AI-assisted Anomalous Event Detection for Connected Vehicles

by

Nima Taherifard

Thesis Supervisor : Dr. Burak Kantarci

A thesis
presented to the University of Ottawa
in fulfillment of the
thesis requirement for the degree of
Master of Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Connected vehicle networks and future autonomous driving systems call for characterization of risky driving events to improve safety applications and autonomous driving features. Precision of driving event characterization (DEC) systems in connected vehicles has become increasingly important with the responsive connectivity that is available to the modern vehicles. While risky behavior patterns entail potential safety issues on road networks, the advent of vehicular sensing and vehicular networks cannot guarantee accurate characterization of driving/movement behavior of vehicles and the precision of such systems still remains an open issue. Additionally, artificial intelligence-backed solutions are vital components towards highly accurate characterization systems in the modern transportation. However, such solutions require significant volume of driving event data for an acceptable level of performance. With this in mind, the proposal of this thesis is three-fold: 1) a reliable methodology to generate representative data under the scarcity of diverse anomalous sensory data, 2) classification of mobility/driving events of vehicles with attention-based deep learning methods, and 3) a modular prior-knowledge input method to the characterization methodologies in order to further improve the trustworthiness of the systems. Implementing the proposed steps, we are able to not only increase the consistency in the training process but also reduce the false positive detection instances compared to the previous models.

One of the roadblocks against robust event characterization systems in connected vehicles that is tackled in this thesis is the scarcity of anomalous driving data to make the training of event classification models robust. To mitigate this issue an optimized deep recurrent neural network-based encoding model is introduced to extract the precise feature representation of the anomalous data. The utilization of the encoded input to the previous network allowed for a 12% accuracy improvement. Furthermore, we introduced a framework for precise risky driving behavior detection that takes advantage of an attention-based neural networks model. Ultimately, the combination of prior knowledge modelling with our network and some optimizations to the network structure, the model outperforms the state-of-the-art solutions by reaching an average accuracy of 0.96 and F1-score of 0.92.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Burak Kantarci for his continuous support and guidance. He opened the door for me to a novel research field, helped me to find a suitable research topic, and guided me to successfully complete my work. Without his guidance, expertise, and attention to detail, I would not have been able to overcome the challenges faced in the work we have done and published the findings. The honest, careful, and hard working spirit of Dr. Burak Kantarci will always inspire me.

I would like to thank Dr. Murat Simsek, whom we have worked together throughout the project. His knowledge and assistance were invaluable and we had a productive collaboration. I would like to thank all the NEXTCON lab members for their help and support that made this thesis possible.

I would thank Raven Connected for supporting this project and providing the opportunity and resources which facilitated the thesis.

Finally, I must express my profound gratitude to my parents for their support in every step of my life and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

This thesis is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under RGPIN/2017-04032 and Ontario Centres of Excellence (OCE) TalentEdge Internship Project 32815 in collaboration with Raven Connected.

Table of Contents

List of Tables

List of Figures

Abbreviations

5G 5th Generation of Communication 1, 2, 4, 11, 42

AA Aggressive Acceleration 1, 30, 33, 37, 44

AE Auto-encoders 1, 23, 24, 26, 69

AI Artificial Intelligence 1, 4, 16, 24, 31, 73

BiLSTM Bidirectional Long Short-term Memory 1, 55

CNN Convolutional Neural Network 1, 16, 21

ConvLSTM Convolutional Long Short-term Memory 1, 65, 67–69, 72

DEC Driving Event Characterization iii, 1, 4–6, 8–10, 12–15, 17, 19–24, 26, 27, 32, 42, 44, 72–74

DIIS Deceleration Intention Inference System 1, 25

DL Deep Learning 1, 16, 17, 22, 31

FN False Negative 1

FP False Positive 1

GAN Generative Adversarial Network 1, 26, 73

GDP Gross Domestic Product 1, 3

GPRS General Packet Radio Service 1

GPS Global Positioning System 1, 15, 18, 19

GRU Gated Recurrent Unit 1

HB Harsh Braking 1, 30, 33, 37, 44

HL Harsh Left Turn 1, 30, 33, 37, 44

IEEE Institute of Electrical and Electronics Engineers 1, 19

IMU Inertial Measurement Unit 1, 18, 30, 31

IoT Internet of Things 1

IoV Internet of Vehicles 1

ISO International Organization for Standardization 1, 14

ITS Intelligent Transportation System 1–4, 9, 11, 12, 16, 42

LSTM Long Short Term Memory 1, 8, 25, 30, 31, 34, 35, 37, 40, 42, 48, 50, 51, 54, 55, 58, 64–66, 71

MEC Mobile Edge Computing 1, 2

MIMO Multiple-input Multiple-output 1, 2

ML Machine Learning 1

MLP Multi-layer Perception 1, 36, 38, 65, 67–69, 72

MSE Mean Square Error 1, 48, 49, 53

NHTSA National Highway Traffic Safety Administration 1, 3

NN Neural Network 1, 17, 21

OBD On-board diagnostics 1, 32, 44

PCA Principal Component Analysis 1, 21, 36, 38, 55

PKI Prior Knowledge Input Modelling 1, 9, 27, 62, 65–69, 72

PKID Prior Knowledge Input with Difference 1, 27

PLR License Plate Recognition 1, 31

QR QR code 1

RD regular Driving 1, 30, 33, 37, 44

ReLU Rectified Linear Units 1, 50

RF Random Forest 1, 17

RL Harsh Right Turn 1, 30, 33, 37, 44

RNN Recurrent neural network 1, 16, 23–25

SAE Society of Automotive Engineers 1

SDN Software Defined Networking 1, 2

SVM Support Vector Machine 1, 17, 21

TN True Negative 1

TP True Positive 1

V2I Vehicle-to-infrastructure Communication 1, 2

V2V Vehicle-to-vehicle Communication 1, 2

V2X Vehicle-to-everything 1, 4, 11

VAE Variational Auto-encoder 1, 21–23

VGG Visual Geometry Group 1, 60

VQA Visual Question Answering 1, 16

WHO World Health Organization 1, 3

List of Symbols

C Number of classes 1, 36

E Event raw signal 1, 46

L Number of LSTM hidden layers 1, 35

$\mathbf{W}$ Windowed Signals 1, 46

a Axes dimension of the signal 1, 36, 46, 49

β_1 The exponential decay rate for the first moment estimates in Adam optimizer 1, 38, 53, 67

β_2 The exponential decay rate for the second-moment estimates in Adam optimizer 1, 38, 53, 67

b_i Bias matrix of hidden layer i 1, 51

d Bottleneck dimension 1, 48, 49

e_{PKI} Error value of PKI model 1

ϵ Epsilon, a very small number to prevent any division by zero in Adam optimizer 1, 38, 53, 67

h_d Hidden states of the decoder 1

h_e Hidden states of the encoder 1, 36

Hz Unit of frequency in the International System of Units (Hertz) 1, 37

i Signal dimension 1, 36, 49

j Signal length 1, 36, 49

n Number of signal windows 1, 44, 46, 53, 66

P_c Prediction of the classifier network 1

p Predicted probability 1, 51

r_n Temporal encoded vector of input window 1, 51

S_n Spatial encoded vector of input window 1, 50

$tanh$ Hyperbolic Tangent 1, 66

t Duration of the event signal 1, 36, 46, 49

v_i Attention vector 1, 51

w_i Weight matrix of hidden layer i 1, 51

$\hat{x}$ Synthetic signal features 1

x Input window size 1, 46

y Input labels 1

Publications of the Candidate During MCS Studies

- **N. Taherifard**, M. Simsek, C. Lascelles, and B. Kantarci, ”Attention-Based Event Characterization for Scarce Vehicular Sensing Data”, in IEEE Open Journal of Vehicular Technology, vol. 1, pp. 317-330, September 2020
- **N. Taherifard**, M. Simsek, C. Lascelles, and B. Kantarci, ”Prior Knowledge Input to Improve LSTM Auto-encoder-based Characterization of Vehicular Sensing Data”, in IEEE International Conference on Communications (ICC), Montreal, QC, Canada, June 2021 (Accepted)
- **N. Taherifard**, M. Simsek, C. Lascelles, and B. Kantarci, ”Machine learning-driven event characterization under scarce vehicular sensing data”, in IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Pisa, Italy (Virtual Event), September 2020
- **N. Taherifard**, M. Simsek, and B. Kantarci, ”Bridging Connected Vehicles with Artificial Intelligence for Smart First Responder Services”, in IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Cabada, November 2019

Chapter 1

Introduction

It has been forecasted to see millions of autonomous and semi-autonomous vehicles on the roads by 2025. This vision is not possible without reaching level-5 vehicle autonomy defined by The Society of Automotive Engineers (SAE). Brief explanation of SAE autonomy levels is provided in 1.1. However, the state-of-the-art autonomous driving systems are still below human confidence levels in driving which prohibits the acceptance of such systems among the public and SAE. Moreover, the success rate of machine learning algorithms heavily depends on the quality of the input features they are given. Autonomous algorithms benefit from more accurate input data. If the algorithms are fed with irrelevant information or inadequate data, they may fail to make the desired decisions that are needed for level-5 driving autonomy [1].

This thesis explores the systems that accurately detect driving behaviors and argues that the driving patterns contain invaluable information to boost autonomous driving performance. Detecting the driving events and feeding the information of the driving behaviors to the autonomous systems provide the systems with knowledge of abnormal situations that is beneficial in decision making by the vehicles [2].

1.1 Motivation

Vehicular communication networks are gaining commercial and practical relevance as a result of recent developments in connection technologies [3]. With recent research and ad-

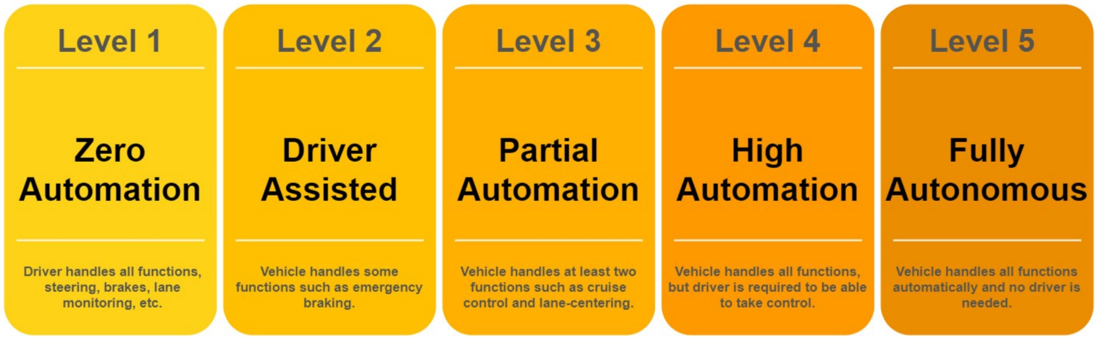

Figure 1.1: Five levels of the driving autonomy.

vancements in multiple-input and multiple-output (MIMO) technologies, significant spectrum efficiency and energy efficiency are achieved in 5G wireless communications [4]. Moreover, the emergence of autonomous vehicles has increased the need to have close to instant information transmission time. The low transmission latency requirement of less than 1 ms [5] has been met with the integration of 5G mobile communication technologies, cloud computing, mobile edge computing (MEC), and software-defined networking (SDN) [6] which paves the way for level-5 autonomy.

Vehicular sensing network is a consequential phenomenon that is emerging following the improved Vehicle-to-vehicle (V2V) and Vehicle-to-infrastructure (V2I) communications by 5G [7]. Information communication in vehicular sensing was absent in the previous generations of communication due to the unsatisfied requirements of the vehicular safety applications [8]. In an Intelligent Transportation System (ITS) environment, vehicles, as network nodes, construct a super-large-scale sensing network. Moreover, the vehicles are heavily equipped with a variety of sensors such as speedometers, airflow, and oxygen sensors, localization sensors and can collect massive information. The communication means are now capable of handling the big-data generated in the vehicular sensors which call for intelligent solutions to utilize the massive information [9].

Additionally, the popularity of mobile devices which are heavily equipped with onboard sensing enables the mobile platforms to be employed as participatory vehicular sensing units [10, 11, 11]. Vehicular sensing networks can also leverage the crowd-sourced information provided by mobile devices for robust and large-scale sensing. Using the information provided by the passengers of the vehicles, numerous data can be mined for traffic forecasting, traffic engineering, and vehicle safety systems in an ITS platform. Vehicles, on the other hand, offer ongoing power generation and therefore can be equipped with more powerful computing and transmission equipment which is crucial for safety applications.

Generally, vehicular sensing platforms are responsible to record, analyze, and transmit appropriate data in the vehicular network. Vehicles are data collecting nodes that provide ongoing information of various sorts. The recorded data can be collected utilizing a wide range of onboard sensing devices (e.g., accelerometer, gyroscope, camera, etc.), which are analyzed for information and patterns (e.g., traffic flow, license plate, driving behaviors, etc) [12].

One of the main issues faced by modern transportation systems is the collisions and the consequences of the collisions on the road. With the advent of connected vehicles and autonomous driving systems, road safety applications have become a crucial requirement [13] for such systems. Facts and reports show that risky driving behavior is the cause of about 25% of the road accidents [14]. In 2018, National Highway Traffic Safety Administration (NHTSA) reported over 2800 claimed lives caused by risky driving behaviors and distracted driving only in the United States. The latest annual report by the World Health Organization (WHO) demonstrates over 1.35 million deaths as a result of road accidents globally and between 20 to 50 million serious injuries every year. Moreover, road crashes are wasting around 3% of Gross Domestic Product (GDP) of the countries and becoming one of the leading causes of death in adults under the age of 29. Thus, there is an essential need for research and advancement to lower these numbers which can contribute to the pace of growth in developing countries.

Vehicles as the main means of transportation have the potential to affect the occurrence of road accidents and directly contribute to the consequent effects of the accidents. With the advancements in sensors, communication technologies, and processing power, vehicles have gained the capability of extracting contextual information through in-vehicle sensors and communication infrastructures [15]. Therefore, vehicles are equipped with sensing, processing, communication, and storage units that lead to the generation of data in big volumes. Despite the surge of vehicular sensory data as a consequence of more advanced vehicles equipped with highly accurate sensors, there is a lack of available effective data for studies on harmful driving events [16]. Moreover, acquiring such data can be costly and not feasible; hence an alternative model is needed to accurately generate such data for simulation and development purposes.

AI methods are major building blocks of an ITS. The advanced event driving characterization (DEC) approaches leverage visual, inertial, and positional data to detect or predict upcoming events. Moreover, statistical modeling can also be leveraged to output the occurrence probability of the road events. Utilizing statistical models, we can further strengthen the performance of DECs in vehicular settings where uncertainty is involved.

Most commonly used DEC methods rely on static or dynamic threshold-based rules. Setting generic rules cannot be sufficient in a vehicular environment which is highly dynamic in term of the vehicle, terrain, and ambiance characteristics; therefore non-intelligent DEC cannot be widely deployed in safety applications.

Additionally, in the 5G Era, several services can leverage the Vehicle to Vehicle/Infrastructure (V2X) concept for the betterment of smart applications and services [17]. Even though 5G networks are still a work in progress, countries around the world are quickly developing the infrastructure for the 5G services. The role of 5G is unavoidable in vehicular connectivity and communication[18]. Connected vehicles play a crucial role in smart cities since, as mentioned, they are manufactured with multi-sensing capabilities. Vehicular sensing coupled with real-time communication enables broader safety and non-safety applications in smart transportation.

Having a variety of new tools in signal processing, statistical modeling, and computing such as recurrent neural networks and deep auto-encoder networks, the introduction of an intelligent DEC that is capable of learning dynamic driving behaviors is not far from reach. However, the intelligent methods also introduce unique roadblocks which motivated us to put them into further experiments in this thesis.

Motivated by the mentioned challenges and tools in hand, this thesis intends to optimize Vehicular Event Characterization (DEC) through vehicular sensing data. Determining various driving events and driver behaviors have gained the attention of the researchers with the modernization of vehicles [19]. DEC is the core of accident prevention and driver scoring systems. High precision DECs can be used in vehicular safety applications to alert the surrounding drivers using the connectivity provided by intelligent transportation system (ITS) and the most recent generation of telecommunication (5G).

1.2 Objectives

When level-5 driving automation is considered, the autonomous driving models need to surpass the 99.99% accuracy level which is the human level of driving accuracy. While the driving models have reached high accuracy rates that are close to human levels, there still is room for improvement. Moreover, researchers have suggested that deploying autonomous systems that are just 10% safer than the average human driver will substantially decrease the number of life losses instead of deploying the perfect autonomous driving model [20].

The reliable DEC systems are potential candidates as an extra source of input features to the autonomous driving models. The addition of mobility and true surrounding awareness to the driving models would help to both improve the autonomous driving models as well as provide more safety to the vehicles by identifying the risky driving behaviors locally on the vehicles. Furthermore, the local driving behavior recognition by the DECs is the most reliable form of driving pattern recognition since the recognition is being performed locally, by each vehicle, based on the directly sensed inertia of the vehicle. For instance, the driving pattern recognition system mounted directly on the vehicle can instantly capture the rate of change in the physical behavior of the vehicle with no communication delay.

Additionally, sophistication and utilities provided by the DEC systems are also needed for the next generation of vehicles. To facilitate more safety and non-safety capabilities in the connected vehicle environment, the DEC systems are required to be able to not only detect the anomaly in the behavior of the vehicles but also to characterize unique category of such behaviors. This necessary requirement adds complications to the problem of DEC. While the simple DECs could effortlessly be implemented based on anomaly detection, such systems cannot be used to identify distinct driving events. Therefore, the task of driving event characterization is essentially a signal classification problem that has its unique hurdles. The main objectives of this work are set to contribute towards overcoming such issues.

To improve vehicular safety and autonomy applications by driving behavior characterization, highly accurate characterization modeling is crucial. However, the utilization of neural networks seems to be the enabling factor for future vehicular safety and autonomy. Neural networks, require an abundance of accurate data in the learning process in order to achieve the required high standards since neural networks only perform as accurately as the input training data they are given. Thus, to surpass the current safety level, the system input must also become the center of attention by researchers. The data bias and imbalance are among the well-known issues in the deep learning methodologies which are also present in the vehicular setting. Such problems need to be tackled and minimized if the accuracy level of human drivers is to be surpassed by the modeled networks.

The proposed aim for the DEC methodology in the thesis can be seen in Figure 1.2. Our objective is to first address and minimize the issues introduced by the training data in the characterization models and then to adopt new solutions to optimize the characterization phase. The improvement of the characterization phase allows the autonomous models to gain access to local vehicle behaviors and movement patterns. We formulate our end-to-end model such that the raw input signals are captured locally on the vehicles and enhanced utilizing novel neural networks for the characterization models. Therefore, the input data collection is optimized for the training process of the characterization methods which are building blocks of autonomous driving systems. Additionally, multiple characterization schemes are introduced and exhaustively studied for the driving behavior characterization as the second step in the end-to-end solution. This thesis divides the end-to-end solution into three stages; (i) data enhancement solution, (ii) attention-based driving event characterization modeling along with in-depth structure optimization studies, and (iii) the enhancement prior knowledge-based method for the characterization system to optimize the end-to-end results.

As it will be discussed in Chapter 3, the risky driving event signals are scarce and hidden in the massive data of the regular driving sessions which leads to the insufficient training process of the neural network-based DECs. Utilizing a novel feature extraction network along with the generative models, we attempt to reduce the obstacles introduced by the driving data to the DEC models. Furthermore, to reduce the false-positive case detection by the DEC networks which can lead to catastrophic results in vehicular safety applications, a careful exploration and further optimization of the DECs are the objectives of the thesis which are discussed in Chapters 4 and 5, respectively.

1.3 Research Methodology

Throughout the conduct of various experiments in the fulfillment of the objectives, certain methodologies were set and strictly followed in order to hold the solidity of the work. In the data collection stage of the project, not only the data was collected anonymously in

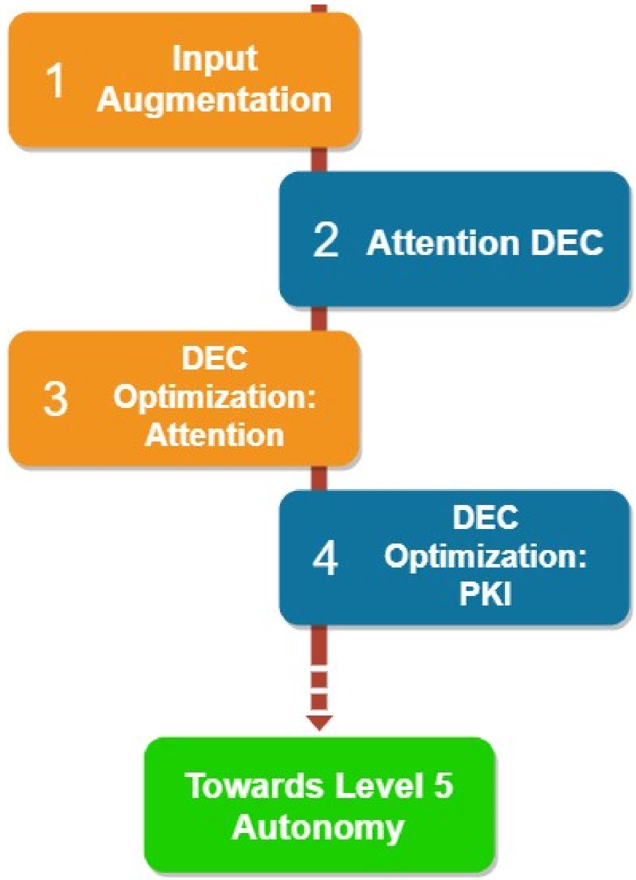

Figure 1.2: Thesis objectives.

order to protect the privacy of the drivers, but the collection was done voluntarily and with no positional information of the drivers. Furthermore, the recorded signals were separated for training and testing purposes and kept identical for different stages of the experiments. The external methods and tools were trained and tested on the same data in order to keep the comparison of the methods as fair as possible.

1.4 Contributions

After a comprehensive literature review during the work, we performed an end-to-end study on a vital problem (i.e. driving event characterization) in vehicular safety and autonomy applications. By forging state-of-the-art neural networks, we reconsider and reinforce the efficiency of the DEC systems for use in modern autonomous systems. We designed and optimized distinct modules to tackle the challenge from both input and characterization aspects. Our contributions in this thesis can be summarized as follows:

1. **Mitigation of data scarcity in vehicular sensing data for accurate training:** The first contribution of the thesis to DEC models is the introduction of a signal augmentation network for driving events and evaluation of the positive impact of the network on the system. We proposed a long short-term memory (LSTM)-based auto-encoder network to boost the vehicular dataset which enables the accuracy of the characterization networks to reach higher levels in detecting risky driving behaviors under limited training data.

2. **Attention-based event characterization model for vehicular sensing:** The second contribution of the thesis is that a novel solution to characterize risky driving behavior is proposed to minimize the false positive detections which are the primary factors for the low reliability of a detection system. An attention-based auto-encoder network is proposed to reconstruct and precisely classify driving event data. Fundamentally, the attention-based neural network performs a self-supervised task to encode behavior characteristics of the input event data as fixed-size vectors. The encoded representation is then used by the decoder network to reconstruct the input signal. The decoder network is capable of outputting more accurate synthetic signals and classify the signals through an attention operation which enables the network to gain the measurement of the importance of the signals and achieve better performance. Furthermore, we extensively experimented with the effect of the network's internal structure on the characterization task to enable the maximum potential of the network performance.

3. **Knowledge-based method for performance boosting:** In order to optimize the proposed characterization network, the effect of knowledge-based modeling is

explored as an additional module to the previously optimized characterization network. The knowledge-based modeling is added to several stages of the characterization networks to study the effectiveness of the system. The introduction of the Prior-knowledge Input (PKI) modeling into the machine learning-based recurrent event characterization models enables the integration of the existing knowledge into the learning process. Utilizing the PKI modeling, further improvement can be possible through the mapping between existing knowledge and desired response of the model. Prior-knowledge Input (PKI) is one of the knowledge-based methodologies is considered to obtain the maximum performance of the proposed characterization network. PKI utilizes the existing knowledge of driving events as supplementary input alongside the sensed input signals and is modulated to the existing characterization network as an add-on.

1.5 Structure of the Thesis

The thesis is comprised of six distinct chapters. The remaining sections of the thesis are organized as follows:

Chapter 2 represents the background and literature review. Section 2.1 introduces the safety vehicular applications in ITS, in Section 2.2 we discuss the studies on a specific type of vehicular safety application that is the focus of this thesis (DECs). In Sub-Sections 2.2.1 and 2.2.2 we categorize the different methodologies that are suggested in the literature in order to build a DEC. Section 2.3 is focused on the major challenges and issues facing the deep learning tools used in DECs. Lastly, Section 2.4 presents the technologies and state-of-the-art works on the tools utilized throughout this thesis.

Chapter 3 proposes the methodology and explains the deep-learning network architecture used as the signal augmentation module. The section proposes to tackle the bias issue in driving datasets and to fulfill the first objective of the thesis. The chapter starts with a brief introduction of the technology followed by the problem statement. Section 3.1, explains the overview of the system with a detailed formulation of the network. Separate steps of the experiment, from driving dataset collection to the training process of the network, are provided in the Sub-Sections 3.1.1, 3.1.2, 3.1.3. The results of the experiments (in Section 3.2) are also included in the chapter.

Chapter 4, presents our solution to the driving event characterization problem on inertial signals. After a brief overview of the diving event characterization models and the problem statement, Section 4.1 elaborates on the proposed pipeline and provides a breakdown of each component in the sub-sections, and Section 4.2 represents the performance metrics. Reflection on the outcome of the extensive internal structure investigation of the network is provided in Sub-Section 4.2.2.

Chapter 5, represents the last contribution of the thesis. The chapter proposes the use of the prior knowledge-based solutions to optimize the existing DEC networks. The chapter first introduces the knowledge-based modeling and states the tackled problem, Section 5.1 provides a system overview followed by detailed formulation and process descriptions in the sub-sections. Section 5.2 represents the detailed discussion on the obtained results.

Ultimately, Chapter 6 winds up the approaches discussed in the thesis, an outcome is conferred, and we review possible additions to be made in the future and finalize the thesis.

Chapter 2

Background and Literature study

2.1 Intelligent Transportation Systems and Safety Applications

The popularity gain of mobile devices that are heavily equipped with onboard sensing enables the mobile platforms to be employed as participatory vehicular sensing units [10, 11]. Vehicular sensing can also be leveraged through the sensing and computing equipment in modern vehicles. Using the information generated in vehicular networks, large-scale data can be mined for vehicular safety and traffic engineering applications in an intelligent transport system (ITS) platform. Additionally, vehicles have the luxury of local power generation; therefore they can be equipped with powerful computing devices that are capable of performing local computing. Moreover, the low communication response time introduced by 5G [21] can be utilized by vehicles and mobile devices which introduces more opportunities to vehicular safety applications.

The available technologies in an ITS further incentivize the investigation and refinement of the vehicular safety applications that can leverage the tools [22]. Vehicular safety applications attempt to improve the safety of the passengers by processing and communicating the locally sensed information V2X [23]. This process seems to be the common approach of the majority of the literature that propose vehicular safety applications.

As mentioned, vehicular safety applications expand to many areas of safety while operating in a similar fashion in essence. For instance, crash prevention application in [24], as well as many non-safety applications, start with input data/mobility followed by a set of actions/calculations that lead to detection. Moreover, Lane change assistance [25], Emergence response time reduction [12], Cooperative collision warning [26], and other safety applications function in the safe way. Vehicular safety applications are extensively surveyed by the researchers as the importance of such applications cannot be overlooked in the future of ITS [23]. Fig. 2.1 shows the researched areas with a focus on vehicular safety applications. Therefore, our focus in this thesis is on the second step of the safety applications (i.e. detection of driving behaviors).

Detection of risky driving behaviors (e.g., harsh cornering, harsh braking, aggressive acceleration, etc.) has found substantial importance in vehicular safety applications [27] since the knowledge of the risky driving behavior of the adjacent vehicles can be a piece of essential information for the autonomous driving models. There have been a variety of solutions presented to characterize such events, which have proven the Driving Event Characterization (DEC) models to be effective. DECs can be the key components to the road safety of the vehicles in ITS where highly reliable connectivity is available between the vehicles [28]. In this section. we survey research works that address the challenges and issues of DECs and the tools that can be effective for optimization of such systems.

2.2 Driving Event Characterization Background

It is proven that specific driving characteristics like sudden lane changing, aggressive acceleration, and such unsafe driving behaviors can increase the probability of road accidents [29]. To detect these behaviors, the DECs follow the objective of detecting anomaly behavior and classifying the pre-defined behaviors from the input data [30]. However, the systems can be deployed to different devices and implemented utilizing a variety of methodologies. Fig. 2.3 summarizes the related work discussed in this chapter. Moreover, after comprehensive discussion of the related studies, qualitative comparison of the approaches are gathered in Table 2.1.

2.2.1 Investigation of the Host Devices for DEC

To limit the delay of the detection and improve the effectiveness, the DEC applications are commonly deployed locally on the vehicles. Typically, distributed sensing platforms such

Figure 2.1: Major fields of vehicular safety applications in the literature [23].

as mobile devices and vehicles on the road are the main deployment destinations. Utilizing the in-vehicle sensors or precise onboard inertial sensors in the mobile devices allows for prompt detection with less lag that can be utilized in the safety applications since inertial data are direct measures of the physical forces applied to the vehicles and the passengers of the vehicle [31]. There has been extensive research and proposed schemes in the literature that can be partitioned based on the deployment device as follows:

Smartphone-based Applications for DEC

A range of accurate sensor data can be acquired from the accelerometer, gyroscope, and magnetometer of a typical smartphone. Using smartphones as host to DECs provides a highly portable setup that can be utilized to extract information of the speed, position, angle, and trajectory of the vehicles since the passenger who is carrying the device will be directly affected by the same physical forces that affect the vehicle [32]. For this reason and to take advantage of the massive availability of smartphones, many researchers have adopted smartphones as the host device for DEC models.

Eren et al. [32] proposed a system to analyze the driver behavior based on smartphone sensors. The system attempts to work independently of the vehicles by eliminating the need for in-vehicle sensors such as gas consumption, tire pressure, or turn rate measurements. The method is based on accelerometer, gyroscope, and magnetometer data of the smartphones and derives speed, acceleration, deceleration, and deflection angle. Analyzing the obtained information, the system is able to extract the statistics on several driving behaviors with relatively low cost of deployment. Utilizing Bayesian classification, the system can correctly classify 14 out of 15 driving sessions. Although this work is not meant to be used as a low-latency DEC for autonomous driving systems, we can observe the need to reorient the smartphone according to the orientation of the vehicle. All the calculations seem to assume a certain orientation of the mobile device which can reduce the reliability of such methods as reliable tools.

Li et al. [33] presented a dangerous driver detection system that is smartphone-based. The system is created for use in driver behavior monitoring or insurance systems. The detection is achieved via the implementation of a novel yaw detection algorithm from low accuracy accelerometer and gyroscope sensors. Additionally, the system has no requirement for specific smartphone placement. The effectiveness of the proposed algorithms has been put to extensive experiments and the results indicate over 90% accuracy in detection of abnormal speeding and steering despite the smartphone model and placement. This method fixes the orientation challenge by introducing a coordination algorithm. However, the experiments are done with user-defined thresholds that can be adjusted as the user demands. This shows that different smartphones might experience different physical forces because of their various sizes and locations in the vehicle.

Researchers in [34] created a smartphone driver scoring application called Health Driving. The software solely operates on the built-in accelerometer of smartphones. They proposed a reorientation calibration algorithm to accurately gain acceleration data of the vehicle from the acceleration data of the smartphone. The application can then measure the seriousness of the condition of the road and score the driver detecting the total occurrence of certain driving events. It is built on ISO 2631 standard which is an evaluation method for human exposure to whole-body vibration and shock. This method employs statistical scoring as opposed to definitive detection output. To utilize the DEC for autonomy, the system should be able to detect defined driving situations.

Vavouranakis et al. [35] contributed to sensor accuracy improvements with sensor calibration and data pre-processing methods. Using the calibrated sensor data, the authors designed a driving event recognition system that can characterize safe and unsafe events and display messages to the drivers. The focus of this word is on creating a methodology for insurance cost estimation applications; however, the heavy need for calibration of smartphone data correction is obvious.

As mentioned, smartphones have been major devices of choice for DECs because of the readily availability and relatively affordable sensing and computing power [36, 37]. Aggressive driver detection systems [36] and fuzzy modeling [38] of driving behavior recognition using smartphones are amongst the most recent smartphone-based proposals for DEC which shows the importance of the mobile devices.

However, smartphones suffer from inaccuracies caused by the sudden movements of the device that may cause false detections. Such movements can be caused by human interaction or the drops in the vehicle; therefore several studies attempted to provide more sophisticated mathematical calculations to compensate for the inaccurate measurements of portable sensors [39, 40]. These systems aim to distinguish between the device and the vehicle movements in order to reduce false event characterizations caused by the movements of the device. Additionally, smartphone orientation is another challenge that can cause issues. Since modern vehicles are now equipped with computing and sensing power comparable to smartphones, DEC models can be directly deployed to the vehicles. Utilizing the direct contact of the sensing device with the vehicle eliminates the need for extra calculations.

Vehicle-based Applications for DEC

A similar route is taken by researchers in the case of input and algorithms in the vehicle-based DECs. Speed, position, visual context, GPS information, etc. are the main utilized inputs. However, in addition to static and dynamic algorithms, more sophisticated intelligence-based models can be exploited [41, 42] with higher computing power and energy in the hand of a modern car.

There has been increasing attention by researchers to computer vision systems to analyze visual data as it has become possible by the advancements in modern cars. The vision-based systems often utilize cameras mounted in the vehicles observing the road ahead. Earlier in our work [12], we proposed an optimized CNN-based neural network to classify the severity of the road accidents. Doing so allowed us to build a model to more accurately suggest and notify the appropriate first-responders to the accident scene with no human interaction.

Jain et al. [43] proposed a system that analyzes road conditions. Having the road condition information, the model tracks the driver's eyes to anticipate driving maneuvers before occurring. On the other hand, in [44] Maaloul et al. worked on statistical optical flow modeling using video input to detect accidents. Combining the extracted optical flow features with a statistical approach, they were able to achieve promising detection results. It is worth noting that visual systems are dependant on the line-of-sight of the visual sensors which can be blocked at times [45], therefore for the sake of reliability, we chose in-vehicle sensory data for precise and reliable analysis of vehicular events.

On the other hand, AIBA [46] is an AI-based model for behavior arbitration in autonomous driving. The system imitates a human cognition model to first gain a driving scene understanding and then approximates the human driver behavior based on a supervised process that can be put to the test in driving simulators. More computationally expensive modeling can be immediately seen in the proposed vehicle-based systems.

Behavior Explanation with Fusion (BEEF) [47] is presented to provide an architecture to verbalize the driving behaviors of the autonomous driving systems. A supervised deep learning (DL) model, based on human driving annotations, combines several levels of information leveraging recent methods in multi-modal fusion. The model is capable of analyzing temporal and spatial aspects of the input data. Additionally, the analyzer is linked with a VQA module to detect the proper verbose output. The capability of recurrent neural networks (RNN) in the analysis of timely data is apparent in this study. The vehicular sensory signals also contain temporal characteristics which can be analyzed using RNNs; thus RNN remains a possibly useful tool in tackling the objectives in this thesis.

In another work, Dogru and Subasi [48] target the problem of road traffic to decrease the frequency and severity of the traffic accidents utilizing ITS communications. In the proposed intelligent traffic accident detection system, the vehicles exchange microscopic sensing variables among each other. Speed and the coordinates of the vehicles are collected to be analyzed via several machine learning-based methods. After a comparison between neural networks (NN), Support Vector Machine (SVM), and Random Forest (RF) on traffic data to characterize accident cases on the road, they concluded that the RF is suited best for the application and greatly benefits the drivers by lowering road collisions. In this work, over simplicity of the input data is stated which can explain the underperformance of the NN-based model.

Chang et al. [49] introduce DeepCrash. In order to adapt the vehicular safety applications for accidents that happen in sparsely populated areas, the system analyzes visual and sensory information. The vehicle sends the information to a cloud-based deep learning server for accident detection and further automatic emergency notifications. The authors argue that the accidents in sparsely populated areas call for such systems as the drivers might not be able to rely on assistance from other drivers. In the conducted experiments, DeepCrash is capable of completing the call for emergency responders in around 7 seconds. This work attests to the flexibility of the DL-based algorithms that can run on a variety of input types.

Although the smartphone-based applications introduce vast availability and significant cost efficiency to the DEC, the sensitive characteristics of the safety applications require onboard vehicular applications when it comes to critical safety applications. As it can be seen in this section, there is a shift towards vehicle-based safety applications in the literature for genuine reasons mentioned. Moreover, we chose to use the vehicular inertial sensory data for further analysis as opposed to visual driving data for the reasons mentioned. Hence, after careful studies on the previously done works, the vehicle-based DEC is chosen for our experiments to tackle the critical required precision by the safety applications of modern vehicles.

2.2.2 Investigation of the Methodologies for DEC

We mentioned the importance of the host device and categorized the DECs based on the deployed target; however, the choice of the device is not the sole enabler of DEC for critical vehicular safety applications. Therefore, to review the related research in the field, the methodologies employed in the driving behavior characterization systems must also be explored. A literature review allows us to forge an initial understanding of the different approaches taken by the researchers in the area to improve the driving characterization models. Luckily there are strong surveys around the issue [50, 51] that along with several more recent publications could help us to choose the most promising approaches. Although there may be different modeling schemes, such as static and dynamic threshold models, statistical, machine learning and deep learning approaches, the systems can be classified into intelligent and non-intelligent categories of models.

Non-intelligent DEC

A series of studies were conducted on the feature extraction methods to gain knowledge from the inertial or global positioning sensed data [52]. Such methods employ diverse signal processing and filtering techniques to acquire noiseless data that best reflects the driving event data characteristics. In the detection phase, models are often monitored by static or dynamic threshold-based methodologies [53] often applied to Inertial Measurement Unit (IMU) data. In the literature, the models are put through extensive experiments to identify the static thresholds of distinct risky driving behaviors in the data. However, the studies on the dynamic threshold-based systems intend to introduce a feedback loop that continuously updates the threshold values concerning the measured metrics utilizing more complex mathematical calculations.

Han and Yang in [54], proposed a reckless driver identifier system to recognize dangerous and non-dangerous driving utilizing automatic black boxes that collect driving information. The proposed system is reported to be useful for insurance fee estimation by first categorizing the sensor data into four distinct events using their pre-defined set of rules, and then applying static thresholds to label each event as dangerous or non-dangerous. The system can identify dangerous driving from non-dangerous ones with high accuracy, which can be used for insurance fee estimation. Although the data collection scheme of this work is interesting for our experiments, the detection modeling is designed as a series of static thresholds. If measurements all simply exceed certain thresholds, the proper driving event is declared. As discussed, this type of modeling has no tangible benefits or modern autonomy systems.

Amin et al. [55] propose accident detection models based on GPS and vehicle speed to send information to emergency service in a timely fashion. At the core, the proposed system analyses the position of the vehicle and the GPS speed through a map matching algorithm every 0.1 seconds. Comparing the vehicle speed in every iteration to the previous moment, accidents are detected if the speed drops faster than the safe threshold. Furthermore, if the map-matching algorithm identifies the position of the vehicle to be outside the pre-defined roads, the accident situation is declared. Despite the novel solution, the reliability of the system for safety applications could be questioned. The system relies heavily on the map database which needs constant updates to avoid false detections in a real-world situation.

More recently, by utilizing satellite GPS data, the authors in [56] proposed an irregular driving identification system. The proposed model in the study aims to increase the accuracy of the positional measurement to less than 50 centimeters and was able to accurately identify driving behavior applying a static threshold to the calibrated GPS data. After careful analysis of this work along with the studies that took a similar approach, the major flaw of the GPS data in vehicular safety applications becomes apparent. This type of study is mainly limited to highway data since the urban terrain jeopardizes the trustworthiness of the GPS data.

A series of studies were conducted based on recording the sensor data during a trip. These types of systems are especially beneficial for fleet management where it is important to have information of each trip and to record the drivers' behavior in detail. In [57], the authors propose a static threshold applied on a unique system that combines velocity and acceleration of the vehicles while considering the vehicle dynamics and the road conditions to classify driving behaviors. The experiments are conducted on pre-recorded minibus taxi trip data to characterize the driving behavior which is beneficial for fleet management systems.

Fernandes et al. [58] present a system called HDy. A vehicular safety application utilizing OBD-II data and IEEE 802.11p for alert dissemination. HDy has a DEC at the core of the system and is proposed to constantly monitor accelerometer, gyroscope, and magnetometer data via OBD-II. The system utilizes a low-pass filter to clean the inherently noisy vehicular sensory signals and outputs the driving events using a sensor fusion algorithm and threshold monitoring. The collective information is then monitored for driving events such as collisions and rollovers and communicated with the surrounding vehicles. The authors mention that they intentionally implemented a countdown in the software to cancel the notifications caused by false-positive detections. This introduces delay if used in critical safety situations. HDy shows us the importance of the DEC precision. If DECs are to be used with no human intervention, they must be able to perform close to perfect.

On the other hand, Hu et al. [59] aims to dynamically identify the normal driving behavior and alert the abnormal events when the measurements exceed the normal region. In this work, a personalized driver model is proposed that predicts an abnormality index value to evaluate the drivers. As they mention in the article, the work is limited due to the lack of vehicular test data which prohibits broader experiments and extension of their study. Mitigation of the effects of driving data shortage opens possible areas of research to enhance the development of DECs.

Driving speed and correlation between lateral acceleration and velocity are studied in [60] where the authors conclude that lateral acceleration and vehicle speed are inversely related to each other by analyzing trip data of vehicles on highways with different topographies and speed limits. The study shows that lateral acceleration decreases as the vehicle speed increases, therefore risky lateral threshold value should be dynamically changing with the speed of the vehicles. This work makes the highly dynamic characteristics of the driving events evident. The addition of a single feature (i.e. lateral acceleration) into the modeling significantly increases the accuracy of the DECs. Thus, to utilize all the possible features, more intelligent systems that can incorporate the massive amount of features are needed to build highly accurate DEC.

Ultimately, the threshold-based methods rely heavily on the physical characteristics of individual vehicles or calculations that are static and might not apply to real-world events [61]. To achieve the full autonomy of Level-5 autonomous systems, the core decision-making model (DEC) must be fully automatic and applicable to every situation. After reviewing the non-intelligent researches, to find studies that have fixed the observed issues and can analyze several input features effectively, we explore the intelligent tools.

Intelligent DEC

Intelligent systems can be categorized based on the system input and the features they analyze. Intelligent systems perform the characterization task by using either visual driving context, inertial sensing event data, or hybrid in which one type of data is applied to augment the other input [62]. Moreover, there is a growing body of studies in the field of neural networks that makes time-series and visual analysis via neural networks more viable than before [63], therefore various studies proposed DECs with the aid of visual or signal processing tools using intelligence-based methods.

Alkinani et al. [64] conduct a comprehensive survey in this field and concludes that intelligence and deep learning-based approaches show the highest performance indications to enhance DECs. Deep learning innovations are amongst the most promising tools in the detection, predictions, and classification of abnormal, inattentive, and aggressive driving behaviors. Earlier we mentioned the possibility of utilizing recurrent neural networks for the analysis of inertial sensing data. There are more innovative intelligent algorithms presented in the literature which we will discuss in this section.

The modern automotive industry relies on machine learning and neural networks for driver assistance and autonomous driving systems [65, 66, 67, 68, 69]. Computer vision methods have also been in the center of focus in the literature to characterize driving events [70]. For instance, an optimized image processing method is utilized in [12] to categorize driving incidents and make the decision on alerting the first-responders to the location. A pre-trained convolutional neural network is used in [71] on the phase-space reconstructed vehicle trajectories to evaluate driving behaviors. The study quantitatively evaluates the abnormal behaviors to then detect the abnormal drivers. Moreover, the distracting behaviors that lead to unsafe driving are the subject of various studies. Rao et al. [72] take the approach of processing the captured camera data via convolutional neural networks (CNN) and Principal Component Analysis (PCA) to whiten the camera feed and classify the whitened images as distracted and non-distracted driver activities.

In [73], the authors present the idea of driver's brain signal processing with machine learning-based clustering and classification algorithms to study driving behavior. Pradhan et al. [74] studied accident severity detection models based on police reports and road features, and compared Neural Network (NN) and Support Vector Machine (SVM) models and conclude that SVM outperforms NN model unless the analyzed data is in big volume.

Considering the massive vehicular data and the most recent innovations in deep learning methodologies, we explore the state-of-the-art solutions in neural networks that are beneficial to vehicular safety applications.

An and Cho [75] target the problem of anomaly detection using variational deep auto-encoder (VAE) networks. Adopting the reconstruction probabilistic measure, they can score the anomalies which are said to be a better scoring method than the reconstruction error of the auto-encoder method. The combination of neural networks that perform well on massive data, generative solutions, and anomaly detection gives us an extremely useful tool for vehicular safety applications because of the characteristics of DECs that were mentioned before.

Hou et al. [76] proposed a novel method for the construction of VAE that preserves the spatial correlation of the input data. The suggested model achieved state-of-the-art accuracy in vision-based predictions. However, this work shows that such systems can be adopted for inertial signal reconstruction and detection of an anomaly in driving signals, hence VAEs are in our list of possibly useful tools for the thesis. Such methods could be useful in tackling the lack of driving event dataset and the characterization of anomalous driving data.

Because of the aforementioned issues in non-intelligent methodologies and variety of intelligent tools in hand, this thesis focuses on intelligent solutions to enhance DEC models as can be seen in Chapters 3, 4, and 5.

2.3 Issues and Challenges in DEC

Autonomous driving introduces a new set of obstacles and challenges in the deployment of DEC which are not observed in a traditional vehicle since autonomy relies on the detection models for critical decision making. To achieve higher levels of trustworthiness that allow the DECs to be used in critical safety applications, it is obvious that intelligent-based models are the preferred tools. However, the intelligent algorithms are more suited while there is an abundance of data in hand and deep learning methods inherently require massive amounts of data [77]. Furthermore, not only does a DL-based architecture need volume in the data but also there is a requirement for variety, velocity, and veracity in the input [78]. Volume, variety, velocity, and veracity are the four "V"s of big data which are the newly presented challenges by the deep learning algorithms to the DEC.

In this section, we describe research works that address different challenges and issues of DEC. Fig. 2.2 shows the challenges that are being addressed by this thesis in regards to DEC models.

2.3.1 Addressing the Four Data Issues of Deep Learning in DEC

Although neural networks can overcome the shortcomings of the previous non-intelligent models regarding accuracy and reliability in the detection of various event types, the lack of

Figure 2.2: Five challenges in deep learning-based DECs as stated in [78].

anomalous driving patterns and heavy imbalance of abnormal event data is inevitable in a vehicular setting. Therefore, the datasets related to driving events often lack risky driving events which are used to train an intelligent DEC. To properly train a neural network for the task, there is a need to address the introduced challenges

- **Volume** issue of the neural network-based driving event characterization models are studied in [75, 79]. The studies adopt variational auto-encoder to generate various timely data which can also be beneficial to our use case.

- **Variety** can also be enhanced via the VAE networks targeting the lower populated events in the generation process.

- While **velocity** issue can only be manually addressed by the constant update of the dataset, the issue of **veracity** can be mitigated using more advanced filters that are suited for the characteristics of the vehicular sensing data. In [76], the encoder network focuses on the underlying spatial features of the signals. These types of studies provide practical ideas for this challenge.

The identified tools that can be of possible merit to consolidate the issues are recurrent neural network (RNN) [80] architecture and auto-encoder (AE) networks. The utilization of RNNs allows the deep learning models to recognize the sequential characteristics of the signals and use signal patterns for prediction or detection [81]. Recurrent networks do not build on the physical features of a vehicle; therefore allowing the models to be globally used regardless of the type and physical characteristics of the vehicle by learning the underlying features of the input data. The process of learning the sequential characteristics in the driving event signals improves the "Veracity" of the inputs by eliminating noisy data. Additionally, auto-encoders are a type of neural network that when combined with the RNN algorithm can generate synthetic signals that are noiseless and highly accurate [82].

2.3.2 Optimizing DEC

As mentioned, extensive research has been done to boost and optimize the performance of different DEC models. Neural network-based processing of signals, augmenting visual and contextual driving data, etc. are among the efforts. However, there is an opportunity for further improvements in intelligent-based DECs by using novel methodologies in AI. For instance, attention networks [83] and knowledge-based network optimization [84] in addition to time-series tools such as RNN and AE are built to allow the neural networks to specialize in different types of data.

Attention models enable the networks to find the significant information in the signals for a better characterization process [85]. On the other hand, a knowledge-based optimizer [86] provides the networks with factual knowledge of the events for further optimization. Motivated by the aforementioned innovations, we ultimately attempt to optimize an in-vehicle intelligent DEC for level-5 driving automation.

2.4 Investigation of Related Work in the Effective Tools

Neural networks have surpassed machine learning and other intelligent methods in terms of performance. Additionally, numerous studies have been proposed to accelerate the feasibility of the novel intelligent models and expand their use cases to other domains. The signal processing domain has also been affected by the aforementioned fields, therefore many researchers are focusing on deep learning methods to improve the state-of-the-art signal analysis systems. In this section, we briefly explore the state-of-the-art methodologies in the field of the four deep learning tools used throughout the thesis to tackle the previously mentioned objectives. We first explore the recent innovations of deep time-series analysis and deep auto-encoder networks to fulfill the first objective of the thesis. Then, the recent works on attention and prior-knowledge input networks are surveyed.

2.4.1 Tools in Time-series Analysis

Kalman filter-driven collision detection systems are widely studied in the literature. As an example, deceleration intention inference system (DIIS) [87] can be considered, which aims to discover vehicle deceleration intentions by incorporating Kalman filter and neural networks. These networks have shown to be capable of acquiring the attributes of input data and be a better data representation scheme to systems. Inspired by such systems, our work utilizes an encoding approach for the task of event characterization. In this thesis, we employ time-series learning techniques in order to characterize our unique type of data which has spatial and temporal characteristics.

The most recent developments in artificial intelligence and machine learning have made the real-time detection of the driving events more feasible and accurate [88, 89] by catering accurate methods for the analysis of time-series data. More specifically, the recurrent networks are able to store a brief history of the past input data at any time that makes them the feasible tools for time-series data analysis. Recurrent models such as Long short-term memory (LSTM) [90] architecture allows the machine-learning systems to reserve and consider longer history of the data in the decision making by the networks. Furthermore, storing recent data information is critical for pattern recognition methods employed for driving event characterization systems. LSTM networks are a refined type of recurrent neural networks that are utilized to extract temporal features in time-series data [91].

The major issue of using such networks for high-frequency signal processing is the phenomenon of vanishing gradient [92] that causes the recurrent networks to discard longer-term features. However, novel resolutions are proposed in [93, 94] which enable the longer-term processing of time-series data using RNNs. Such networks are the main enabling factors of the contemporary weather forecasting and language models which heavily depend on the sequential history of the data [95]. Furthermore, there have been numerous studies in order to expand the application of recurrent networks into data compression [96].

2.4.2 Auto-encoder Networks

Combining generative and discriminative networks, a denoising auto-encoder is introduced in [97] in which the model has the ability to take the input features on the fly. An auto-encoder network with the support of random sampling from the encoder latent space is proposed with a generative objective in [98]. The model mixes a combination of adversarial and reconstruction losses, but unlike Generative Adversarial Networks (GAN) discriminators, the authors employed progressive growing of generator and encoder networks.

Moreover, auto-encoders (AEs) as suitable tools for analysis where data is limited, are studied for anomaly detection [99, 100]. For instance, Malhotra et al. [101] attempt to perform 'Remaining Useful Life (RUL)' analysis of machines incorporating AE networks and anomaly detection techniques. Safe airplane navigation analysis is performed on rare data using such techniques in [102]. As stated by the researchers, AE networks have proven to be more reliable input representation schemes to systems.

The feature extraction of auto-encoder networks is also studied [103]. In this work, the authors present the relational auto-encoders which improves the issue of the auto-encoder where they fail to consider the relationships of data samples.

The study of the vast use cases of the auto-encoder networks and the networks that are designed to analyze unique types of data that are similar to driving event data persuades us to experiment with auto-encoders (AE) in our work. As can be seen in Chapter 3, an AE-based solution is proposed to tackle the first objective of the thesis.

2.4.3 Attention Networks

In addition to the recurrent networks and auto-encoders that are utilized in the data augmentation and detection modeling in Chapter 3, attention-based neural networks are also a focus in an attempt to optimize the existing classification model.

In [104], a complex attention-based speech enhancement framework is presented. Chen et al. [85] utilizes the attention networks in machine reading comprehension systems. Furthermore, attention is expanded to various use cases including the AE networks [105, 106], and introduces the benefit of selectively focusing on sections of the input data that contain the desired events. This feature of the attention networks holds value for DEC models as the events in the vehicular sensing data are scarce and sparse. The effectiveness of the attention architecture is experimented with in Chapter 4.

2.4.4 Knowledge-based Modeling

Knowledge-based modeling is suitable for the driving event classification systems since it only requires the input-output relations in a supervised manner [107]. In a deep auto-encoder network, a large training dataset is required to meet the optimal stopping conditions which are not convenient in vehicular applications where there is an imbalance in the data.

In [86], machine-learning detection models are boosted by novel knowledge-based methods and feature selection mechanisms. Utilizing Prior Knowledge Input (PKI) and Prior Knowledge Input with Difference (PKID), the authors can significantly improve the average accuracy of the existing machine-learning methods. The PKI is widely adopted in neural network-based modeling systems such as adversarial task detection in mobile crowd-sensing to microwave modeling for computer-aided design [108]. Using the PKI modeling, it is possible to remarkably improve the driving event characterization model in terms of accuracy and overall performance.

In Chapter 5, we intend to illustrate the advantages of the knowledge-based optimization on the DECs.

Table 2.1: Collective overview of the literature.

	Method		**Device**			**Issue Tackled**		
	Non-intelligent	Intelligent	Vehicle	Phone	Offline	Augmentation	Modeling	Optimization
[29, 44, 45]		✓			✓		✓	
[30]		✓	✓				✓	✓
[31]	✓				✓		✓	
[32]		✓		✓			✓	
[33, 34, 35, 36, 37, 39]	✓			✓			✓	
[38]		✓		✓	✓		✓	
[40]		✓		✓			✓	
[41]		✓			✓			✓
[42, 12]		✓	✓				✓	
[43]		✓			✓		✓	✓
[46]		✓	✓				✓	
[47]		✓			✓			✓
[48]		✓	✓				✓	
[49]	✓	✓	✓					✓
[52, 55, 56]	✓				✓		✓	
[54, 57]	✓		✓				✓	
[58, 60]	✓			✓			✓	
[61, 71, 72, 73, 74, 82]		✓			✓		✓	
[75, 76, 79, 83, 84, 85, 86]		✓			✓			✓
[88, 89, 93, 96, 103]		✓			✓	✓		
[105, 106]		✓			✓	✓		✓

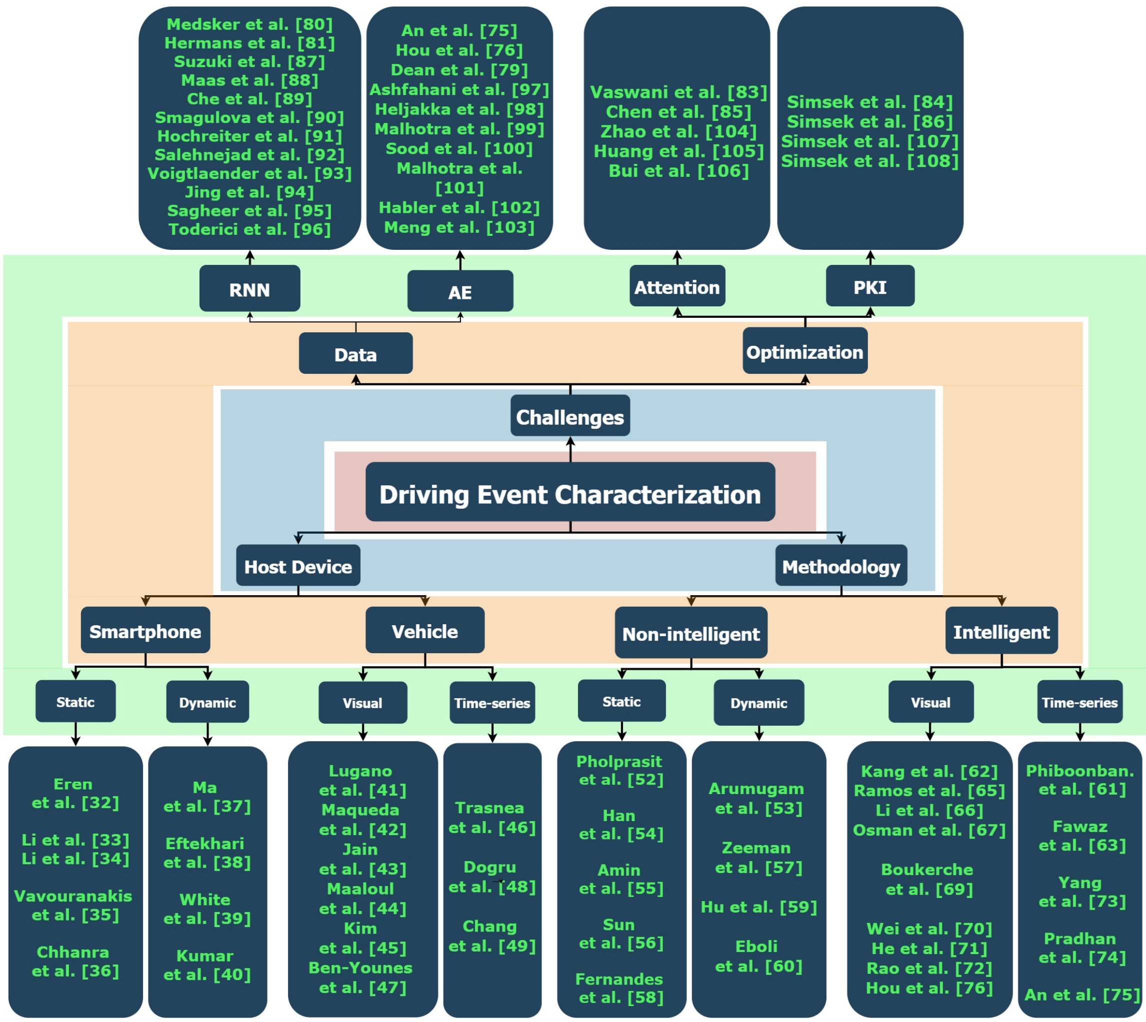

Figure 2.3: Classification of the research works in DEC.

Chapter 3

Driving Event Characterization Under Scarce Driving Data

Event characterization systems are commonly implemented based on inertial and visual sensor inputs and work by identifying specific events from raw vehicular data. There are specific driving events and behaviors that are commonly known as anomalies. These events include aggressive accelerating (AA), harsh braking (HB), aggressive lane changing and turns to the left or right (HL and RL), and sudden impacts or accidents. Systems usually attempt to differentiate these events from regular driving (RD) characteristics. For event types that have a direct effect on the physics of the vehicle, the Inertial Measurement Unit (IMU) sensor data is the input choice since this sensor gauges physical specific forces, angular rate, and orientation alteration of the vehicle. Moreover, the analysis of time-series information has become more feasible by recent developments in artificial intelligence and recurrent neural networks and more specifically by Long short-term memory architecture (LSTM) [90] improvements. Recurrent networks store the state of the past input data and combine it with the input's current state to obtain timely information that holds the characteristics of IMU data. The ability to store data is a crucial feature for any event detection system.

Despite the surge of vehicular sensory data as a consequence of more advanced vehicles equipped with highly accurate sensors, there is a lack of available effective data for studies on harmful driving events [16]. Moreover, acquiring such data can be costly and not feasible. On the other hand, acquiring driving data from vehicular simulation does not contain reach information for safety applications [109]. Therefore, an alternative input modeling is needed to capitalize upon the real-world data in hand.

Deep learning (DL) methods are effective tools to address various big data analytics problems [78]. Therefore, data analytics augmented with deep learning can facilitate smart city ecosystems where connected objects can acquire and communicate data in a network. Just to name a few, path suggestion for congestion avoidance and license plate recognition (PLR) are among possible applications [110]. Furthermore, with the rise of self-driving cars, Artificial Intelligence (AI) and DL-based solutions are now heavily researched [111] for safety and non-safety vehicular applications. For instance, LSTM networks that are types of neural networks utilized when the input data has sequential meaning [112] are employed in time-series analysis. Applications such as weather forecasting and language processing are based on sequential data analysis and are enhanced with the use of LSTM networks. Auto-encoders, on the other hand, are types of networks that do data translation and extract meaning from the data [113]. An auto-encoder is implemented in two modules. First, an encoder module maps the input to a fixed-size vector representation. The target output is then generated by a decoder module using the fixed-size vector. Combining auto-encoders with LSTM, the network is more suitable for IMU sensor data which is sequential and is used in vehicular settings.

In this chapter, to mitigate the effects of data scarcity, a state-of-the-art recurrent network is chosen and optimized as the baseline for the task. The network is combined with feature extraction and classification models for better performance. Recurrent networks are employed for memorizing temporal dynamic behaviors of the input data. Moreover, the feature extraction components are utilized to extract the spatial characteristics of the data. We propose an LSTM auto-encoder structure to not only overcome the issue of data scarcity by performing data augmentation but to also improve the detection model performance by providing the characterization network with the underlying information in the signals. Doing so allows the system to reconstruct precise synthetic data from existing signals for a more robust training process.

As illustrated in Fig. 3.1, the raw sensor data is acquired from Raven sensors and infused with random noise to achieve a noise-less input representation vector. Vectors are used in the training process by a decoder network to reconstruct noisy input data. A multi-layer perceptron network is trained on the generated data, and inference is done by directly feeding the vectors to the network.

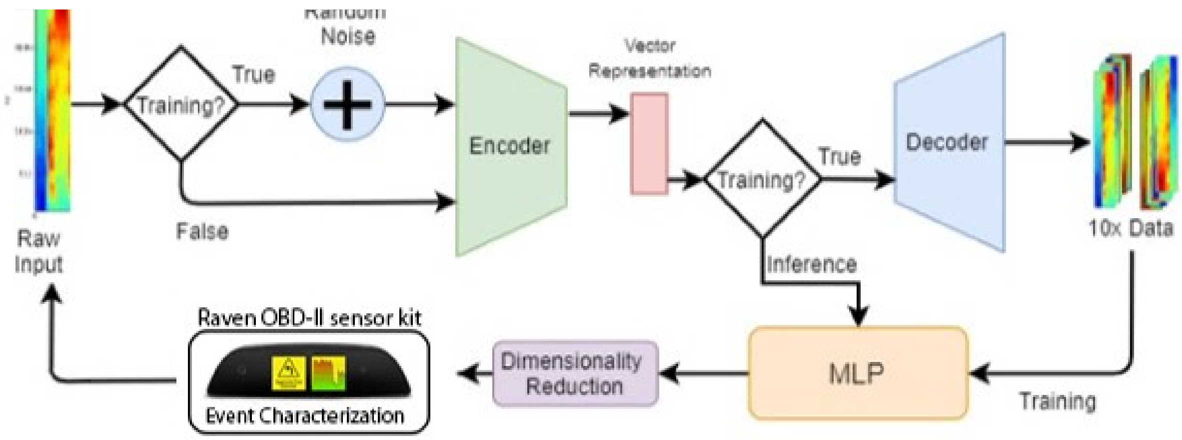

Figure 3.1: Process flow of the auto-encoder network utilized in signal augmentation.

3.1 Proposed Solution

An auto-encoder-based DEC methodology is the main proposal of the chapter to boost the performance issues caused by the scarcity of risky driving data. The model not only extracts a multi-dimensional feature representation of the noisy sequential input data but also reconstructs accurate time-series data for a robust training process. The trained model can seamlessly infer the unseen data by utilizing the proposed pipeline as mentioned in subsection 3.1.3. The data gathering procedure is explained in subsection 3.1.1. To put the proposed input scheme to a test, we have also tested several recurrent and feature extraction networks, the best of which are chosen as a baseline for our proposed model. The details of the baseline model are presented in subsection 3.1.2.

3.1.1 Gathering Driving Event Data

Our methodology takes advantage of 2-dimensional accelerometer data. To collect proper data, Raven OBD-II sensor kit [114] which is an android-based connected vehicle device is mounted on a car dashboard with the following orientation specifications: X-axis along

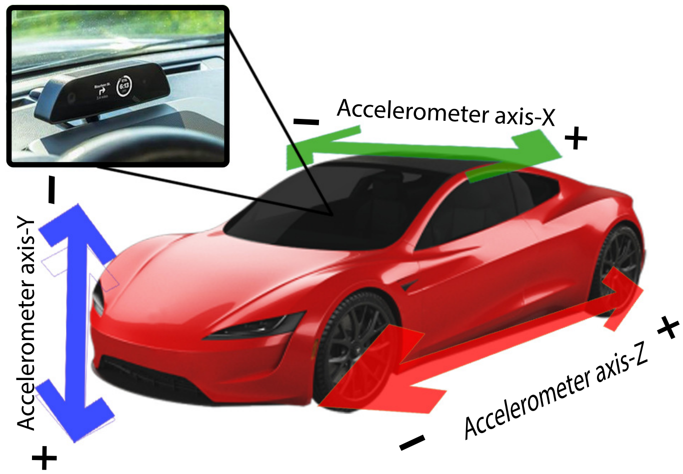

Figure 3.2: Vehicular sensor orientation in the data collection stage.

the latitudinal, Y-axis along the altitudinal, and Z-axis along the longitudinal axis of the vehicles as illustrated in Fig. 3.2. We collected roughly equal distribution of driving events among uncommon behaviors such as harsh braking (HB), aggressive acceleration (AA), harsh left and right lane changes (HL/RL), and regular driving events (RD). It is worth noting that to build the recurrent-based model in subsection 3.1.2, a part of the collected data is used for training of the model while the rest is used for testing. Nevertheless, the proposed auto-encoder-based model as mentioned in subsection 3.1.3, can be trained on reconstructed data while being tested on the full dataset.

3.1.2 Baseline Recurrent-Based Models

Considering our unique data which is noisy and has temporal and spatial features, we experimented with combinations of convolutional and recurrent networks to extract spatial and temporal features from the data, respectively. The features are then fed to a multi-layer perceptron network for event characterization. The best performing network was chosen as the baseline for our proposed system. The baseline network is a three convolutional block-model (as shown in Fig. 3.3). Each convolutional block consists of two convolution layers and an average pooling layer. Convolutional blocks are used to extract spatial features for all three axes of data along the axes of time, i.e., X, Y, Z, and t-axes. The number

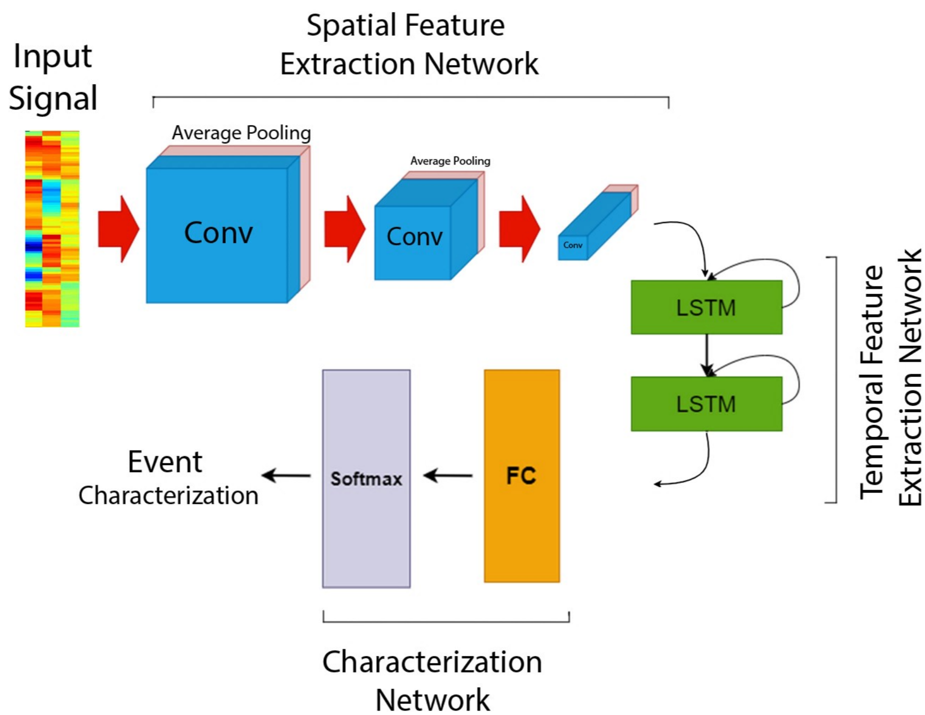

Figure 3.3: The structure of the baseline classification recurrent network.

of convolutional kernels increases as the kernel size decreases along with the network. The spatial features are then fed to a recurrent long short-term memory (LSTM) network to gain temporal information. Eventually, the information is streamed into a 2-layered perceptron network with a softmax output layer for event characterization. With the use of an optimal learning rate and cross-entropy loss optimization, the model converged to an accuracy rate of above 80%.

3.1.3 Auto-Encoder Based Model

The problem addressed by this study is two-fold: 1) scarcity of diverse anomalous data from the vehicular sensor arrays, 2) event classification with high accuracy. The proposed solution to address these two challenges leverages an auto-encoder network in order to extract the essence of sequential data and to reconstruct more feature-rich synthetic data. To overcome the problem of data scarcity, sample data is first mixed with random Gaussian

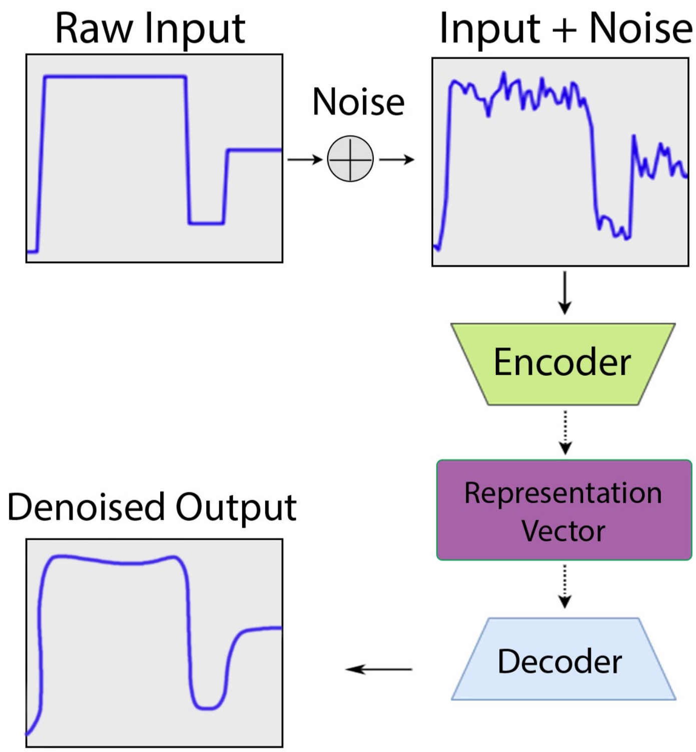

Figure 3.4: Noisification module of the auto-encoder network.

noise as shown in Fig. 3.4 to then be fed into the auto-encoder network. The data that is produced by the auto-encoder is proven to be more precise and less affected by the noise, which is inevitable in the vehicular environment. The decoder module learns to reconstruct the original data from the noisy encoded input.

The encoder module of our model consists of $L = 2$ recurrent LSTM hidden layers and outputs a fixed-length vector representation of the input data. A mirrored decoder network uses the vector representation, i.e., last hidden layer of the encoder module, to generate noise-free sequential data. The encoder and decoder networks work in conjunction to minimize the absolute error (3.1) between original input and reconstructed data points:

$$\sum_{j} \sum_{i} \left\| x^{(j)} - y^{(j)} \right\|^2 \tag{3.1}$$

in which $j \in \{1, 2, \ldots, t\}$ and $i \in \{1, 2, ..., a\}$ where a is the axis dimension of the data and t is the length of the event. Consider a sequential input of dimension 1 and length t ($X = \{x^1, x^2, ..., x^t\}$), the encoder network outputs the vector representation in its last hidden layer ($h_{2e} = \{h^1_{2e}, h^2_{2e}, ..., h^t_{2e}\}$). A mirrored decoder network attempts to predict y^t only when the last node in the last encoder hidden layer is generated (i.e. h^t_{2e}). Using the vector h_e, the decoder network starts reconstructing synthetic data from the last state going backwards (i.e. $\{y^t, y^{t-1}, ..., y^1\}$). In the reconstruction process, the network predicts each state considering the previous state, i.e., $y^t = f(h^t_{2d}, h^{t-1}_{2d})$

Our proposed pipeline utilizes the network to generate 10 times data events, each reconstructed with randomized noise. A multi-layer perceptron (MLP) network is trained on the generated data for event characterization. Each event data is one-hot labeled. The MLP classification network aims to minimize categorical cross-entropy function over our $C = 5$ classes of events to reach the most accurate probability distribution of output. Categorical cross-entropy is implemented with a softmax activation (3.2) of output x plus a cross-entropy loss function (3.3) of target label t to perform multi-class classification.

$$f(x) = \frac{e^x}{\sum_c e^x} \tag{3.2}$$

$$CE = -\sum_c t\, log(f(x)) \tag{3.3}$$

We kept our original dataset isolated from the training process and only trained the model on the reconstructed data. Unseen event signals are fed to the encoder with no added noise for the task of inference. Last but not least, Principal Component Analysis (PCA) dimensionality reduction was applied to the encoded output (i.e. h_{2e}) to visualize the precision of data. The distinct separation between the reconstructed data events is observed which testifies the advantage of having the vector representation as input over the original signals. The results are to be presented in Section 3.2.

3.2 Performance Evaluation

Simulation results are obtained on a dataset consisting of 70 driving events, and the performance of the proposed model is compared to the baseline model. Testings of the baseline

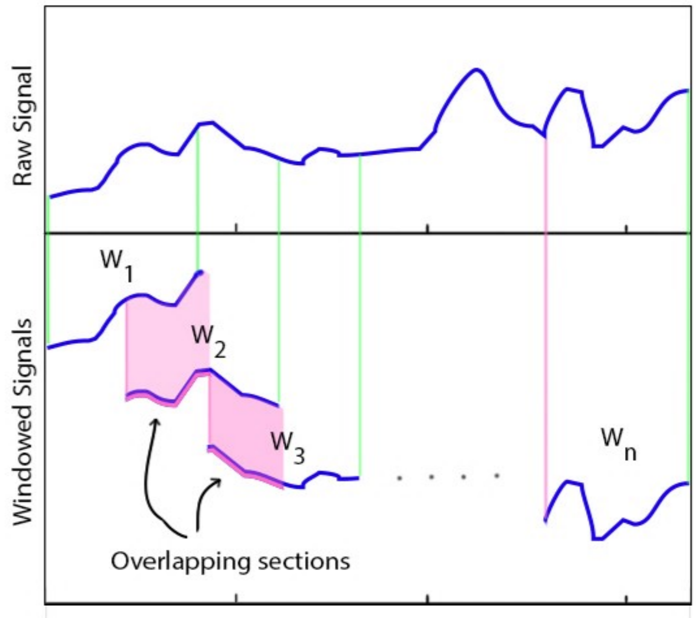

Figure 3.5: Abstract visualization of the signal slicing module.

and the proposed models are performed on an identical test set to validate the effectiveness of the proposed LSTM-based auto-encoder model.

3.2.1 Experimental Settings

To collect proper driving data for the study, accelerometer data of several risky driving events are collected. In total, 70 different driving events are captured with a duration varying from 2 to 3.6 seconds and with a sensor sensitivity at 25 Hz. Furthermore, 13 HB, 12 AA, 16 HL, 15 RL, and 14 RD events are accumulated, and each event is split into 3-axis data of 600 millisecond-windows with a 50% overlap between two consecutive windows as shown in Fig. 3.5. This varied duration of data leads to a total of 572 data slices. A brief overview of the data distribution, as well as the number of slices, is presented in Table 3.1.

As stated earlier in this chapter, the best performing recurrent neural network is chosen as a baseline to demonstrate the performance benefits of the proposed auto-encoder. 400 (70%) of the data slices are chosen randomly and are evenly split between the labels for training. The rest of the data is used for testing. To train the auto-encoder, each event

Table 3.1: Breakdown of the event distribution in the dataset.

Event Type	Event Count	Slice Count
Harsh Brake	13	104
Aggressive Acceleration	12	108
Harsh Left Turn	16	126
Harsh Right Turn	15	121
Normal Driving	14	113

signal is mapped onto 10 input signals each of which is obtained through the original signal and an additive Gaussian noise with randomized weight. The encoding network vectorizes all 10 noisy signals and the decoder module attempts to reconstruct the original signal from each noisy input. All the reconstructed 5720 (i.e. 10 times) data slices are used in training the MLP classification network and the original 572 data events were kept unseen. However, to keep the comparison between the performance of both models relevant, the testing results of the auto-encoder model are also inferred on the 172 data slices that are used for testing the recurrent model.

The auto-encoder follows a self-supervised process as its objective is to generate synthetic data from the input data. To attain this objective, the error is computed as the mean absolute error of each data-point which allows the network to reconstruct the original data from noisy variations. The encoder unit is implemented in three layers of decreasing sizes (i.e. 1000, 500, and 300 nodes), and the decoder unit is a mirrored three-layer network that reconstructs the signal only when h_{2e}^{300} is output as depicted in Fig. 3.6. The characterization module is implemented as a supervised learning process. It aims to minimize the distance between the probability distribution of the hypothesis and the one-hot encoded probability distribution of the input. The network uses categorical cross-entropy to optimize each step. The classification module that is trained through a set of 500 iterations at a learning rate of 0.03 utilizes Adam optimizer with following configuration: $\beta_1 = 0.9$ and $\beta_2 = 0.95$ and $\epsilon = 10^{-8}$.

Ultimately, singular value decomposition-based Principal Component Analysis (PCA) is applied to map the input signal as well as the vector representations onto two-dimensional space for a more thorough comparison. The 2-D mapping of the original data events as

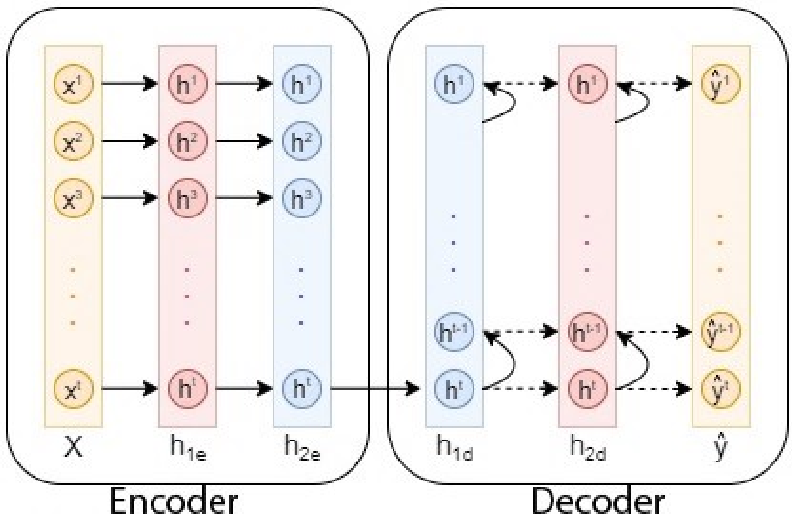

Figure 3.6: Abstract structure of the auto-encoder network.

well as the mapping of the vector representations are given in Fig. 3.7.

3.2.2 Numerical Results and Discussions

The key observations of experiments are as follows: 1) as illustrated in Fig. 3.7, extracting the encoded vector representation of data offers more distinct data separation that carries less noise; hence, it facilitates event characterization. Thus, separation is the primary contributor to the 12% performance improvement by the proposed model. 2) The ten-fold input data with additive noise at the decoder unit helps with the robustness of classification training. Neural network-based classification models are known to be directly affected by the amount of available data. Training data multiplication helps the model to converge to the optimum iteration more consistently in 10 individual runs. Finally, as shown in the table 3.2, training the model with the representation vector input increases the precision. A higher precision value indicates the ability of the model to classify with lower false positives which is a primary concern in event characterization models.

The baseline recurrent model results in 0.81 accuracy on the test set with a precision of 0.77, whereas the proposed auto-encoder model is able to reach an accuracy of over

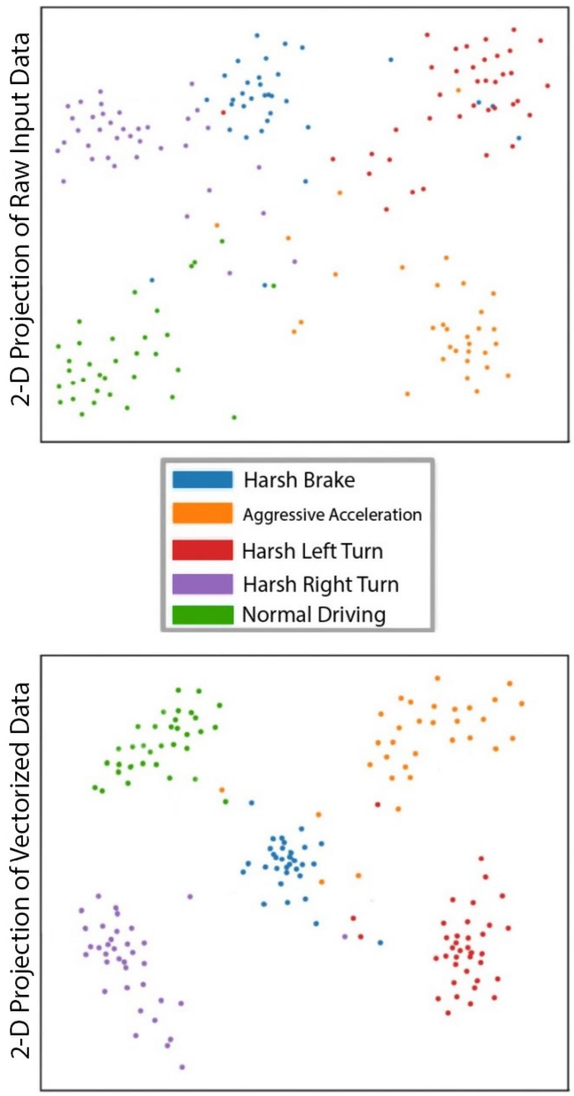

Figure 3.7: 2D projection of the clustered signals after the signal augmentation method.

0.93 with a precision of 0.89. A performance breakdown of both models is represented in Table 3.3 to provide more insight into the efficiency of the baseline (optimized recurrent neural network) and the proposed model (LSTM-based auto-encoder). The more significant improvement areas are highlighted in bold.

Table 3.2: Numerical results of the classification model with the augmented signals.

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Baseline	0.81±0.009	0.77±0.010	0.79±0.009	0.78±0.009
Auto-encoder	**0.93±0.007**	**0.89±0.008**	**0.90±0.008**	**0.89±0.007**
Improvement	0.12±0.002	0.12±0.001	0.11±0.001	0.11±0.002

Table 3.3: Breakdown of the predictions after the signal augmentation model.

Event	**Count**	**False (F), True(T) Predictions Baseline Model**	**False(F), True (T) Predictions Proposed Model**
HB	32	6(F),27(T)	2(F) 30(T)
AA	32	7(F), 25(T)	4(F), 28(T)
HL	38	9(F), 29(T)	**3(F), 35(T)**
HR	36	7(F), 29(T)	**1(F), 35(T)**
RD	34	4(F), 30(T)	1(F), 33(T)

Chapter 4

Attention-Based Driving Event Characterization Model

With the advent of the connected and autonomous driving paradigm, detection of risky driving behavior (e.g., harsh cornering, harsh braking, aggressive acceleration) has become an essential component in a connected vehicle setting [27]. There have been a variety of solutions presented to characterize such events, which have proven to be effective. DEC systems as the fundamental core of accident prevention models coupled with low latency vehicular connectivity provided by 5G and Beyond [21] can be utilized in intelligent transportation systems (ITS) for centralized traffic control systems. Centralized control systems can be effective tools for road safety and congestion control [22].

In the previous chapter, we proposed a long short-term memory (LSTM)-based auto-encoder network as input scheme which showed promising improvements to the performance of the classification network under limited training data [115]. This chapter explores a novel solution for the first time to characterize risky driving behavior using limited vehicular sensor data while minimizing the false positives which are the primary factors for the low reliability of a detection system. To this end, an attention-based auto-encoder network is proposed to reconstruct and precisely classify driving event data. Specifically, the attention-based neural network performs a self-supervised task to encode behavior characteristics of the input event data as fixed-size vectors. The encoded representations are then used by the decoder network to reconstruct the input signal. The decoder network is capable of outputting accurate synthetic signals and classify the signals through an attention operation which enables the network to gain information about the feature importance of the signals. Extensive experiments are carried out to study the effect of the network internal structure on the characterization task and to enable the maximum network performance potential. Experimental results show that the proposed attention-based neural network model can result in an average accuracy of 0.96 and an F-1 score of 0.92 for all classes of driving events.

4.1 Proposed Solution

In this section, the process flow and the data gathering method are discussed and illustrated in Fig. 4.1. At the beginning of the process, the sensing device, i.e. Raven, starts recording the accelerometer data along three directions. The signals are then sent to the pre-processing module to be distributed to the network. In the pre-processing module of the system, the event signals are sliced into the desired length signal windows. The windowed signals are duplicated and varied by random noise before being sent to the generator module for the offline training process. Alternatively, real-time signals are sent to the characterization module, directly, for the real-time inference process.

The second step is introduced to supplement the training data with more variety and quantity. In this module, the underlying behavioral features of the events are learned from the noisy variants of the input signal utilizing a denoising auto-encoder network [116]. The learned features are then passed through the decoder network to populate the training dataset with synthetically created signals. Further details of the module along with existing experimental proofs of the network's legitimacy are provided in the corresponding section.

Finally, the characterization network is trained with the more robust training dataset created by the generator module. The trained network is then used to infer the live signals and detect risky driving events in real-time. This module consists of three individual networks to extract spatial, temporal, and attention encoding of the input signals, sequentially. Detected events form the output of this module which is sent to the device for safety notification and applications. The purpose of each module in the process flow along with their formulation is explained in detail in individual sections.

4.1.1 System Overview

Our proposal for the DEC module builds on a three-stage model that employs recurrent and attention auto-encoder models to recognize various risky driving behavior in a signal. Fig. 4.2 illustrates an overview of the proposed process flow. Accelerometer signal along three orientation axes of the vehicles is captured and fed into the process flow as the input. In the pre-processing module, the data split into n equally sized windows through a sliding window mechanism. Before training, the data windows are multiplied and fed into the denoising recurrent auto-encoder network with added random noise. The decoder network attempts to reconstruct the original signal, thus creating noise-less variations of the original input windows. Then the reconstructed training data are used to train an attention-based encoder network for event characterization. The characterization module uses $softmax$ output layer to classify the signals under multiple categories.

Inference can seamlessly be performed by passing the unseen signals to the characterization module. A more detailed explanation of each module can be seen in the following sections.

4.1.2 Driving Event Dataset Gathering

The proposed system is implemented and tested on the identical dataset used in the experiments of Section 3.1.1. As mentioned, accelerometer sensor data is collected as input to the network. The sensor is oriented along x, y, and z axes of the vehicle and over variable time span as Comma-separated files using Raven OBD-II sensor kit [114] as illustrated in Fig. 3.2. The collected data is evenly distributed between five distinct driving behaviors, namely: Regular driving (RD), Harsh left lane change (HL), Harsh right lane change (RL), Harsh braking (HB), and aggressive acceleration (AA). It's worth noting that the data was recorded on a number of vehicles with different physical attributes and no filtering or signal processing methods were applied.

4.1.3 Pre-processing Module

Pre-processing module initiates the proposed pipeline and is based on the pre-processing methods that were used in the previous experiment. Multiple raw driving signals of various

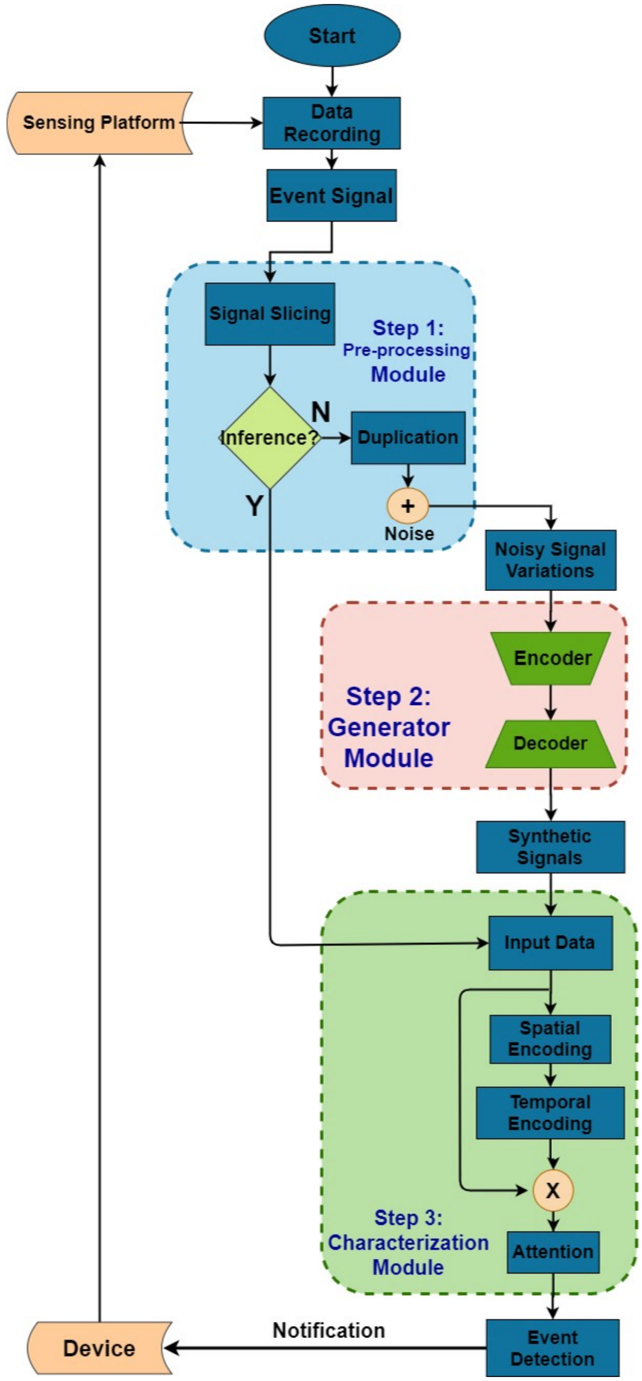

Figure 4.1: Training and inference process flow of the proposed pipeline.

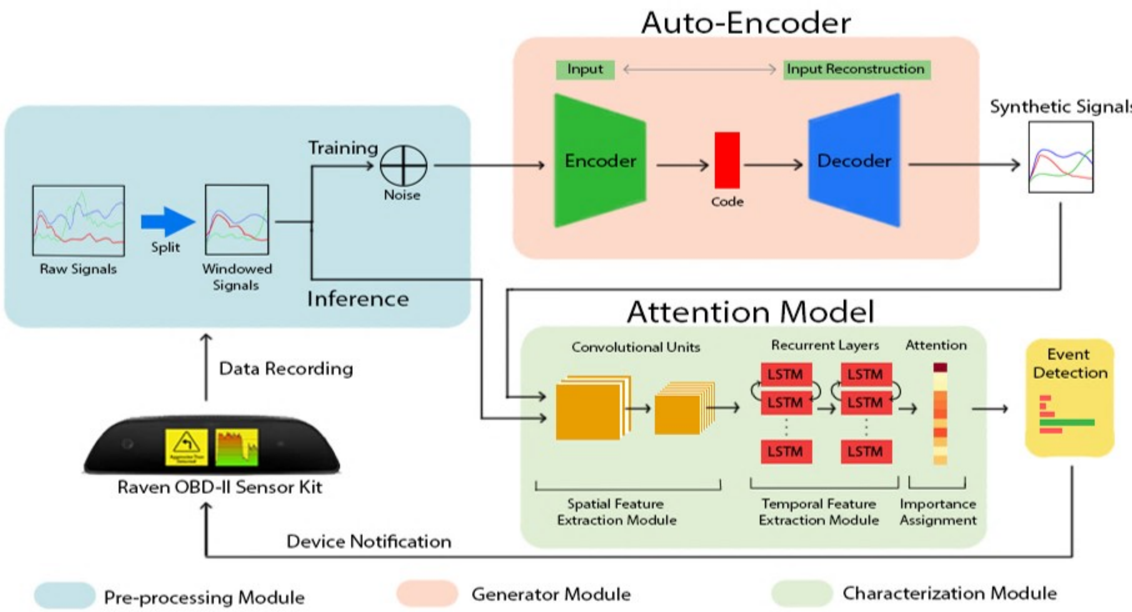

Figure 4.2: Pipeline system overview using the three proposed modules.

time lengths and three axes (i.e. X, Y, and Z) are fed into the module. Since the following modules are designed to take fixed-sized input, a sliding window mechanism splits the data into several event windows of size x. Given a driving event signal of E with duration of t captured along axes $a = 3$ ($E_t^a \in \mathbb{R}^{t \times a}$), the windowing mechanism outputs several evenly split windowed signals $W_n \in \mathbb{R}^{a \times x}$, where W is the signal window, n is the number of created windows, a is the number of axes in the signal, and x denotes input size required by the auto-encoder network. To preserve signal correlations and provide the system with more signal variation, the sliding mechanism [117] splits the signal with overlap as demonstrated in Fig. 3.5. To perform experiments on the result of overlapping section of the data on the model performance, and due to the variable input data length, this module can split data into windows with different overlap values; thus, the overlap between the last two windows is flexible to avoid data loss. The aforementioned overlap value results are presented in the next section.

At the second stage, the pre-processing module identifies the module that the data is going to be streamed through. The training data is sent to the generator module while the inference signals are passed directly to the characterization module. Each training data

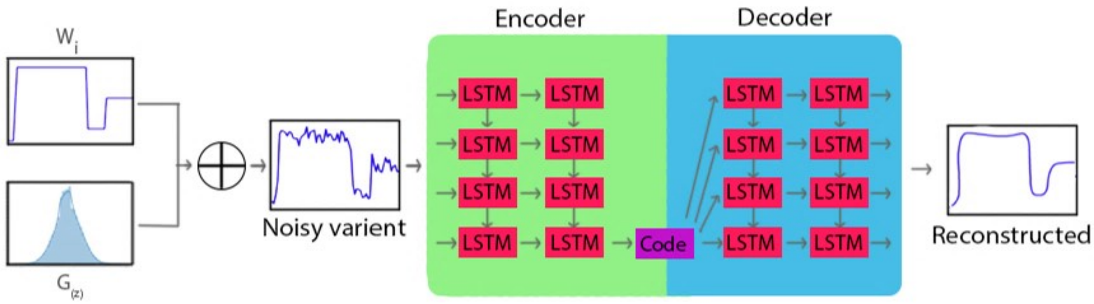

Figure 4.3: Generator module network architecture.

window is duplicated into m varied training data windows with added random Gaussian noise. We then have $W_{nm} \in \mathbb{R}^{a \times x}$, obtained from each input signal window W_n. The data duplication is performed to overcome the issue of scarce risky driving data which often causes data imbalance for deep learning-based event characterization systems. Each copy of the signal is augmented with random noise to gain variations of an event signal. The noisy data windows are then fed into the generator module to initiate denoising before being classified in the characterization module. However, inference data skip this stage and is directly fed to the Characterization module to be identified to the corresponding categories.

4.1.4 Generator Module

The generator module is built based on the previously proposed auto-encoder network. It is responsible to populate the training dataset and extract the underlying characteristics of the signals. Here we introduce, in more detail, the proposed deep neural network utilized in the generator module to not only learn the underlying features of the signals but to also denoise the noisy signals in a vehicular environment. The learned features of the signals by the auto-encoder network allow us to multi-fold the existing training dataset and introduce precise variations of event signals to the dataset. The gained improvements of the method are explained and illustrated in the following sections. The solution adopted here is a recurrent deep auto-encoder network that is suitable for extraction of event behavior in sequential data and is capable of reproducing feature-rich and noise-free data [118]. The objective of such a network is to reconstruct synthetic data learned from the underlying features of input operating as a self-supervised process. The auto-encoder utilized in this module is a denoising auto-encoder that learns to derive the original input from the noisy input. Signals with added noise are fed as input to this type of auto-encoders and the network does not see the original signal. Therefore, the network is not able to predict the output without learning the underlying features of the signals. Denoising auto-encoders take disrupted input and learn to identify its features. High-dimensional data and massive size are often simpler to identify in lower dimensions. Therefore, the network maps the input onto a bottleneck of lower dimension which carries the input features.

The auto-encoder consists of three components: encoder, code, and decoder as illustrated in Fig. 4.3. The encoder maps the input onto a fixed-size context vector of lower dimensionality known as code or the bottleneck. The code is a reduced and compressed representation vector that contains the intrinsic characteristics of the input. The decoder network then pursues reconstruction of the original signal from the noisy signal only when the last node of the code is generated. This component works backward and generates each data point in reverse order, i.e., from the last to the first [99]. Furthermore, auto-encoders can be implemented using a variety of layer and neuron types. While convolutional kernels can be utilized to extract spatial features, recurrent layers that can be employed for temporal feature learning are constructed using feed-forward networks. Since accelerometer sensor data analysis depends on timely features, a recurrent auto-encoder network is chosen as a classifier in this study.

The encoder network of this module is designed with 2 recurrent hidden layers to capture the temporal characteristics of the signals. Though, the conventional recurrent networks lead to the vanishing gradient issue which causes the network to be unable to update its weights and biases properly during the back-propagation [119]. To overcome the issue, this study implements the network with Long short-term memory (LSTM) [120] layers of decreasing sizes that map the input to the bottleneck of size $d = 100$ in the last hidden layer. The symmetrical recurrent decoder network uses the encoded bottleneck vector to reconstruct noise-less signals. Utilizing Adam optimization method [121], the network minimizes the mean squared error (MSE) between each data-point of the hypothesis and the original signal at each batched train as shown in Eq. 4.1:

$$\sum_j \sum_i \left(x^{(j)} - y^{(j)}\right)^2 \tag{4.1}$$

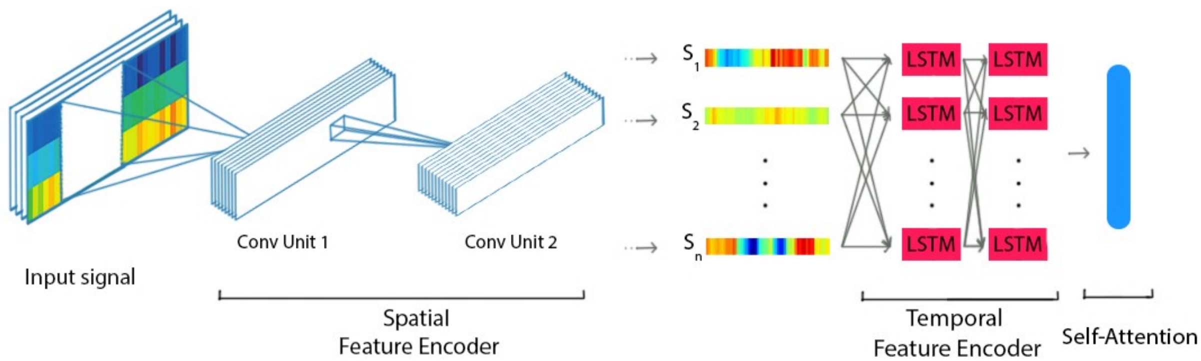

Figure 4.4: Network architecture for the characterization module.

where $j \in \{1, 2, \ldots, t\}$ and $i \in \{1, 2, ..., a\}$ where a is the axis dimension of the input signal and t is the length of the input signal. This module utilizes the Mean Square Error (MSE) loss function as opposed to the cross-entropy loss since the span of sensor measurements exceed the range [0, 1]. Given a sequential input signal of single axis and variable length d: $X = \{x^1, x^2, ..., x^t\}$, the last encoder hidden layer h_{2e} learns of the intrinsic input features : $h_{2e} = \left\{h_{2e}^1, h_{2e}^2, ..., h_{2e}^d\right\}$ where d denotes bottleneck size which is set to 100 in our proposed model. The symmetrical decoding network starts to predict y^t upon generation of the last node in the bottleneck (i.e. h_{2e}^d). This process is performed by predicting from the last state towards the start (i.e. $\{y^t, y^{t-1}, ..., y^1\}$) considering the previous states in each prediction, i.e., $y^t = f(h_{2d}^t, h_{2d}^{t-1})$.

The Generator module runs 10 times for each signal input; Each time with a noisy variation of the input to generate 10 synthetic signals that have the same average statistics with the input but with different values. The synthetic signal generation which shares the same behavior features as the raw input signals grants the possibility to expose the characterization network to higher quantities of training data as well as training data with slight variation. This exposure results in a more robust training process which leads to a higher performance outcome.

It is worth noting that extensive experiments on several recurrent cell types, the number of the hidden layers, and various hyper-parameters of the network are carried out to choose the network details. The impact of all the details on the network performance is presented in the next section.

4.1.5 Characterization Module

The characterization module is used to categorize the unseen signals under the five aforementioned driving events and gets activated to train on the generator module outputs or directly operates to characterize the inference signals as shown in Fig. 4.4.

In the training process, the reconstructed data from the generator module is fed into the characterization module as the input. The module encodes the spatial features of the signals utilizing the custom-sized convolutional filters. Conventional convolutional layers are designed as square-shaped filters to output features along both directions in a 2D input which makes them effective tools on images with observable objects. However, accelerometer signal $E_t^a \in \mathbb{R}^{t \times a}$ is unique two-dimensional data in a sense that the data only has immediate correlation and feature along the axis of time. The other axis consists of signals independent of each other with longer-term relations. Therefore, we implemented a specific convolutional kernel to capture short-term timely features as well as long-term relations between signals of different axes. The convolutional kernel is implemented with the height equal to the number of signal axes a and with stretched kernel width. Doing so lets the convolutional kernel observe a wider body of the signals and therefore have longer-term information. Moreover, the operational nodes in the convolutional network are utilizing Leaky ReLU [122, 123] activation functions which pass a small positive gradient when the unit has a value of zero as Eq 4.2:

$$f\left(x = wx + b\right) = \begin{cases} x & x > 0 \\ 0.01x & otherwise \end{cases} \tag{4.2}$$

where x is a function of weights and biases of the convolutional layer. Using Leaky ReLU allows us to avoid the problem of "dead ReLU" which restricts the network learning from nodes with a derivative of zero. To downsize the extracted features by the convolutional network and store more prominent characteristics, an average pooling layer is applied to the convolutional network output as illustrated in Fig. 4.7.

The second unit of the networks is an attention-based recurrent network to encode temporal dynamics of the signals. The downsized features $S_n = AvgPool(f(x))$ are fed to the Long Short-Term Memory layers. There are two LSTM layers with cell size equal to the number of windowed input of the network. At any moment, the recurrent cells utilize the memory of the previous stages, therefore preserving the temporal history of the signals.

The utilized driving event dataset consists of periods of regular driving along with sudden risky event signals. To lower the false negative predictions, emphasizing the distinct periods that contain the events is necessary. The recurrent network output $r_n = LSTM(S_n)$ is fed into the attention layer for importance assignment to the each input data section. The hidden state of the second LSTM layer feeds the input to the attention layer as the last step before classification. The main purpose of this layer is to weigh out the false detections caused by the dynamics of certain sections in a signal where there is no significant movement information.

As illustrated in Fig. 4.5, each encoded signal section is mapped to a latent space utilizing the following function in Eq. 4.3 where $w_i \in \mathbb{R}^{l \times h_a}$ and $b_i \in \mathbb{R}^{h_a}$ are weight and bias matrix of the direct input to the hidden layer of size h_a.

$$H_i = tanh(w_i h_i' + b_i) \tag{4.3}$$

To learn the importance of each section of the signal, the nonlinear representation H_i is fed to a softmax activation function which is formulated in Eq. 4.4 where v_i denotes to attention vector of each training iteration. The attention vector is learned during the training and is then connected to a softmax layer for final signal classification.

$$V_i = \frac{exp(H_i^T v_i)}{\sum_i exp(H_i^T v_i)} \tag{4.4}$$

The classification layer aims to minimize the cross-entropy error of all labeled sections of the signals, calculated through Eq. 4.5 where p is the network predictions which is a function of network input and y denotes to the target label of the signals.

$$CE = -\sum_c y \; log(p_{(x_n)}) \tag{4.5}$$

The characterization module also utilizes the Adam optimizer function. Adam optimizer is a decaying momentum optimizing method modeled on physical friction [121]. It updates parameters through Eq. 4.6, and stores exponentially decaying average of past squared gradients ($v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$) and an exponentially decaying average of past gradients ($m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$).

$$\Theta_{t+1} = \Theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \tag{4.6}$$

Adam optimizer has a tendency to zero-bias during the initial steps. To fix the bias issue in the gradients, corrected estimates of initial moments are calculated as shown in Equations 4.7 and 4.8.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4.7}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4.8}$$

4.2 Performance Evaluation

In this section, we describe the setting values used in the training process of the proposed model in Section 4.2.1. The detailed experiments regarding the internal structure of the system are provided in section 4.2.2. Last but not least, a comparison of the final result of the pipeline in comparison with state-of-the-art models is presented in section 4.2.3.

Last but not least, randomly selected 70% of the data sessions were split as training and the rest as the testing set. To keep the comparisons fair, the same train and test set were utilized in all our experiments.

4.2.1 Experimental Settings

Initially, ten random Gaussian noise of mean zero and standard deviation of 0.1 by probability distribution are added to each window of data to obtain noisy variations of the input as shown in Eq. 4.9

$$p_G(W_i) = \frac{1}{0.1\sqrt{2\Pi}} e^{-\frac{W_i^2}{2(0.1)^2}} \tag{4.9}$$

The noisy data is then fed into the generator module which is a three-layer stacked denoising auto-encoder network with layer sizes of 600, 300, 100, respectively. The last

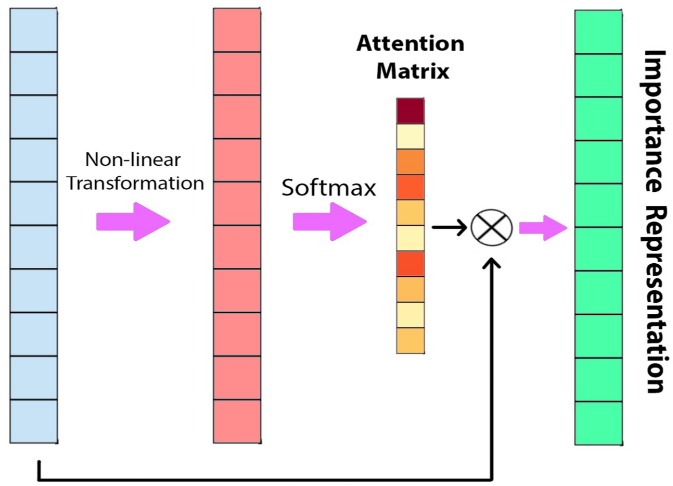

Figure 4.5: Self-attention layer illustration.

encoder layer is used as the vector bottleneck layer. Google's Tensorflow framework was employed to deploy the auto-encoder networks. The decoding network is symmetrical to the encoder i.e. the decoder is implemented with layers of 100, 300, and 600 nodes, respectively. The layers consist of LSTM cells to preserve longer-term memory of the data.

The generator network optimizes its error, calculated by Mean Squared Error (MSE), utilizing Adam optimizer with decay values of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ on each iteration. A learning rate of 0.03 is chosen for network parameter optimization on each training epoch.

The auto-encoder network is trained for 1000 iterations before stopping and the output signals are kept as a training dataset to the characterization encoding network.

The characterization module trains on the reconstructed dataset for 1000 iterations. It optimizes the cross-entropy error using Adam optimizer with decay values of $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-8}$ and the learning rate of 0.03 over the iterations. Dropout layers of probability 0.6 are added to avoid over-training the network. The spatial feature extraction units are operating with custom-sized convolution kernels of size 3×15 and filter depth of 64. The output of the spatial feature extraction unit is a linear code of size $1 \times (n-14) \times 64$, where n is the time length of each signal window. We select no-padding in the convolutional steps. The pooling layers are averaging 30 nodes i.e. filter size of 1×30 on each step with a stride value of 10. The temporal feature extraction network is made of LSTM layers of 128 nodes, feeding the encoded information into a self-attention module of size 256. Lastly, a softmax layer is set as the output layer to classify the features into five event classes.

4.2.2 Internal Structure Investigation

To achieve the third contribution of the thesis and gain knowledge of how the structure and hyper-parameters of the networks affect the performance of the pipeline, extensive experiments were conducted. The experiments are done in a controlled manner to isolate the effect of the subject parameter from other parameters. Moreover, all the experiments are performed 10 times and the average results are presented with 95% confidence levels. In this section, first, the effects of the pre-processing and generator networks are presented and then, we explore the internal structure effects of the characterization network.

As the first experiment, the effect of the overlap percentage in the signal windowing mechanism is investigated. Introducing data overlap significantly improves the overall network performance. The performance boost grows as the overlapping section expands, though it reaches a point of diminishing return after 50%, and values of over that number do not show performance benefits to the process flow (Table 4.1).

Table 4.1: Signal overlap test results.

Overlap Value	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
no-overlap	0.9114 ±0.0108	0.8953 ±0.0127	0.7368 ±0.0131	0.7778 ±0.0129	0.7568 ±0.0134
20%	0.9347 ±0.0140	0.9244 ±0.0153	0.8286 ±0.0138	0.8056 ±0.0136	0.8169 ±0.0137
30%	0.9444 ±0.0128	0.9419 ±0.0131	0.8824 ±0.0129	0.8333 ±0.0126	0.8571 ±0.0125
40%	0.9562 ±0.0117	0.9521 ±0.0106	0.8889 ±0.0108	0.8889 ±0.0131	0.8889 ±0.0136
50%	**0.9704 ±0.0073**	**0.9651 ±0.0089**	**0.9429 ±0.0085**	**0.8919 ±0.0098**	**0.9167 ±0.0096**
60%	0.9859 ±0.0063	0.9593 ±0.0085	0.9167 ±0.0065	0.8919 ±0.0071	0.9041 ±0.0070
70%	0.9837 ±0.0077	0.9588 ±0.0108	0.8824 ±0.0109	0.9091 ±0.0103	0.9055 ±0.0101

To study the generator module in detail, the number of symmetrical and asymmetrical [124] LSTM auto-encoders are deployed and tested on the dataset (Table 4.2). We conclude that symmetrical designs are more consistent and outperform the auto-encoders of

Table 4.2: Generator network depth test results.

Shape	Layers	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Symmetrical	2-2	0.9005 ±0.0196	0.8659 ±0.0217	0.7347 ±0.0164	0.7660 ±0.0172	0.7500 ±0.0170
	3-3	**0.9765 ±0.0073**	**0.9607 ±0.0087**	**0.9403 ±0.0089**	**0.8892 ±0.0106**	**0.9113 ±0.0101**
	4-4	0.9776 ±0.0091	0.9592 ±0.0103	0.9384 ±0.0109	0.8921 ±0.0118	0.9147 ±0.0116
	5-5	0.9604 ±0.0111	0.9603 ±0.0086	0.9224 ±0.0091	0.9008 ±0.0088	0.9114 ±0.0089
Asymmetrical	3-2	0.8654 ±0.0181	0.7733 ±0.0170	0.4884 ±0.0196	0.5526 ±0.0169	0.5185 ±0.0173
	3-4	0.9371 ±0.0139	0.8895 ±0.0140	0.7500 ±0.0138	0.7692 ±0.0148	0.7595 ±0.0141
	3-5	0.9449 ±0.0093	0.9075 ±0.0117	0.8049 ±0.0142	0.8049 ±0.0123	0.8049 ±0.0136

Table 4.3: Generator network node test results.

Node Type	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
ReLU	0.7617 ±0.0164	0.7368 ±0.0183	0.4808 ±0.0149	0.5208 ±0.0181	0.5000 ±0.0173
LSTM	**0.9711 ±0.0087**	**0.9611 ±0.0107**	**0.9415 ±0.0088**	**0.8924 ±0.0106**	**0.9162 ±0.0095**
BiLSTM	0.9703 ±0.0081	0.9592 ±0.0102	0.8734 ±0.0116	0.8828 ±0.0109	0.8780 ±0.0112
GRU	0.9667 ±0.0099	0.9509 ±0.0103	0.8804 ±0.0117	0.8734 ±0.0121	0.8768 ±0.0120

asymmetrical shape. Moreover, while the model performance improves with the number of stacked hidden layers, networks with more than three stacked layers do not show any signs of improvement. As a result of this experiment, a symmetrical network of 3 encoder and 3 decoder layers is chosen for further network investigations.

To test out the effectiveness of several recurrent and non-recurrent neurons, a diverse range of state-of-the-art neurons are selected and the final classification results of the process flow are recorded as presented in Table 4.3. The most superior performance is obtained under the LSTM and BiLSTM nodes. LSTMs are chosen since they reduce the computation complexity of the network compared with BiLSTMs, thus reducing the training time.

Lastly, the dimensionality of the signals is reduced to 2D using Principal Component Analysis (PCA) method. It has been shown that PCA performs poorly when the input data is noisy. To study the usefulness of the generator module in denoising of signals, the reconstructed signals exhibit more separation of values, compared with original signals, in

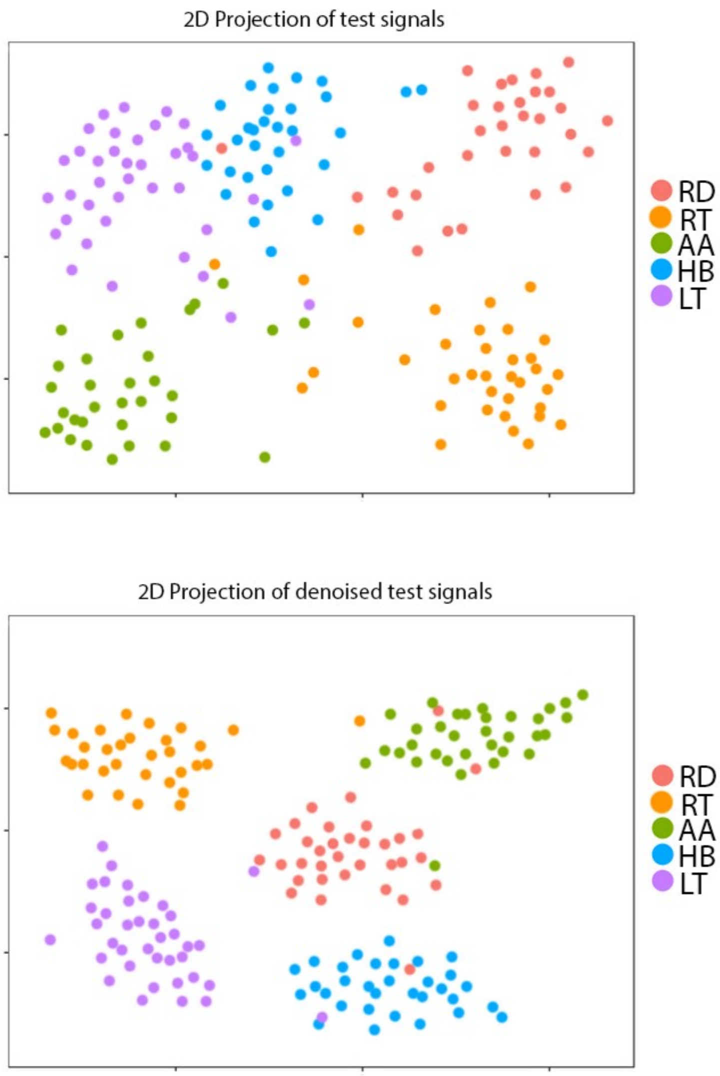

Figure 4.6: 2D projections of the signals clustered after the attention characterization system.

two dimensions which can be described as a better understanding of the underlying features in the signals by the network. The 2D representation of original and reconstructed signals are shown in Figure 4.6.

The first experiment regarding the characterization module layer size is presented in

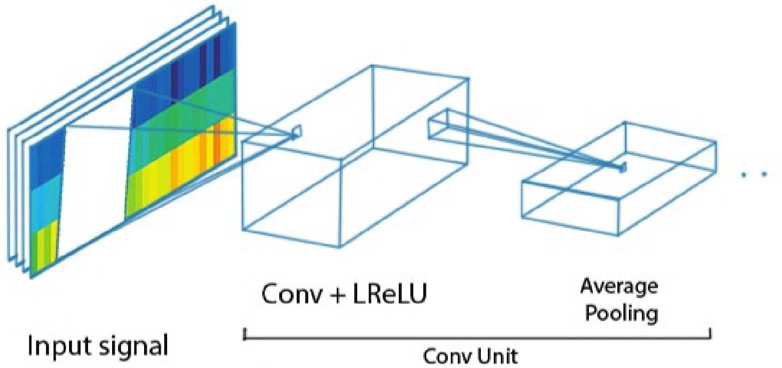

Figure 4.7: Demonstration of the custom-sized convolutional units.

Table 4.4. The characterization encoding network is implemented with one to three spatial feature extraction units as demonstrated in Fig. 4.7 (i.e. a combination of convolutional and pooling layers as a unit). Stacking two spatial feature extraction units is shown to be the most effective. With two convolutional units as default, the number of needed recurrent layers is studied. The experimental results show that stacking two recurrent layers slightly enhances the classification outcome whereas three layers have virtually no positive effect on the performance. Moreover, for the practicality of the attention mechanism in the tests, the module is first trained with no attention layer, and then with an attention layer of different sizes. As reported in the table, the introduction of the attention mechanism has the most significant impact on the performance, though high layer size is shown to lead to the diminishing return of the performance.

As mentioned, a custom-sized convolutional filter is employed that is suitable for vehicular sensor data in the characterization module. The experiments in Table 4.5 are performed on the filter size to find the best fit kernel for the vehicular sensory signals. Additionally, the impact of the convolutional padding and stride on the system performance are investigated. Numerical results concerning the impact of the activation functions in the convolution operations are reported in Table 4.6. The same experiments are repeated on the recurrent neurons in the generator module. Neurons are swapped in the temporal feature extraction unit of the characterization module to record the performance of the system with each neuron type. The top-performing settings are shown in bold in the table.

Table 4.4: Characterization network module test results.

Tested	Dimension	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Conv	1	0.9549 ±0.0086	0.9477 ±0.0119	0.8824 ±0.0106	0.8571 ±0.0119	0.8696 ±0.0112
	2	**0.9741 ±0.0065**	**0.9593 ±0.0086**	**0.9394 ±0.0068**	**0.8611 ±0.0071**	**0.8986 ±0.0069**
	3	0.9703 ±0.0075	0.9535 ±0.0083	0.9257 ±0.0084	0.8857 ±0.0077	0.9052 ±0.0081
LSTM	1	0.9471 ±0.0087	0.9128 ±0.0109	0.8000 ±0.0126	0.7778 ±0.0128	0.7887 ±0.0127
	2	**0.9760±0.0078**	**0.9708 ±0.0084**	**0.9394 ±0.0093**	**0.9118 ±0.0091**	**0.9254 ±0.0090**
	3	0.9732 ±0.0081	0.9649 ±0.0092	0.9116 ±0.0101	0.9121 ±0.0085	0.9118 ±0.0093
Attention	no-attention	0.9214 ±0.0116	0.8837 ±0.0151	0.7273 ±0.0117	0.6857 ±0.0138	0.7059 ±0.0136
	128	0.9624 ±0.0081	0.9591 ±0.0084	0.9091 ±0.0067	0.8824 ±0.0078	0.8955 ±0.0073
	256	**0.9706 ±0.0066**	**0.9649 ±0.0084**	**0.9091 ±0.0074**	**0.9091 ±0.0073**	**0.9091 ±0.0074**
	512	0.9697 ±0.0061	0.9617 ±0.0067	0.9017 ±0.0065	0.8913 ±0.0061	0.8964 ±0.0063

4.2.3 Numerical Results

Ablation studies are carried out to examine the individual component's effectiveness on the performance of the model. Furthermore, several state-of-the-art models, as well as the baseline model, are tested and compared to the proposed model so to demonstrate the superiority of the model in terms of classification accuracy and lower false-positive performance.

The effect of each module in the pipeline is studied with a leave-one-out approach. First, the generator module is disabled and the characterization module is trained and tested on the original dataset to explore the effect the reconstructed signals have on the system performance. Then, spatial feature extraction, temporal feature extraction, and attention modules of the characterization network are abandoned in individual experiments. It is worth noting that the structure and settings of all the components are chosen from the internal structure investigation reported in the previous section. The ablation study results are stated in Table 4.7, in which the generator module is referred to as "G", the spatial and the temporal feature extraction modules of the characterization module as "S" and "T", respectively. Last but not least, the attention module is referred to as "A".

Moreover, state-of-the-art models are trained and tested on our dataset to illustrate a fair comparison of the models. As the baseline, a modified recurrent LSTM driving event detection model [125] is implemented and optimized as described in Chapter 3. The

Table 4.5: Convolution module setting test results.

Test	Setting	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Conv Kernel Size	3x5	0.9592 ±0.0093	0.9379 ±0.0091	0.8732 ±0.0093	0.8591 ±0.0085	0.8660 ±0.0086
	3x10	0.9647 ±0.0086	0.9593 ±0.0084	0.9028 ±0.0072	0.8920 ±0.0064	0.8973 ±0.0066
	3x15	**0.9715 ±0.0102**	**0.9610 ±0.0081**	**0.9048 ±0.0080**	**0.8706 ±0.0068**	**0.8873 ±0.0072**
	3x20	0.9706 ±0.0098	0.9593 ±0.0085	0.9118 ±0.0081	0.8857 ±0.0079	0.8986 ±0.0080
	3x25	0.9712 ±0.0088	0.9603 ±0.0084	0.9103 ±0.0079	0.8920 ±0.0075	0.9010 ±0.0077
Pooling Kernel Size	10	0.9730 ±0.0081	0.9610 ±0.0089	0.9115 ±0.0078	0.8903 ±0.0066	0.9007 ±0.0071
	20	0.9711 ±0.0086	0.9609 ±0.0093	0.9089 ±0.0084	0.8932 ±0.0086	0.9009 ±0.0085
	30	**0.9735 ±0.0078**	**0.9611 ±0.0103**	**0.9048 ±0.0075**	**0.8706 ±0.0071**	**0.8873 ±0.0072**
	40	0.9553 ±0.0102	0.9419 ±0.0094	0.8788 ±0.0105	0.8286 ±0.0113	0.8529 ±0.0109
	50	0.9244 ±0.0124	0.9070 ±0.0133	0.7941 ±0.0119	0.7500 ±0.0141	0.7714 ±0.0138
Conv Padding	**No-Pad**	**0.9711 ±0.0092**	**0.9564 ±0.0103**	**0.8926 ±0.0085**	**0.8742 ±0.0092**	**0.8833 ±0.0088**
	0-Pad	0.9703 ±0.0083	0.9468 ±0.0108	0.8874 ±0.0085	0.8617 ±0.0081	0.8743 ±0.0083
	Same-Pad	0.9694 ±0.0085	0.9573 ±0.0093	0.8894 ±0.0081	0.8723 ±0.0083	0.8807 ±0.0082
Stride	5	0.9701 ±0.0079	0.9588 ±0.0083	0.9010 ±0.0071	0.8988 ±0.0068	0.8998 ±0.0070
	10	**0.9721 ±0.0076**	**0.9593 ±0.0087**	**0.9384 ±0.0091**	**0.8921 ±0.0102**	**0.9147 ±0.0098**
	15	0.9719 ±0.0088	0.9542 ±0.0101	0.9273 ±0.0083	0.8968 ±0.0097	0.9117 ±0.0090
	20	0.9637 ±0.0083	0.9477 ±0.0081	0.8857 ±0.0075	0.8611 ±0.0085	0.8732 ±0.0081

Table 4.6: Module activation function test results.

Module	Nodes	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Convolutional	ReLU	0.9688 ±0.0093	0.9535 ±0.0105	0.8824 ±0.0086	0.8824 ±0.0077	0.8824 ±0.0081
	tanh	0.9664 ±0.0103	0.9477 ±0.0091	0.8857 ±0.0099	0.8611 ±0.0081	0.8732 ±0.0090
	LReLU	**0.9721 ±0.0096**	**0.9590 ±0.0084**	**0.9091 ±0.0091**	**0.8824 ±0.0090**	**0.8955 ±0.0091**
Recurrent	**LSTM**	**0.9703 ±0.0085**	**0.9593 ±0.0098**	**0.9118 ±0.0093**	**0.8857 ±0.0079**	**0.8986 ±0.0085**
	BiLSTM	0.9682 ±0.0087	0.9532 ±0.0095	0.9090 ±0.0079	0.8841 ±0.0086	0.8963 ±0.0082
	GRU	0.9702 ±0.0094	0.9438 ±0.0103	0.8824 ±0.0098	0.8333 ±0.0097	0.8571 ±0.0093

baseline model employs convolutional neural networks as well as recurrent layers as feature extraction methods and trains a fully connected neural network. The classification network of the baseline model attempts to output the probability distribution of the event categories

Table 4.7: Module importance test results.

Network	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Spatial	0.7849 ±0.0144	0.71820 ±0.0182	0.5000 ±0.0140	0.5405 ±0.0136	0.5195 ±0.0138
Temporal	0.7965 ±0.0178	0.7019 ±0.0156	0.5610 ±0.0167	0.5750 ±0.0170	0.5679 ±0.0169
ST	0.8372 ±0.0164	0.7857 ±0.0132	0.5946 ±0.0158	0.6286 ±0.0155	0.6111 ±0.0157
STA	0.9241 ±0.0124	0.9070 ±0.0130	0.7714 ±0.0119	0.7714 ±0.0109	0.7714 ±0.0114
GST	0.8814 ±0.0116	0.8663 ±0.0133	0.6857 ±0.0117	0.6667 ±0.0146	0.6761 ±0.0131
baseline	0.8517 ±0.0115	0.7753 ±0.0135	0.5476 ±0.0151	0.5227 ±0.0143	0.5349 ±0.0147
sparse AE	0.9509 ±0.0101	0.9296 ±0.0117	0.8491 ±0.0088	0.8824 ±0.0096	0.8654 ±0.0092
spatio-temporal	0.8953 ±0.0128	0.8142 ±0.0139	0.6341 ±0.0150	0.6047 ±0.0145	0.6190 ±0.0147
GSTA	**0.9774 ±0.0073**	**0.9608 ±0.0082**	**0.9275 ±0.0079**	**0.9071 ±0.0093**	**0.9171 ±0.0085**

using a softmax layer as the output. This model achieves a test accuracy of above 77% with an F1-score of 0.53 on the dataset. The "Regular Driving" event category of signals are the events that the baseline model detects best with an 83% success rate while facing the worst detection rate of 74% in the "Harsh Left Turn" category. However, the success rate of the proposed model is provided in Fig. 4.8. The system is able to classify the "Harsh Left Turn" signals with only 2 miss-classified signals.

Additionally, the adopted Spatio-temporal classification method [126] that utilizes standard VGG-16 CNN architecture for spatial feature extraction is able to reach 81% in classification accuracy and F1-score of 0.62. This model employs the pooling techniques to gain temporal information across space and time from the data and is end-to-end classification trainable using a classification loss. Furthermore, the adopted sparse auto-encoder classifier network [127] classifies the signals at 93% accuracy and results in a 0.86 F1-score. Last but not least, our proposed system reaches the accuracy of 96% and attains 0.91 F1-score improving 15% upon the baseline.

Lastly, since lowering the false positive predictions is a major focus of ours, a prediction distribution of the test-set, classified by our system is specified in Fig. 4.8 to provide more insight into the efficiency of the model.

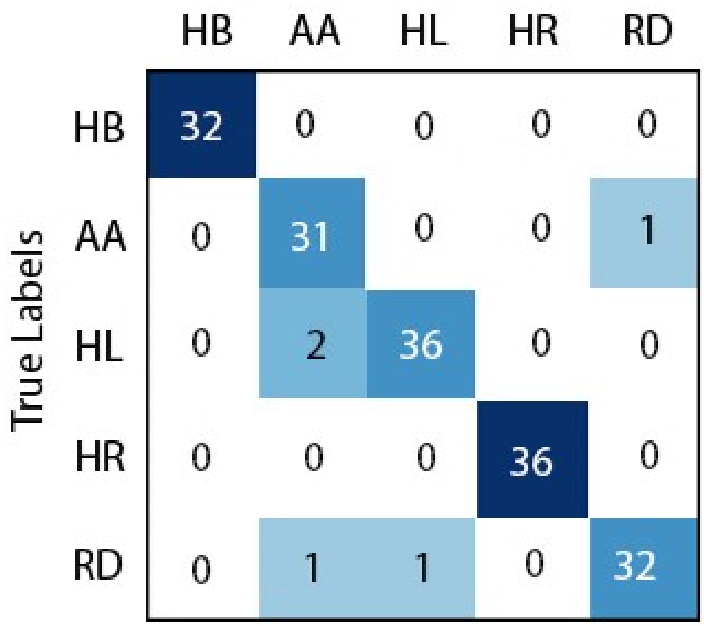

Figure 4.8: Breakdown of the inference of the test-set using the proposed pipeline.

Chapter 5

Knowledge-based Optimization of Driving Event Characterization Models

In this chapter, to optimize the DEC networks and study the effect of knowledge-based modeling, the previously studied and optimized pipeline is chosen as the baseline for the experiments. The main purpose of this chapter is to introduce the Prior Knowledge Input (PKI) modeling to machine learning-based recurrent event characterization models. Knowledge-based modeling has been developed to integrate the existing knowledge into the learning process so further improvement can be made possible through the mapping between existing knowledge and desired responses of the networks. In this chapter, Prior Knowledge Input (PKI) that is one of the knowledge-based methods is considered to obtain the better response (up to 0.96 accuracy) when compared to the results from Chapter 4. PKI utilizes the existing knowledge of the driving events as supplementary input alongside with the sensed input signals. The PKI is modulated to the existing characterization networks as an add-on, which can enable the models to reach the optimal training conditions.

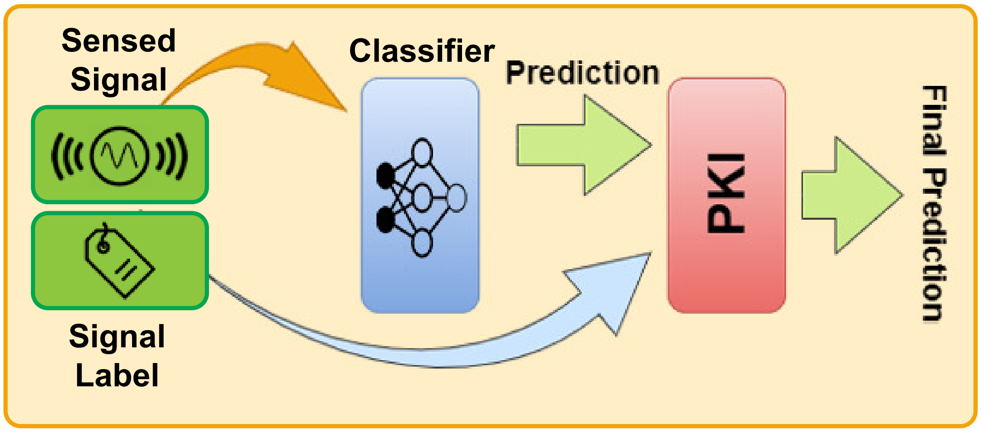

Figure 5.1: Abstract PKI modulation to the classification networks.

5.1 Proposed Solution

In this section, the prior knowledge-based signal classification model which is modulated on top of our signal generation and characterization network, as illustrated in Fig. 5.1, is presented in detail. Machine-learning is modeled to imitate the learning process of the brain, and its classification accuracy and the training process efficiency can be improved by modeling the networks utilizing a knowledge-based modulation. We employ a prior knowledge input method to boost our classification performance which is presented in Section 5.1.3. A brief introduction of the collection process of our dataset and the details of our signal generation model are revisited in Section 5.1.1 and Section 5.1.2, respectively.

5.1.1 Driving Event Data Gathering

Both the signal generation network, as well as the knowledge-based classification network, utilize raw accelerometer sensor data over varied recording sessions. To conduct the experiments while keeping the continuity of the results throughout the thesis, the experiments in this chapter are also performed on the identical dataset that was initially collected. Moreover, the description of the data generation methodology is presented in Section 3.1.1 and illustrated in Fig. 3.2.

5.1.2 Revisiting Synthetic Signal Generation

In an attempt to overcome the data imbalance and infrequency, we previously proposed an auto-encoder model in Chapter 3 which can accurately extract the underlying behavior of the signals and generate precise synthetic signals. The model consists of two modules, namely, a LSTM recurrent encoder-decoder network which is leveraged to populate the training dataset. The second module is used for the classification of the signals and is trained on the synthetic dataset.

To generate the signals, a windowing mechanism runs over each raw signal. Using the windowing mechanism, the raw signals are split into several signal windows of fixed length with overlapping sections. The overlapping is performed to avoid discontinuities in the windowed signals.

The auto-encoder consists of an encoding network that encodes the input signals to vectors of lower dimensions that are feature-rich and contain less noise [128]. Moreover, the encoded vectors are used by the decoding network to reconstruct synthetic signals. The synthetic signal reconstruction is carried out to populate the events in the dataset and to strengthen the classification training process by mitigating data imbalance.

To train the recurrent auto-encoder for synthetic signal generation, the signal windows are infused with randomly generated noise and fed as the input to the auto-encoder net-

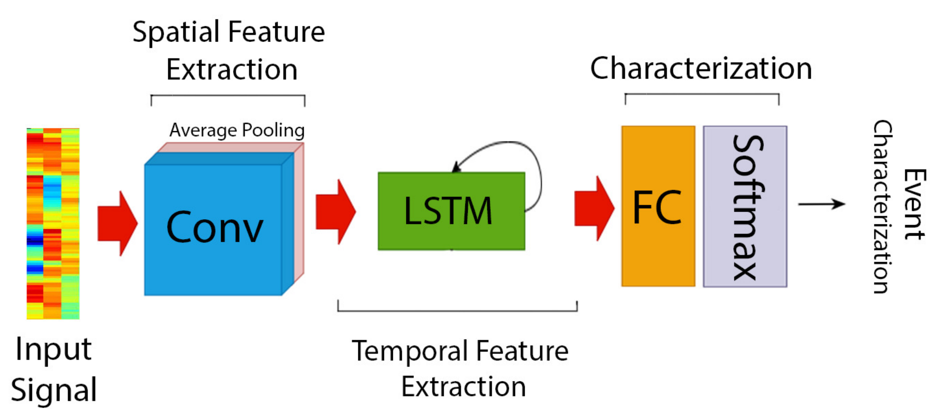

Figure 5.2: Illustration of the baseline convolutional recurrent neural network.

work. The recurrent auto-encoder network consists of a two-layer LSTM network as the encoder which maps the signal windows into fixed-size vectors. It is worth mentioning that the conversion of the signals into lower dimension vectors by the encoder network has the property of signal denoisification since high dimensional data is often more convenient to learn in lower dimensions as it contains less noise. The decoder network attempts to reconstruct the original signals from the vectors. We set the network to generate 10 times synthetic signal windows for the training process of the classification module. In the training process, the auto-encoder network is trained in a self-supervised manner with the intention of minimizing the mean absolute error (5.1) of the original and generated signals.

$$\sum_j \sum_i \left\| x^{(j)} - \hat{x}^{(j)} \right\|^2 \tag{5.1}$$

In order to demonstrate the improvement introduced by the PKI based modeling proposed in this chapter, we trained a feed-forward multi-layer perceptron (MLP) of 3 layers and a convolutional recurrent classification (ConvLSTM) model which adds the Spatio-temporal feature extraction layers to the MLP network. The ConvLSTM model utilizes multiple feature extraction layers of decreasing sizes as illustrated in Fig. 5.2. However, the objective of both classification networks is to categorize the signals into our 5 predefined driving event categories using categorical cross-entropy which is implemented as the softmax (5.2) of cross-entropy function (5.3).

$$f(x) = \frac{e^x}{\sum_C e^x} \tag{5.2}$$

$$CE = -\sum_C t \, log(f(x)) \tag{5.3}$$

5.1.3 Prior Knowledge Input

To boost the performance of the classification networks on the driving event signals, we propose knowledge-based schemes. The Prior Knowledge Input (PKI) method, specifically, is used to incorporate the existing knowledge of the signals into the characterization process. Doing so allows for a more efficient and less complex training process that requires less amount of training data [86] and aids the characterization systems to push past the limited accuracy of deep learning methods and achieve their potential accuracy rates [107]. Moreover, any modeling scheme can be utilized to generate the input knowledge when there is no available prior knowledge of the inputs. Though, neural networks are often the method of choice for knowledge-based modeling [129].

The PKI model embeds the experience/knowledge of the signal category into the training process which allows for the reduction of model complexity through augmentation of the model inputs. The training and testing process of the PKI model is shown in Fig 5.3. The PKI network is implemented in 2 hidden layers that utilize *tanh* activation function with a softmax classification layer to output the event prediction. The training process can be formulated with (5.4) while iterating to lower the margin of the network predictions and the prior knowledge (5.5) for n training iterations. The inference process is also calculated by (5.6) as illustrated in Fig. 5.3.

$$(PKI)_{Train} = \arg\min_{x} \left\| \ldots \quad e_{PKI}^{(n)^T} \quad \ldots \right\| \tag{5.4}$$

$$e_{PKI}^{(i)} = C_{events}(x^{(i)}) - PKI(x^{(i)}, P_c^{(i)}) \tag{5.5}$$

$$P_{PKI} = (PKI)_{Train}(x, P_c) \tag{5.6}$$

5.2 Experimental Evaluation

The synthetic data generation is performed on the training data through the auto-encoder network with 3 LSTM layers with decreasing hidden layer sizes, from 1000 to 300 cells. However, each input signal is first duplicated into 10 individual signals and each is infused with random Gaussian noise. The noisy signals are mapped to vectors of size 300 at the final hidden layer of the encoder network. Subsequently, the decoder network is implemented as a mirrored LSTM network with the same dimensions to the encoder and attempts to reproduce the raw signals from the noisy inputs. Furthermore, the decoder output (i.e. synthetic training data) is used in the training process of the classifier networks. However, in the inference process, the decoder network is deactivated and the encoded vectors are

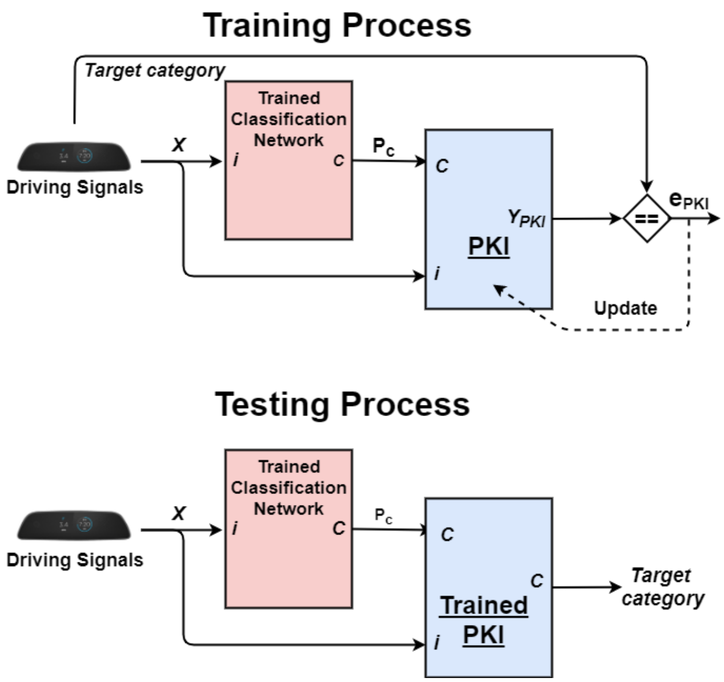

Figure 5.3: Training and testing process of the prior knowledge input method.

directly passed to the classifier network. The process flow of the system is depicted in Fig. 5.4.

To demonstrate the impact of the PKI module, we chose a fully connected feed-forward classifier network, more specifically a Multi-Layer Perceptron (MLP), as a baseline as well as an optimized convolutional recurrent classifier network (ConvLSTM) for further experimentation. Moreover, the classification modules perform a supervised task utilizing categorical cross-entropy for optimization on each training iteration and aim to lower the margin of the predicted events and the target categories. Adam optimizer [121] with training parameter of $\beta_1 = 0.9$ and $\beta_2 = 0.95$ and $\epsilon = 10^{-8}$ is selected for both classifier networks. Lastly, the networks are trained with a learning rate of 0.03 until the stopping is triggered.

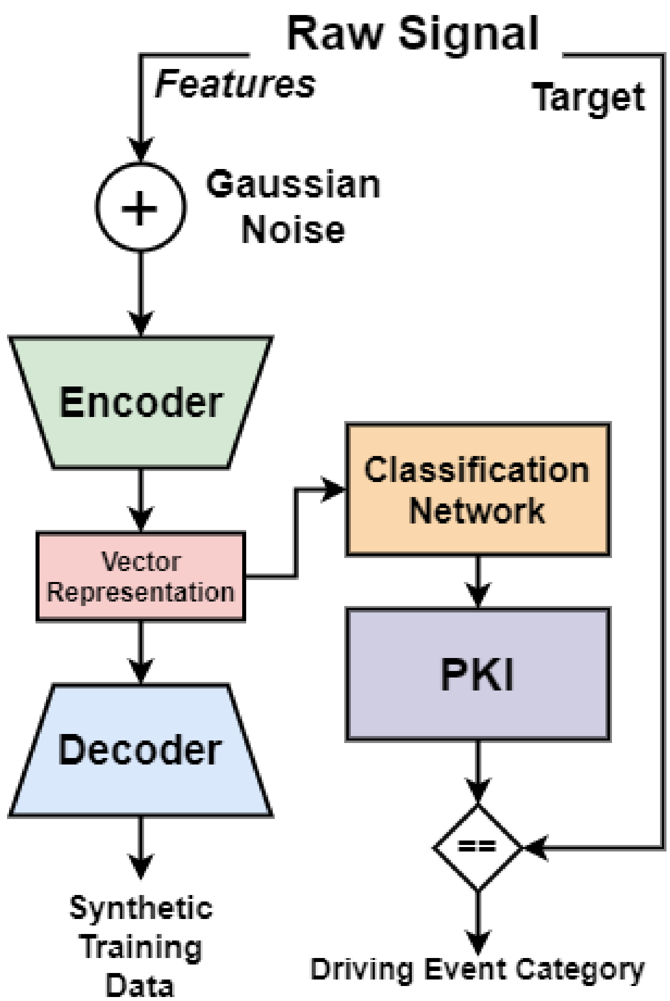

Figure 5.4: Training process flow of the classification pipeline with the PKI method.

5.2.1 Numerical Results

As mentioned earlier, the MLP and ConvLSTM classification networks are trained on our driving event dataset with and without the PKI modulation. Additionally, the networks are trained and tested on raw signals, in the absence of the reconstructed synthetic data, for a more comprehensive demonstration of the PKI benefits. Through testing of the models on an identical test set, the impact of the PKI model is demonstrated in Table 5.1.

Collecting the experimental results, the performance improvement of PKI modulation

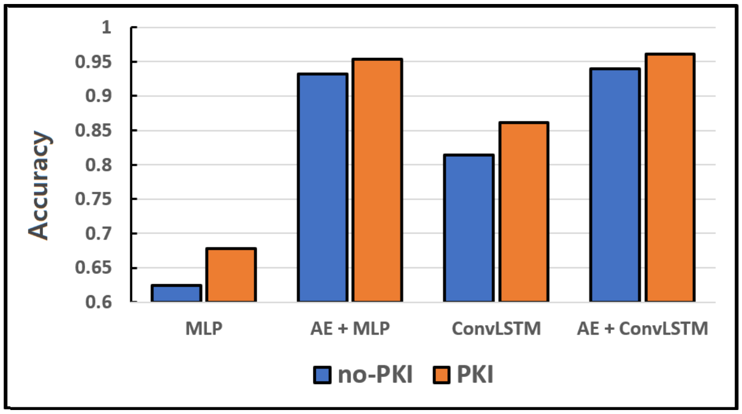

Figure 5.5: Prior Knowledge Input modulation impact comparison.

can be observed across all the tested classification models. The experiments on each classifier network are performed over 10 separate runs for the statistical presentation of the results which are stated with confidence levels of 95%. While the baseline MLP classifier gained the most significant accuracy improvement (by average of over 5%), the PKI modulation proved to increase our most accurate model (i.e. auto-encoder with ConvLSTM classifier) to over 96% average accuracy as shown in Fig. 5.5. Last but not least, as a benefit of PKI accuracy improvements, the decrease in false-positive cases of the models is reflected in improved precision, recall, and therefore F1-score values across the models.

Though the performance boost of the PKI modulation for the most accurate model (i.e. "AE + ConvLSTM") is marginal in comparison to that of the "MLP" model, it is worth noting that the average accuracy increase of 2% is substantial when considering the high accuracy of the end model.

Table 5.1: Final numerical results.

PKI Modulation	Model	Accuracy	Precision	Recall	F1-Score
OFF	MLP	0.624±0.014	0.27±0.011	0.30±0.012	0.29±0.012
OFF	ConvLSTM	0.814±0.009	0.77±0.010	0.79±0.009	0.78±0.009
OFF	AE + MLP	0.932±0.007	0.89±0.008	0.90±0.008	0.89±0.007
OFF	AE + ConvLSTM	0.940±0.004	0.12±0.003	0.11±0.002	0.11±0.003
ON	MLP	0.678±0.009	0.39±0.010	0.44±0.009	0.42±0.009
ON	ConvLSTM	0.861±0.008	0.68±0.008	0.62±0.010	0.65±0.009
ON	AE + MLP	0.954±0.006	0.87±0.008	0.90±0.006	0.88±0.007
ON	AE + ConvLSTM	0.961±0.003	0.87±0.002	0.93±0.002	0.90±0.002

Chapter 6

Conclusion and Future Directions

6.1 Conclusion of the Thesis

The state-of-the-art recurrent neural network models were studied in this thesis to improve the performance of event characterization models to be used in connected vehicle settings. As the first contribution, we proposed a new method to generate precise synthetic data to overcome the challenge of data scarcity for anomalous data points in various driving events. The proposed model builds on an LSTM auto-encoder network to represent noisy vehicular sensor data and to improve the detection performance of anomalous events. Evaluation of the proposed model results in 0.93 accuracy in event characterization, which translates into a 12% improvement in accuracy by reducing the false positive cases when compared to a baseline approach that has been identified as the best performing recurrent neural network model.

As the second part of the work, the rarely studied problem of characterizing the risky driving events in a connected vehicle is tackled. In chapter 4, we proposed a novel neural network-based process flow to not only advance the prediction scores of the existing models but to also address the data scarcity problem in event characterization. The proposed model splits the driving event signals, learns the underlying behavioral features, and de-noises them for the training process. In training, the model extracts Spatio-temporal features of the reconstructed signals, leverages the attention-based neural network, and achieves more than 96% test accuracy which roughly translates to 15% improvement over the baseline. Detailed experiments with regards to the underlying parameters of the networks and their effect on the task performance are also presented to fulfill the third objective.

Ultimately, we investigated a novel approach to boost the classification events that are recognized from vehicular sensors. To do so, the integration of a Prior Knowledge Input (PKI) modeling into the event characterization networks has been proposed to not only improve the overall classification accuracy but also reduce the false detections. The PKI model leverages the existing knowledge of the input signals as a supporting feature to lower the complexity of the model and therefore improve the detection accuracy of the baseline classification networks. The results prove that the performance of the classification models can be significantly improved with the introduction of the PKI modeling. Our results reveal that using the PKI method improves the performance of the baseline MLP classifier by over 5%. Similarly, the ConvLSTM network experiences a 4.7% improvement when coupled with the prior knowledge module while the performance of the auto-encoder with ConvLSTM is also improved by an additional 2%.

6.2 Possible future extensions to the thesis

We have achieved relatively high accuracy results combining the proposed data augmentation network along with the introduced optimization methods to driving event characterization model but we also acknowledge that there is room for improvement in all the three contributions of the thesis, namely auto-encoder synthetic signal generator networks, customized characterization algorithms, and knowledge-based optimization tools. Moreover, other tools and aspects of the DECs such as risky driving datasets or false-detection safety measures are amongst the limitations of the thesis and can benefit from further research and development which are not subject to experimentation.

6.2.1 Data Scarcity

In this study, there is a lack of risky driving event data which pushed us towards working on the first contribution of the thesis. However, high precision driving simulators [130] are now in access which open the doors for novel research on data augmentation modules.

Driving simulators are now advanced enough that they are utilized in the first stages of autonomous driving systems. Additionally, there has been substantial research in the field of data augmentation which have pushed the existing decision making systems to the optimal boundaries. Most recently, there have been ongoing extensions to the task of signal augmentation utilizing GAN networks [131]. With systems that are more capable of understanding the semantic meaning of the existing driving data, incorporating the DEC in real-world would turn into a more straightforward process.

6.2.2 Characterization Systems

For the second stage of the thesis, the experimentation to run the model on a larger dataset where data scarcity is not a challenge is a possible future extension of the thesis in order to study the optimization methods that benefit from big data. The characterization of additional risky behavior classes is also included in our ongoing work which can expand the development opportunities the DECs provide.

Furthermore, there are signs of significant improvement in AI-based signal classification models. Adaptive boosting [132] and other intelligent modelings are now being proposed for signal processing problems. Exploration of the most recent introductions in signal processing field can be beneficial to build the most accurate DEC systems. The future extension of our study should include enhancements and added modulations to achieve higher detection performance.

6.2.3 Optimization

It has been shown that the optimization of the existing model is possible through add-on modules. The modulated approach is effective when optimizing domain-specific issues in characterization systems [133]. The ordinal method to classification problems can be worthy investment in order to further reduce the false-positive detection of the DECs. Utilizing the ordinal networks could allow the systems to overcome the issues each unique vehicle introduces to the decision making process.

The infusion of the sensory-based work in this thesis with our previous vision-based model would also be interesting and challenging work. Multi-modal learning, undeniably, results in better performance for wide variety of deep learning problems [134]. There is room for significant improvement to DECs using different multi-modal learning approaches, such as fusion or reinforcement learning networks. There are existing tools like early or late-fusion of multiple inputs, incremental and several other reinforcement learning models, and stochastic regularization approaches to incorporate learning from multiple models which can be possible extensions to this thesis.

6.2.4 False Detection Control

As discussed in earlier, false detection cases can be fatal in vehicular environments. Additionally, the ultimate goal of smart vehicular safety applications is to reach accuracy levels beyond that of human drivers which is already extremely high. While the characterization systems are improving, control measures [135] are being studied in the literature to mitigate the possible consequences the false detections. Therefore, vehicular safety applications can benefit from further research to integrate the control systems into DECs and be the immediate extension of the work presented in this thesis.

References

[1] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," 1998.

[2] C. Ryan, F. Murphy, and M. Mullins, "End-to-end autonomous driving risk analysis: A behavioural anomaly detection approach," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[3] H. Wymeersch, G. Seco-Granados, G. Destino, D. Dardari, and F. Tufvesson, "5G mmwave positioning for vehicular networks," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 80–86, 2017.

[4] S. Chen, F. Qin, B. Hu, X. Li, and Z. Chen, "User-centric ultra-dense networks for 5G: challenges, methodologies, and directions," *IEEE Wireless Communications*, vol. 23, no. 2, pp. 78–85, 2016.

[5] X. Ge, H. Cheng, G. Mao, Y. Yang, and S. Tu, "Vehicular communications for 5G cooperative small-cell networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7882–7894, 2016.

[6] X. Ge, Z. Li, and S. Li, "5G software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017.

[7] J. Wang, C. Jiang, K. Zhang, T. Q. Quek, Y. Ren, and L. Hanzo, "Vehicular sensing networks in a smart city: Principles, technologies and applications," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 122–132, 2017.

[8] M. Ferreira, R. Fernandes, H. Conceiçao, P. Gomes, P. M. d'Orey, L. Moreira-Matias, J. Gama, F. Lima, and L. Damas, "Vehicular sensing: Emergence of a massive urban

scanner," in *International Conference on Sensor Systems and Software*, pp. 1–14, Springer, 2012.

[9] B. Sliwa, R. Falkenberg, T. Liebig, J. Pillmann, and C. Wietfeld, "Machine learning based context-predictive car-to-cloud communication using multi-layer connectivity maps for upcoming 5G networks," in *IEEE Vehicular Technology Conference*, pp. 1–7, 2018.

[10] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pp. 593–598, 2014.

[11] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *International Conference on Distributed Computing and Internet Technology*, pp. 19–26, Springer, 2013.

[12] N. Taherifard, M. Simsek, and B. Kantarci, "Bridging connected vehicles with artificial intelligence for smart first responder services," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, 2019.

[13] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things J.*, vol. 1, pp. 289–299, Aug. 2014.

[14] T. A. Ranney, W. R. Garrott, and M. J. Goodman, "NHTSA driver distraction research: Past, present, and future," tech. rep., SAE Technical Paper, 2001.

[15] H. Zhang, F. Li, J. Wang, Z. Wang, L. Shi, J. Zhao, C. Sanín, and E. Szczerbicki, "Adding intelligence to cars using the neural knowledge DNA," *Cybernetics and Systems*, vol. 48, no. 3, pp. 267–273, 2017.

[16] J. Golias and G. Yannis, "Dealing with lack of exposure data in road accident analysis," in *Intl. Conf.: Traffic Safety on Three Continents*, vol. 558, 2001.

[17] I. Hashem, V. Chang, N. Anuar, A. S, I. Yaqoob, A. Gani, E. Ahmed, and H. Chiroma, "The role of big data in smart city," *Intl. J. of Information Management*, vol. 36, Oct. 2016.

[18] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger, and W. Xu, "Use cases, requirements, and design considerations for 5G V2X," *CoRR*, 2017.

[19] K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

[20] N. Kalra and D. G. Groves, *The Enemy of Good: Estimating the Cost of Waiting for Nearly Perfect Automated Vehicles*. Santa Monica, CA: RAND Corporation, 2017.

[21] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, 2017.

[22] N. Taherkhani and S. Pierre, "Centralized and localized data congestion control strategy for vehicular ad hoc networks using a machine learning clustering algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3275–3285, 2016.

[23] J. A. Sanguesa, M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, and C. T. Calafate, "A survey and comparative study of broadcast warning message dissemination schemes for vanets," *Mobile Information Systems*, 2016.

[24] J. Wang, Y. Kong, T. Fu, and J. Stipancic, "The impact of vehicle moving violations and freeway traffic flow on crash risk: An application of plugin development for microsimulation," *PLoS one*, vol. 12, no. 9, 2017.

[25] D. Tian, G. Wu, P. Hao, K. Boriboonsomsin, and M. J. Barth, "Connected vehicle-based lane selection assistance application," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2630–2643, 2018.

[26] G. Jornod, R. Alieiev, A. Kwoczek, and T. Kürner, "Environment-aware communications for cooperative collision avoidance applications," in *IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 588–599, 2018.

[27] M. Shams and V. Rahimi-Movaghar, "Risky driving behaviors in Tehran, Iran," *Traffic injury prevention*, vol. 10, no. 1, pp. 91–94, 2009.

[28] F. A. Silva, A. Boukerche, T. R. B. Silva, L. B. Ruiz, E. Cerqueira, and A. A. Loureiro, "Vehicular networks: A new challenge for content-delivery-based applications," *ACM Comp Surv*, vol. 49/1, pp. 1–29, 2016.

[29] J. M. Scanlon, R. Sherony, and H. C. Gabler, "Models of driver acceleration behavior prior to real-world intersection crashes," *IEEE Trans. on Intel. Transportation Systems*, vol. 19/3, pp. 774–786, 2017.

[30] N. Taherifard, M. Simsek, C. Lascelles, and B. Kantarci, "Attention-based event characterization for scarce vehicular sensing data," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 317–330, 2020.

[31] J. Ryu and J. C. Gerdes, "Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control," *J. Dyn. Sys., Meas., Control*, vol. 126, no. 2, pp. 243–254, 2004.

[32] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *IEEE Intelligent Vehicles Symposium*, pp. 234–239, 2012.

[33] F. Li, H. Zhang, H. Che, and X. Qiu, "Dangerous driving behavior detection using smartphone sensors," in *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1902–1907, 2016.

[34] Y. Li, F. Xue, L. Feng, and Z. Qu, "A driving behavior detection system based on a smartphone's built-in sensor," *International Journal of Communication Systems*, vol. 30, no. 8, p. e3178, 2017.

[35] P. Vavouranakis, S. Panagiotakis, G. Mastorakis, C. X. Mavromoustakis, and J. M. Batalla, "Recognizing driving behaviour using smartphones," in *Beyond the Internet of Things*, pp. 269–299, Springer, 2017.

[36] R. Chhabra, S. Verma, and C. R. Krishna, "Detecting aggressive driving behavior using mobile smartphone," in *Intl. Conf. on Communication, Computing and Networking*, pp. 513–521, Springer, 2019.

[37] C. Ma, X. Dai, J. Zhu, N. Liu, H. Sun, and M. Liu, "Drivingsense: Dangerous driving behavior identification based on smartphone autocalibration," *Mobile Information Systems*, vol. 2017, pp. 1–15, 2017.

[38] H. R. Eftekhari and M. Ghatee, "A similarity-based neuro-fuzzy modeling for driving behavior recognition applying fusion of smartphone sensors," *Journal of Intelligent Transportation Systems*, vol. 23, no. 1, pp. 72–83, 2019.

[39] J. White, C. Thompson, H. Turner, B. Dougherty, and D. C. Schmidt, "Wreckwatch: Automatic traffic accident detection and notification with smartphones," *Mobile Networks and Applications*, vol. 16/3, p. 285, 2011.

[40] N. Kumar, A. Barthwal, D. Acharya, and D. Lohani, "Modeling vehicle fall detection event using internet of things," in *International Symposium on Signal Processing and Intelligent Recognition Systems*, pp. 220–233, Springer, 2019.

[41] G. Lugano, "Virtual assistants and self-driving cars," in *IEEE 15th International Conference on ITS Telecommunications (ITST)*, pp. 1–5, 2017.

[42] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5419–5427, 2018.

[43] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3182–3190, 2015.

[44] B. Maaloul, A. Taleb-Ahmed, S. Niar, N. Harb, and C. Valderrama, "Adaptive video-based algorithm for accident detection on highways," in *IEEE Intl. Symp. on Industrial Embedded Sys.*, pp. 1–6, 2017.

[45] J.-G. Kim, J.-H. Yoo, and J.-C. Koo, "Road and lane detection using stereo camera," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 649–652, 2018.

[46] B. Trăsnea, C. Pozna, and S. M. Grigorescu, "AIBA: An AI model for behavior arbitration in autonomous driving," in *International Conference on Multi-disciplinary Trends in Artificial Intelligence*, pp. 191–203, Springer, 2019.

[47] H. Ben-Younes, É. Zablocki, P. Pérez, and M. Cord, "Driving behavior explanation with multi-level fusion," *arXiv preprint arXiv:2012.04983*, 2020.

[48] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," in *IEEE 15th learning and technology conference (L&T)*, pp. 40–45, 2018.

[49] W.-J. Chang, L.-B. Chen, and K.-Y. Su, "Deepcrash: A deep learning-based internet of vehicles system for head-on and single-vehicle accident detection with emergency notification," *IEEE Access*, vol. 7, pp. 148163–148175, 2019.

[50] R. Patel, V. K. Dabhi, and H. B. Prajapati, "A survey on iot based road traffic surveillance and accident detection system (a smart way to handle traffic and concerned problems)," in *IEEE Innovations in Power and Advanced Computing Technologies (i-PACT)*, pp. 1–7, 2017.

[51] U. Khalil, T. Javid, and A. Nasir, "Automatic road accident detection techniques: A brief survey," in *IEEE International Symposium on Wireless Systems and Networks (ISWSN)*, pp. 1–6, 2017.

[52] T. Pholprasit, W. Choochaiwattana, and C. Saiprasert, "A comparison of driving behaviour prediction algorithm using multi-sensory data on a smartphone," in *IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 1–6, 2015.

[53] S. Arumugam and R. Bhargavi, "A survey on driving behavior analysis in usage based insurance using big data," *Journal of Big Data*, vol. 6, no. 1, pp. 86–106, 2019.

[54] I. Han and K. Yang, "Characteristic analysis for cognition of dangerous driving using automobile black boxes," *International journal of automotive technology*, vol. 10, no. 5, pp. 597–605, 2009.

[55] M. S. Amin, M. A. S. Bhuiyan, M. B. I. Reaz, and S. S. Nasir, "GPS and map matching based vehicle accident detection system," in *IEEE Student Conference on Research and Developement*, pp. 520–523, 2013.

[56] R. Sun, W. Y. Ochieng, and S. Feng, "An integrated solution for lane level irregular driving detection on highways," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 61–79, 2015.

[57] A. S. Zeeman and M. J. Booysen, "Combining speed and acceleration to detect reckless driving in the informal public transport industry," in *Intl. IEEE Conf. on Intelligent Transportation Systems*, pp. 756–761, 2013.

[58] B. Fernandes, V. Gomes, J. Ferreira, and A. Oliveira, “Mobile application for automatic accident detection and multimodal alert,” in *IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2015.

[59] J. Hu, L. Xu, X. He, and W. Meng, “Abnormal driving detection based on normalized driving behavior,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 6645–6652, 2017.

[60] L. Eboli, G. Mazzulla, and G. Pungillo, “Combining speed and acceleration to define car users’ safe or unsafe driving behaviour,” *Transportation research part C: emerging technologies*, vol. 68, pp. 113–125, 2016.

[61] T. Phiboonbanakit, V.-N. Huynh, T. Horanont, and T. Supnithi, “Detecting abnormal behavior in the transportation planning using long short term memories and a contextualized dynamic threshold,” in *ACM International Symposium on Wearable Computers*, pp. 996–1007, 2019.

[62] H.-B. Kang, “Various approaches for driver and driving behavior monitoring: A review,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 616–623, 2013.

[63] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, “Deep learning for time series classification: a review,” *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[64] M. H. Alkinani, W. Z. Khan, and Q. Arshad, “Detecting human driver inattentive and aggressive driving behavior using deep learning: Recent advances, requirements and open challenges,” *IEEE Access*, vol. 8, pp. 105008–105030, 2020.

[65] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, and C. Rother, “Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling,” in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1025–1032, 2017.

[66] L. Li, K. Ota, and M. Dong, “Humanlike driving: Empirical decision-making system for autonomous vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 6814–6823, 2018.

[67] T. Osman, S. S. Psyche, J. S. Ferdous, and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," in *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–7, 2017.

[68] M. Soyturk, K. N. Muhammad, M. N. Avcil, B. Kantarci, and J. Matthews, "From vehicular networks to vehicular clouds in smart cities," in *Smart Cities and Homes*, pp. 149–171, Elsevier, 2016.

[69] A. Boukerche, B. Kantarci, and C. Kaptan, "Towards ensuring the reliability and dependability of vehicular crowd-sensing data in GPS-less location tracking," *Perv.& Mobile Comp.*, vol. 68, p. 101248, 2020.

[70] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Trans. on Intelligent Transportation Sys.*, 2019.

[71] X. He, L. Xu, and Z. Zhang, "Driving behaviour characterisation by using phase-space reconstruction and pre-trained convolutional neural network," *IET Intelligent Transport Systems*, vol. 13, no. 7, pp. 1173–1180, 2019.

[72] X. Rao, F. Lin, Z. Chen, and J. Zhao, "Distracted driving recognition method based on deep convolutional neural network," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–8, 2019.

[73] L. Yang, R. Ma, H. M. Zhang, W. Guan, and S. Jiang, "Driving behavior recognition using eeg data from a simulated car-following experiment," *Accident Analysis & Prevention*, vol. 116, pp. 30–40, 2018.

[74] B. Pradhan and M. Al-Zuhairi, *Modeling Traffic Accident Severity Using Neural Networks and Support Vector Machines*, pp. 111–117. Mar. 2019.

[75] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[76] X. Hou, L. Shen, K. Sun, and G. Qiu, "Deep feature consistent variational autoencoder," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1133–1141, 2017.

[77] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and electronics in agriculture*, vol. 153, pp. 46–53, 2018.

[78] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, p. 1, Feb. 2015.

[79] S. N. Dean and S. A. Walper, "Variational autoencoder for generation of antimicrobial peptides," *ACS omega*, vol. 5, no. 33, pp. 20746–20754, 2020.

[80] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[81] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," *Advances in neural information processing systems*, vol. 26, pp. 190–198, 2013.

[82] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5659–5670, 2015.

[83] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[84] M. Simsek, A. A. Obinikpo, and B. Kantarci, *Deep Learning in Smart Health: Methodologies, Applications, Challenges*, pp. 23–46. Cham: Springer International Publishing, 2020.

[85] N. Chen, F. Liu, C. You, P. Zhou, and Y. Zou, "Adaptive bi-directional attention: Exploring multi-granularity representations for machine reading comprehension," in *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7833–7837, 2021.

[86] M. Simsek, B. Kantarci, and A. Boukerche, "Knowledge-based machine learning boosting for adversarial task detection in mobile crowdsensing," in *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–7, 2020.

[87] H. Suzuki and S. Wakabayashi, "Intent inference of driver deceleration behavior by using unscented kalman filter integrated with conventional artificial neural network model," in *International Conference on Human Interaction and Emerging Technologies*, pp. 129–134, Springer, 2020.

[88] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Y. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[89] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, pp. 1–12, 2018.

[90] K. Smagulova and A. P. James, "A survey on lstm memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.

[91] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[92] H. Salehinejad, J. Baarbe, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *ArXiv*, vol. abs/1801.01078, 2018.

[93] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *IEEE 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 228–233, 2016.

[94] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, and Y. Bengio, "Gated orthogonal recurrent units: On learning to forget," *Neural computation*, vol. 31, no. 4, pp. 765–783, 2019.

[95] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[96] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5306–5314, 2017.

[97] A. Ashfahani, M. Pratama, E. Lughofer, and Y.-S. Ong, "DEVDAN: Deep evolving denoising autoencoder," *Neurocomputing*, vol. 390, pp. 297–314, 2020.

[98] A. Heljakka, A. Solin, and J. Kannala, "Pioneer networks: Progressively growing generative autoencoder," in *Asian Conference on Computer Vision*, pp. 22–38, Springer, 2018.

[99] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," in *Proc. Int. Conf. Mach. Learn. Workshop Anomaly Detect.*, 2016.

[100] A. Sood, M. Simsek, Y. Zhang, and B. Kantarci, "Deep learning-based detection of fake task injection in mobile crowdsensing," in *IEEE Global Conf. on Signal and Information Processing*, pp. 1–5, 2019.

[101] P. Malhotra, V. Tv, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Multi-sensor prognostics using an unsupervised health index based on lstm encoder-decoder," *1st SIGKDD Workshop on Machine Learning for Prognostics and Health Management*, 08 2016.

[102] E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Computers & Security*, vol. 78, pp. 155–173, 2018.

[103] Q. Meng, D. Catchpoole, D. Skillicom, and P. J. Kennedy, "Relational autoencoder for feature extraction," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 364–371, 2017.

[104] S. Zhao, T. H. Nguyen, and B. Ma, "Monaural speech enhancement with complex convolutional block attention module and joint time frequency losses," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6648–6652, 2021.

[105] F. Huang, X. Zhang, C. Li, Z. Li, Y. He, and Z. Zhao, "Multimodal network embedding via attention based multi-view variational autoencoder," in *ACM on International Conference on Multimedia Retrieval*, pp. 108–116, 2018.

[106] T. Bui, D. Cooper, J. Collomosse, M. Bell, A. Green, J. Sheridan, J. Higgins, A. Das, J. R. Keller, and O. Thereaux, "Tamper-proofing video with hierarchical attention

autoencoder hashing on blockchain," *IEEE Transactions on Multimedia*, vol. PP, pp. 1–15, 01 2020.

[107] M. Simsek and A. Aoad, "Efficient reconfigurable microstrip patch antenna modeling exploiting knowledge based artificial neural networks," in *Simulation-Driven Modeling and Optimization* (S. Koziel, L. Leifsson, and X.-S. Yang, eds.), pp. 185–206, Springer International Publishing, 2016.

[108] M. Simsek, Q.-J. Zhang, H. Kabir, Y. Cao, and n. s. Sengor, "The recent developments in microwave design," *Int. J. of Mathematical Modelling and Numerical Optimisation*, vol. 2, pp. 213 – 228, 04 2011.

[109] Y. Ali, A. Sharma, M. M. Haque, Z. Zheng, and M. Saifuzzaman, "The impact of the connected environment on driving behavior and safety: A driving simulator study," *Accident Analysis & Prevention*, vol. 144, p. 105643, 2020.

[110] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan, "Deep automatic license plate recognition system," in *Proc. of the Tenth Indian Conf. on Computer Vision, Graphics and Image Processing*, ICVGIP '16, pp. 6:1–6:8, ACM, 2016.

[111] W. Quan, N. Cheng, P. Jing, G. Liu, and X. S. Shen, "VeData: Promoting AI assisted autonomous vehicles," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, (New York, NY, USA), pp. 771–773, ACM, 2018.

[112] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, and D. Gunopulos, "Approximate embedding-based subsequence matching of time series," in *ACM SIGMOD Intl. Conf. on Management of Data*, pp. 365–378, 2008.

[113] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[114] Raven Connected Inc., "Raven OBD-II sensor kit." https://ravenconnected.com/, 2020.

[115] N. Taherifard, M. Simsek, C. Lascalles, and B. Kantarci, "Machine learning-driven event characterization under scarce vehicular sensing data," in *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2020.

[116] X. Feng, Y. Zhang, and J. Glass, "Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 1759–1763, 2014.

[117] M. N. Azadani and A. Boukerche, "Performance evaluation of driving behavior identification models through can-bus data," in *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2020.

[118] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[119] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[120] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE access*, vol. 6, pp. 1662–1669, 2017.

[121] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR), San Diego, CA, USA*, 2015.

[122] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *ArXiv*, vol. 1505.00853, 2015.

[123] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Citeseer, 2013.

[124] A. Majumdar and A. Tripathi, "Asymmetric stacked autoencoder," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 911–918, 2017.

[125] K. Saleh, M. Hossny, and S. Nahavandi, "Driving behavior classification based on sensor data fusion using lstm recurrent neural networks," in *IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, 2017.

[126] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 971–980, 2017.

[127] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, 2016.

[128] W. Yu, G. Zeng, P. Luo, F. Zhuang, Q. He, and Z. Shi, "Embedding with autoencoder regularization," in *Joint European Conf. on ML and Knowledge Discovery in Databases*, pp. 208–223, Springer, 2013.

[129] M. Simsek, "Efficient neural network modeling of reconfigurable microstrip patch antenna through knowledge-based three-step strategy," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 30, no. 3-4, p. e2160, 2017.

[130] M. Karthaus, E. Wascher, and S. Getzmann, "Effects of visual and acoustic distraction on driving behavior and eeg in young and older car drivers: a driving simulation study," *Frontiers in aging neuroscience*, vol. 10, p. 420, 2018.

[131] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.

[132] H. Al-Hadeethi, S. Abdulla, M. Diykh, R. C. Deo, and J. H. Green, "Adaptive boost LS-SVM classification approach for time-series signal classification in epileptic seizure diagnosis applications," *Expert Systems with Applications*, vol. 161, p. 113676, 2020.

[133] C. W. Yong, K. Teo, B. P. Murphy, Y. C. Hum, Y. K. Tee, K. Xia, and K. W. Lai, "Knee osteoarthritis severity classification with ordinal regression module," *Multimedia Tools and Applications*, pp. 1–13, 2021.

[134] D. Ramachandram and G. W. Taylor, “Deep multimodal learning: A survey on recent advances and trends,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[135] R. Casado-Vara, F. Prieto-Castrillo, and J. M. Corchado, “A game theory approach for cooperative control to improve data quality and false data detection in WSN,” *International Journal of Robust and Nonlinear Control*, vol. 28, no. 16, pp. 5087–5102, 2018.