Yannis. “A review of spatial approaches in road safety”. In: *Accident Analysis & Prevention* 135 (2020), p. 105323.

[12] Sarbast Moslem et al. “Application of the AHP-BWM model for evaluating driver behavior factors related to road safety: A case study for Budapest”. In: *Symmetry* 12.2 (2020), p. 243.

[13] Jennifer Howcroft et al. “Impact of health differences and longitudinal changes on deceleration driving patterns in older adult drivers”. In: *Transportation research part F: traffic psychology and behaviour* 60 (2019), pp. 137–146.

[14] Gulbadan Sikander and Shahzad Anwar. “Driver fatigue detection systems: A review”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.6 (2018), pp. 2339–2352.

[15] John HL Hansen et al. “Driver modeling for detection and assessment of driver distraction: Examples from the UTDrive test bed”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 130–142.

[16] Sarah Kadhim Alluhaibi, Munaf S Najim Al-Din, and Aiman Moyaid. “Driver behavior detection techniques: A survey”. In: *Int. J. Appl. Eng. Res* 13.11 (2018), pp. 8856–8861.

[17] Zouhair Elamrani Abou Elassad et al. “The application of machine learning techniques for driving behavior analysis: A conceptual framework and a systematic literature review”. In: *Engineering Applications of Artificial Intelligence* 87 (2020), p. 103312.

[18] Shiyang Yan et al. “Driver behavior recognition based on deep convolutional neural networks”. In: *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE. 2016, pp. 636–641.

[19] Leandro N Balico et al. “Localization prediction in vehicular ad hoc networks”. In: *IEEE Communications Surveys & Tutorials* 20.4 (2018), pp. 2784–2803.

[20] Sherrie-Anne Kaye, Ioni Lewis, and James Freeman. “Comparison of self-report and objective measures of driving behavior and road safety: A systematic review”. In: *Journal of safety research* 65 (2018), pp. 141–151.

[21] Mohammad Shahverdy et al. "Driver behavior detection and classification using deep convolutional neural networks". In: *Expert Systems with Applications* 149 (2020), p. 113240.

[22] Fabio Tango and Marco Botta. "Real-time detection system of driver distraction using machine learning". In: *IEEE Transactions on Intelligent Transportation Systems* 14.2 (2013), pp. 894–905.

[23] Jennifer M Cori et al. "An evaluation and comparison of commercial driver sleepiness detection technology: A rapid review". In: *Physiological measurement* 42.7 (2021), p. 074007.

[24] Federico Guede-Fernandez et al. "Driver drowsiness detection based on respiratory signal analysis". In: *IEEE access* 7 (2019), pp. 81826–81838.

[25] Ali Darzi et al. "Identifying the causes of drivers' hazardous states using driver characteristics, vehicle kinematics, and physiological measurements". In: *Frontiers in neuroscience* 12 (2018), p. 568.

[26] Mohammad Naim Rastgoo et al. "A critical review of proactive detection of driver stress levels based on multimodal measurements". In: *ACM Computing Surveys (CSUR)* 51.5 (2018), pp. 1–35.

[27] Sandipann P Narote et al. "A review of recent advances in lane detection and departure warning system". In: *Pattern Recognition* 73 (2018), pp. 216–234.

[28] Haiyan Zhang et al. "A framework of abnormal behavior detection and classification based on big trajectory data for mobile networks". In: *Security and Communication Networks* 2020 (2020), pp. 1–15.

[29] Oscar Oviedo-Trespalacios et al. "The impact of road advertising signs on driver behaviour and implications for road safety: A critical systematic review". In: *Transportation research part A: policy and practice* 122 (2019), pp. 85–98.

[30] Mkhuseli Ngxande, Jules-Raymond Tapamo, and Michael Burke. "Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques". In: *2017 pattern recognition Association of South Africa and Robotics and mechatronics (PRASA-RobMech)* (2017), pp. 156–161.

[31] German Castignani et al. "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring". In: *IEEE Intelligent transportation systems magazine* 7.1 (2015), pp. 91–102.

[32] Harpreet Singh and Ankit Kathuria. "Analyzing driver behavior under naturalistic driving conditions: A review". In: *Accident Analysis & Prevention* 150 (2021), p. 105908.

[33] Gys Albertus Marthinus Meiring and Hermanus Carel Myburgh. "A review of intelligent driving style analysis systems and related artificial intelligence algorithms". In: *Sensors* 15.12 (2015), pp. 30653–30682.

[34] Ishita Seth. "A survey on driver behavior detection techniques". In: *Int. J. Sci. Res. Sci. Technol* 7 (2020), pp. 401–404.

[35] Anaıos Halin, Jacques G Verly, and Marc Van Droogenbroeck. "Survey and synthesis of state of the art in driver monitoring". In: *Sensors* 21.16 (2021), p. 5558.

[36] Lien-Wu Chen and Hsien-Min Chen. "Driver behavior monitoring and warning with dangerous driving detection based on the internet of vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 22.11 (2020), pp. 7232–7241.

[37] Yuichi Saito, Makoto Itoh, and Toshiyuki Inagaki. "Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents". In: *IEEE Transactions on Human-Machine Systems* 46.5 (2016), pp. 660–671.

[38] Wanghua Deng and Ruoxue Wu. "Real-time driver-drowsiness detection system using facial features". In: *Ieee Access* 7 (2019), pp. 118727–118738.

[39] Shabnam Abtahi et al. "A yawning measurement method using embedded smart cameras". In: *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE. 2013, pp. 1605–1608.

[40] Heejin Jeong and Yili Liu. "Effects of non-driving-related-task modality and road geometry on eye movements, lane-keeping performance, and workload while driving". In: *Transportation research part F: traffic psychology and behaviour* 60 (2019), pp. 157–171.

[41] Nitin Kumar Gupta et al. "Review paper on yawning detection prediction system for driver drowsiness". In: *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE. 2021, pp. 1–6.

[42] Jiadi Yu et al. "Fine-grained abnormal driving behaviors detection and identification with smartphones". In: *IEEE transactions on mobile computing* 16.8 (2016), pp. 2198–2212.

[43] Kwok Tai Chui, Miltiadis D Lytras, and Ryan Wen Liu. “A generic design of driver drowsiness and stress recognition using MOGA optimized deep MKL-SVM”. In: *Sensors* 20.5 (2020), p. 1474.

[44] Angelica Reyes-Muñoz et al. “Integration of body sensor networks and vehicular ad-hoc networks for traffic safety”. In: *Sensors* 16.1 (2016), p. 107.

[45] Koichi Fujiwara et al. “Heart rate variability-based driver drowsiness detection and its validation with EEG”. In: *IEEE transactions on biomedical engineering* 66.6 (2018), pp. 1769–1778.

[46] Yuhong Xu et al. “Real-time monitoring system of automobile driver status and intelligent fatigue warning based on triboelectric nanogenerator”. In: *ACS nano* 15.4 (2021), pp. 7271–7278.

[47] Amirmasoud Ahmadi, Hanieh Bazregarzadeh, and Kamran Kazemi. “Automated detection of driver fatigue from electroencephalography through wavelet-based connectivity”. In: *Biocybernetics and Biomedical Engineering* 41.1 (2021), pp. 316–332.

[48] Atif Alamri et al. “An effective bio-signal-based driver behavior monitoring system using a generalized deep learning approach”. In: *IEEE Access* 8 (2020), pp. 135037–135049.

[49] Zahid Halim and Mahma Rehan. “On identification of driving-induced stress using electroencephalogram signals: A framework based on wearable safety-critical scheme and machine learning”. In: *Information Fusion* 53 (2020), pp. 66–79.

[50] Haoning Xi, Yi Zhang, and Yi Zhang. “Detection of Safety Features of Drivers Based on Image Processing”. In: *18th COTA International Conference of Transportation Professionals*. American Society of Civil Engineers Reston, VA. 2018, pp. 2098–2109.

[51] Manuela Witt et al. “Driver profiling–data-based identification of driver behavior dimensions and affecting driver characteristics for multi-agent traffic simulation”. In: *Transportation research part F: traffic psychology and behaviour* 64 (2019), pp. 361–376.

[52] Alexey Kashevnik, Igor Lashkov, and Andrei Gurtov. “Methodology and mobile application for driver behavior analysis and accident prevention”. In: *IEEE transactions on intelligent transportation systems* 21.6 (2019), pp. 2427–2436.

[53] Christopher Streiffer et al. "Darnet: A deep learning solution for distracted driving detection". In: *Proceedings of the 18th acm/ifip/usenix middleware conference: Industrial track*. 2017, pp. 22–28.

[54] Faiqa Sajid et al. "An efficient deep learning framework for distracted driver detection". In: *IEEE Access* 9 (2021), pp. 169270–169280.

[55] Jafar A Alzubi et al. "Distracted driver detection using compressed energy efficient convolutional neural network". In: *Journal of Intelligent & Fuzzy Systems* 42.2 (2022), pp. 1253–1265.

[56] HaiLong Liu et al. "Visualization of driving behavior based on hidden feature extraction by using deep learning". In: *IEEE Transactions on Intelligent Transportation Systems* 18.9 (2017), pp. 2477–2489.

[57] Kwan Woo Lee et al. "Convolutional neural network-based classification of driver's emotion during aggressive and smooth driving using multi-modal camera sensors". In: *Sensors* 18.4 (2018), p. 957.

[58] Changxi You, Jianbo Lu, and Panagiotis Tsiotras. "Nonlinear driver parameter estimation and driver steering behavior analysis for ADAS using field test data". In: *IEEE Transactions on Human-Machine Systems* 47.5 (2017), pp. 686–699.

[59] Dajun Wang et al. "Risky driver recognition based on vehicle speed time series". In: *IEEE Transactions on Human-Machine Systems* 48.1 (2017), pp. 63–71.

[60] Juan Carmona et al. "Embedded system for driver behavior analysis based on GMM". In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2016, pp. 61–65.

[61] Scott Schnelle et al. "A personalizable driver steering model capable of predicting driver behaviors in vehicle collision avoidance maneuvers". In: *IEEE Transactions on Human-Machine Systems* 47.5 (2016), pp. 625–635.

[62] Juan Carmona et al. "Data fusion for driver behaviour analysis". In: *Sensors* 15.10 (2015), pp. 25968–25991.

[63] Deepak Kumar Dewangan and Satya Prakash Sahu. "Driving behavior analysis of intelligent vehicle system for lane detection using vision-sensor". In: *IEEE Sensors Journal* 21.5 (2020), pp. 6367–6375.

[64] Wenshuo Wang et al. "Driving style classification using a semisupervised support vector machine". In: *IEEE Transactions on Human-Machine Systems* 47.5 (2017), pp. 650–660.

[65] Yang Zheng and John HL Hansen. "Lane-change detection from steering signal using spectral segmentation and learning-based classification". In: *IEEE Transactions on Intelligent Vehicles* 2.1 (2017), pp. 14–24.

[66] Zhijun Chen et al. "Dangerous driving behavior detection using video-extracted vehicle trajectory histograms". In: *Journal of Intelligent Transportation Systems* 21.5 (2017), pp. 409–421.

[67] Alexey Kashevnik et al. "Driver distraction detection methods: A literature review and framework". In: *IEEE Access* 9 (2021), pp. 60063–60076.

[68] Charlotte Jacobé de Naurois et al. "Detection and prediction of driver drowsiness using artificial neural network models". In: *Accident Analysis & Prevention* 126 (2019), pp. 95–104.

[69] Alexandros Nikitas et al. "Cycling in the era of COVID-19: Lessons learnt and best practice policy recommendations for a more bike-centric future". In: *Sustainability* 13.9 (2021), p. 4620.

[70] Glen Fuller et al. "The reactivated bike: Self-reported cycling activity during the 2020 COVID-19 pandemic in Australia". In: *Transportation research interdisciplinary perspectives* 10 (2021), p. 100377.

[71] Robel Feleke et al. "Comparative fatality risk for different travel modes by age, sex, and deprivation". In: *Journal of Transport & Health* 8 (2018), pp. 307–320.

[72] Nikiforos Stamatiadis, Salvatore Cafiso, and Giuseppina Pappalardo. "A comparison of bicyclist attitudes in two urban areas in USA and Italy". In: *Data Analytics: Paving the Way to Sustainable Urban Mobility: Proceedings of 4th Conference on Sustainable Urban Mobility (CSUM2018), 24-25 May, Skiathos Island, Greece*. Springer. 2019, pp. 272–279.

[73] Mehdi Hosseinpour et al. "An in-depth analysis of self-reported cycling injuries in single and multiparty bicycle crashes in Denmark". In: *Journal of safety research* 77 (2021), pp. 114–124.

[74] Evelien Polders and Tom Brijs. *How to analyse accident causation? A handbook with focus on vulnarable road users*. Hasselt University, 2018.

[75] Grégory Vandenbulcke, Luc Int Panis, and Isabelle Thomas. "On the location of reported and unreported cycling accidents: A spatial network analysis for Brussels". In: *Cybergeo: European Journal of Geography* (2017).

[76] Nicholas N Ferenchak and Wesley E Marshall. "Advancing healthy cities through safer cycling: An examination of shared lane markings". In: *International journal of transportation science and technology* 8.2 (2019), pp. 136–145.

[77] David Shinar et al. "Under-reporting bicycle accidents to police in the COST TU1101 international survey: Cross-country comparisons and associated factors". In: *Accident Analysis & Prevention* 110 (2018), pp. 177–186.

[78] Ashutosh Arun et al. "A systematic mapping review of surrogate safety assessment using traffic conflict techniques". In: *Accident Analysis & Prevention* 153 (2021), p. 106016.

[79] Attila Borsos. "Application of Bivariate Extreme Value models to describe the joint behavior of temporal and speed related surrogate measures of safety". In: *Accident Analysis & Prevention* 159 (2021), p. 106274.

[80] Marco Dozza and Andre Fernandez. "Understanding bicycle dynamics and cyclist behavior from naturalistic field data (November 2012)". In: *IEEE Transactions on intelligent transportation systems* 15.1 (2013), pp. 376–384.

[81] Chunyong Yin et al. "Anomaly detection based on convolutional recurrent autoencoder for IoT time series". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.1 (2020), pp. 112–122.

[82] Ramin Arvin, Asad J Khattak, and Hairong Qi. "Safety critical event prediction through unified analysis of driver and vehicle volatilities: Application of deep learning methods". In: *Accident Analysis & Prevention* 151 (2021), p. 105949.

[83] Katja Schleinitz et al. "Conflict partners and infrastructure use in safety critical events in cycling–Results from a naturalistic cycling study". In: *Transportation research part F: traffic psychology and behaviour* 31 (2015), pp. 99–111.

[84] Christine Chaloupka-Risser and Elisabeth Füssl. "The importance of communication between cyclists and other traffic participants and its potential in reducing traffic safety-critical events". In: *Transactions on Transport Sciences* 8.1 (2017), pp. 24–30.

[85] Julia Werneke, Marco Dozza, and MariAnne Karlsson. "Safety–critical events in everyday cycling–Interviews with bicyclists and video annotation of safety–critical events in a naturalistic cycling study". In: *Transportation Research Part F: Traffic Psychology and Behaviour* 35 (2015), pp. 199–212.

[86] Alejandra Sofía Fonseca-Cabrera et al. "Micromobility users' behaviour and perceived risk during meeting manoeuvres". In: *International journal of environmental research and public health* 18.23 (2021), p. 12465.

[87] Marco Dozza and Julia Werneke. "Introducing naturalistic cycling data: What factors influence bicyclists' safety in the real world?" In: *Transportation research part F: traffic psychology and behaviour* 24 (2014), pp. 83–91.

[88] Emanuele Murgano et al. "A novel acceleration signal processing procedure for cycling safety assessment". In: *Sensors* 21.12 (2021), p. 4183.

[89] Salvatore Cafiso et al. "Urban road pavements monitoring and assessment using bike and e-scooter as probe vehicles". In: *Case Studies in Construction Materials* 16 (2022), e00889.

[90] Kushal Mehta, Babak Mehran, and Bruce Hellinga. "A methodology to estimate the number of unsafe vehicle-cyclist passing events on urban arterials". In: *Accident Analysis & Prevention* 124 (2019), pp. 92–103.

[91] Stefan Candefjord et al. "Using smartphones to monitor cycling and automatically detect accidents: Towards ecall functionality for cyclists". In: *International cycling safety conference 2014, 18-19 november 2014, göteborg, sweden.* 2014.

[92] Salvatore Cafiso, Giuseppina Pappalardo, and Nikiforos Stamatiadis. "Observed risk and user perception of road infrastructure safety assessment for cycling mobility". In: *Infrastructures* 6.11 (2021), p. 154.

[93] Asma Belhadi et al. "Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges". In: *ACM Transactions on Management Information Systems (TMIS)* 11.3 (2020), pp. 1–29.

[94] Qingfan Wang et al. "A data-driven, kinematic feature-based, near real-time algorithm for injury severity prediction of vehicle occupants". In: *Accident Analysis & Prevention* 156 (2021), p. 106149.

[95] Sepehr Sabour, Sanjeev Rao, and Majid Ghaderi. "Deepflow: Abnormal traffic flow detection using siamese networks". In: *2021 IEEE International Smart Cities Conference (ISC2)*. IEEE. 2021, pp. 1–7.

[96] Nicolette Formosa et al. "Predicting real-time traffic conflicts using deep learning". In: *Accident Analysis & Prevention* 136 (2020), p. 105429.

[97] Tingting Huang, Shuo Wang, and Anuj Sharma. "Highway crash detection and risk estimation using deep learning". In: *Accident Analysis & Prevention* 135 (2020), p. 105392.

[98] Zubayer Islam et al. "Crash data augmentation using variational autoencoder". In: *Accident Analysis & Prevention* 151 (2021), p. 105950.

[99] Abdelhamid Mammeri et al. "Vulnerable road users detection based on convolutional neural networks". In: *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE. 2020, pp. 1–6.

[100] Weishan Dong et al. "Characterizing driving styles with deep learning". In: *arXiv preprint arXiv:1607.03611* (2016).

[101] Arianna Bichicchi et al. "Analysis of road-user interaction by extraction of driver behavior features using deep learning". In: *IEEE Access* 8 (2020), pp. 19638–19645.

[102] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. "LLNet: A deep autoencoder approach to natural low-light image enhancement". In: *Pattern Recognition* 61 (2017), pp. 650–662.

[103] Yali Peng et al. "CNN-GRU-AM for shared bicycles demand forecasting". In: *Computational intelligence and neuroscience* 2021 (2021).

[104] Shaojie Qiao et al. "A dynamic convolutional neural network based shared-bike demand forecasting model". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.6 (2021), pp. 1–24.

[105] Matej Babič et al. "A new composite method of modeling bicycle traffic using convolutional neural networks and genetic programming". In: *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*. IEEE. 2021, pp. 1–6.

[106] Sandra Detzer et al. "Analysis of traffic safety for cyclists: The automatic detection of critical traffic situations for cyclists". In: *WIT Transactions on The Built Environment* 138 (2014).

[107] Rushdi Alsaleh and Tarek Sayed. "Markov-game modeling of cyclist-pedestrian interactions in shared spaces: A multi-agent adversarial inverse reinforcement learning approach". In: *Transportation research part C: emerging technologies* 128 (2021), p. 103191.

[108] Keneth Morgan Kwayu et al. "Automatic topics extraction from crowd-sourced cyclists near-miss and collision reports using text mining and Artificial Neural Networks". In: *International Journal of Transportation Science and Technology* 11.4 (2022), pp. 767–779.

[109] Houxiang Zhou et al. "Assessment of the performance of carrier-phase and Doppler smoothing code for low-cost GNSS receiver positioning". In: *Results in Physics* 19 (2020), p. 103574.

[110] Hadi Kordestani and Chunwei Zhang. "Direct use of the savitzky–golay filter to develop an output-only trend line-based damage detection method". In: *Sensors* 20.7 (2020), p. 1983.

[111] Ruyin Cao et al. "A simple method to improve the quality of NDVI time-series data by integrating spatiotemporal information with the Savitzky-Golay filter". In: *Remote Sensing of Environment* 217 (2018), pp. 244–257.

[112] Pei Li, Mohamed Abdel-Aty, and Jinghui Yuan. "Using bus critical driving events as surrogate safety measures for pedestrian and bicycle crashes based on GPS trajectory data". In: *Accident Analysis & Prevention* 150 (2021), p. 105924.

[113] William Grant Hatcher and Wei Yu. "A survey of deep learning: Platforms, applications and emerging research trends". In: *IEEE Access* 6 (2018), pp. 24411–24432.

[114] Gavneet Singh Chadha et al. "Deep convolutional clustering-based time series anomaly detection". In: *Sensors* 21.16 (2021), p. 5488.

[115] Zewen Li et al. "A survey of convolutional neural networks: analysis, applications, and prospects". In: *IEEE transactions on neural networks and learning systems* (2021).

[116] Xifeng Guo et al. "Deep clustering with convolutional autoencoders". In: *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part II 24*. Springer. 2017, pp. 373–382.

[117] César Ferri, José Hernández-Orallo, and R Modroiu. "An experimental comparison of performance measures for classification". In: *Pattern recognition letters* 30.1 (2009), pp. 27–38.

[118] Jillian Strauss et al. "Cyclist deceleration rate as surrogate safety measure in Montreal using smartphone GPS data". In: *Accident Analysis & Prevention* 99 (2017), pp. 287–296.

[119] Sergio A Useche et al. "Healthy but risky: A descriptive study on cyclists' encouraging and discouraging factors for using bicycles, habits and safety outcomes". In: *Transportation research part F: traffic psychology and behaviour* 62 (2019), pp. 587–598.

[120] Georgios Kapousizis et al. "A review of state-of-the-art bicycle technologies affecting cycling safety: Level of smartness and technology readiness". In: *Transport Reviews* (2022), pp. 1–23.

[121] Amirhossein Nourbakhshrezaei, Mojgan Jadidi, and Gunho Sohn. "Improving Cyclists' Safety Using Intelligent Situational Awareness System". In: *Sustainability* 15.4 (2023), p. 2866.

[122] Freya Slootmans. "European Road Safety Observatory Facts and Figures – Cyclists - 2021". In: *European Road Safety* (2021), p. 26.

[123] Antonella Scarano et al. "Systematic literature review of 10 years of cyclist safety research". In: *Accident Analysis & Prevention* 184 (2023), p. 106996.

[124] Nikiforos Stamatiadis, Salvatore Cafiso, and Giuseppina Pappalardo. "A comparison of bicyclist attitudes in two urban areas in USA and Italy". In: *Conference on Sustainable Urban Mobility*. Springer. 2018, pp. 272–279.

[125] Miao Cai et al. "The association between crashes and safety-critical events: Synthesized evidence from crash reports and naturalistic driving data among commercial truck drivers". In: *Transportation research part C: emerging technologies* 126 (2021), p. 103016.

[126] Shumayla Yaqoob et al. "Detection of anomalies in cycling behavior with convolutional neural network and deep learning". In: *European Transport Research Review* 15 (2023).

[127] April Gadsby and Kari Watkins. "Instrumented bikes and their use in studies on transportation behaviour, safety, and maintenance". In: *Transport reviews* 40.6 (2020), pp. 774–795.

[128] Mohamed R Ibrahim et al. “Cycling near misses: a review of the current methods, challenges and the potential of an AI-embedded system”. In: *Transport reviews* 41.3 (2021), pp. 304–328.

[129] Alireza Darzian Rostami et al. “Predicting critical bicycle-vehicle conflicts at signalized intersections”. In: *Journal of advanced transportation* 2020 (2020), pp. 1–16.

[130] Rushdi Alsaleh and Tarek Sayed. “Microscopic modeling of cyclists interactions with pedestrians in shared spaces: a Gaussian process inverse reinforcement learning approach”. In: *Transportmetrica A: transport science* 18.3 (2022), pp. 828–854.

[131] Yuanli Gu et al. “A deep learning framework for cycling maneuvers classification”. In: *IEEE Access* 7 (2019), pp. 28799–28809.

[132] Ahmet-Serdar Karakaya et al. “CycleSense: Detecting near miss incidents in bicycle traffic from mobile motion sensors”. In: *Pervasive and Mobile Computing* 91 (2023), p. 101779.

[133] Yang Chen et al. “A practical approach to reconstruct high-quality Landsat NDVI time-series data by gap filling and the Savitzky–Golay filter”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 180 (2021), pp. 174–190.

[134] Xue Yang et al. “Enhanced Spatial–Temporal Savitzky–Golay Method for Reconstructing High-Quality NDVI Time Series: Reduced Sensitivity to Quality Flags and Improved Computational Efficiency”. In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–17.

[135] Salvatore Cafiso and Carmelo D’Agostino. “Reliability-based assessment of Benefits in roadway safety management”. In: *Transportation research record* 2513.1 (2015), pp. 1–10.

[136] Weibo Liu et al. “A survey of deep neural network architectures and their applications”. In: *Neurocomputing* 234 (2017), pp. 11–26.

[137] Serkan Kiranyaz et al. “1D convolutional neural networks and applications: A survey”. In: *Mechanical systems and signal processing* 151 (2021), p. 107398.

[138] Guangle Yao, Tao Lei, and Jiandan Zhong. “A review of convolutional-neural-network-based action recognition”. In: *Pattern Recognition Letters* 118 (2019), pp. 14–22.

[139] Jianxin Wu. "Introduction to convolutional neural networks". In: *National Key Lab for Novel Software Technology. Nanjing University. China* 5.23 (2017), p. 495.

[140] Mohsin Munir et al. "DeepAnT: A deep learning approach for unsupervised anomaly detection in time series". In: *Ieee Access* 7 (2018), pp. 1991–2005.

[141] Yifei Zhang. "A better autoencoder for image: Convolutional autoencoder". In: *ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/-Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_ 58. pdf (accessed on 23 March 2017)*. 2018.

[142] Jiayang Xu and Karthik Duraisamy. "Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics". In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020), p. 113379.

[143] Zhihong Zhang et al. "Depth-based subgraph convolutional auto-encoder for network representation learning". In: *Pattern Recognition* 90 (2019), pp. 363–376.

[144] Jianmin Wang, Yan Wang, et al. "Analysis of 5G Smart Communication Base Station Doppler-Smoothed Pseudorange Single-Point Geodesic Positioning Accuracy". In: *Journal of Computer Networks and Communications* 2023 (2023).

[145] Shumayla Yaqoob et al. "Detection of anomalies in cycling behavior with convolutional neural network and deep learning". In: *European Transport Research Review* 15.1 (2023), p. 9.

[146] Jie Hu, Xiaoqin Zhang, and Stephen Maybank. "Abnormal driving detection with normalized driving behavior data: a deep learning approach". In: *IEEE transactions on vehicular technology* 69.7 (2020), pp. 6943–6951.

[147] Zengyu Ding et al. "Comparison of estimating missing values in iot time series data using different interpolation algorithms". In: *International Journal of Parallel Programming* 48 (2020), pp. 534–548.

[148] Satya Narayan Shukla and Benjamin M Marlin. "Interpolation-prediction networks for irregularly sampled time series". In: *arXiv preprint arXiv:1909.07782* (2019).

[149] Davide Morelli et al. "Analysis of the impact of interpolation methods of missing RR-intervals caused by motion artifacts on HRV features estimations". In: *Sensors* 19.14 (2019), p. 3163.

[150] Christopher W Seymour et al. "Derivation, validation, and potential treatment implications of novel clinical phenotypes for sepsis". In: *Jama* 321.20 (2019), pp. 2003–2017.

[151] Hwanjun Song et al. "Learning from noisy labels with deep neural networks: A survey". In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[152] Yuji Roh, Geon Heo, and Steven Euijong Whang. "A survey on data collection for machine learning: a big data-ai integration perspective". In: *IEEE Transactions on Knowledge and Data Engineering* 33.4 (2019), pp. 1328–1347.

[153] Ihab F Ilyas and Xu Chu. *Data cleaning.* Morgan & Claypool, 2019.

[154] Malek Karaim, Aboelmagd Noureldin, and Tashfeen B Karamat. "Low-cost IMU data denoising using Savitzky-Golay filters". In: *2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. IEEE. 2019, pp. 1–5.

[155] Jianye Zhang and Peng Yin. "Multivariate time series missing data imputation using recurrent denoising autoencoder". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2019, pp. 760–764.

[156] Hsin-Tien Chiang et al. "Noise reduction in ECG signals using fully convolutional denoising autoencoders". In: *Ieee Access* 7 (2019), pp. 60806–60813.

[157] Michael Greenacre et al. "Principal component analysis". In: *Nature Reviews Methods Primers* 2.1 (2022), p. 100.

[158] Srikanth Thudumu et al. "A comprehensive survey of anomaly detection techniques for high dimensional big data". In: *Journal of Big Data* 7 (2020), pp. 1–30.

[159] Stephan Frisbie and Mohamed Younis. "Data-Driven Wireless Anomaly Detection Using Spectral Features". In: *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE. 2022, pp. 711–716.

[160] Tailai Wen and Roy Keyes. "Time series anomaly detection using convolutional neural networks and transfer learning". In: *arXiv preprint arXiv:1905.13628* (2019).

[161] Narges Ehsani, Farrokh Aminifar, and Hamed Mohsenian-Rad. "Convolutional autoencoder anomaly detection and classification based on distribution PMU measurements". In: *IET Generation, Transmission & Distribution* 16.14 (2022), pp. 2816–2828.