Driver Behavior Modelling and Risk Profiling Using Large-Scale Naturalistic Driving Data

by

Abdalla Abdelrahman

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
October 2019

Dedication

To my mother Sahar, my father Ibrahim, my wife Tasneem, and my two little daughters Deema and Leena

Abstract

Driver risk profiling is an emerging scheme in the field of Intelligent Transportation Systems (ITS). Conventionally, a risk score of a driver is calculated on a per-trip basis according to the number of harsh braking, hard cornering, aggressive acceleration, and excessive speeding events. Risk scoring in the academic literature and industry has two main limitations. First, risk scoring has been primitively performed based on a pre-assumption on the risk weights of driving behaviors. Second, the conventional method of risk scoring ignores the individual differences between drivers and the variation in their skillfulness levels.

In this thesis, we tackle the first limitation through the utilization of the Strategic Highway Research Program 2 (SHRP2) large-scale Naturalistic Driving Study (NDS) dataset (i.e., the largest of its kind to date) and performed by Virginia Tech Transportation Institute (VTTI) to develop reliable and robust risk scoring functions. We first utilize the behavioral information of more than 3,000 drivers during crash, near-crash and normal driving events to develop a robust machine learning model that is able to predict the driving risk quantified in terms of crash and near-crash events of drivers given their long-term behavioral patterns. A complete driver profiling framework that considers the joint effect of driving behaviors and environment conditions

on driving risk is then proposed and validated. Validation results indicate the robustness of the developed models and framework. Then, a novel safety-based route planner that utilizes the personalized risk profiles of drivers in suggesting individualized routing options is proposed and analysed through a real-world case study that highlights the significance of the proposed route planner.

To tackle the second limitation, we propose a fault inference profiling system in which drivers are profiled based on their individual risk rate. Following the detection of risky events, proposed system can infer the fault contribution of drivers using the time-series radar and vehicular data prior and after risky events. Fault inference is performed through training five customized Hidden Markov Models (HMMs), each representing a behavioral class, on 248 risky events. Promising classification results are obtained and discussed.

Co-Authorship

[1] A. Abdelrahman, N. Abu-Ali, and H. S. Hassanein, "Driver Behavior Classification in Crash and Near-Crash Events Using 100-CAR Naturalistic Data Set ," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference,* pages 1-6, Dec. 2017.

[2] A. Abdelrahman, N. Abu-Ali, and H. S. Hassanein, "On the Effect of Traffic and Road Conditions on the Drivers' Behavior: A Statistical Analysis," in *International Wireless Communications Mobile Computing Conference (IWCMC),* pages 892-897, Jun. 2018.

[3] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "Data-driven Robust Scoring Approach for Driver Profiling Applications," in *IEEE Global Communications Conference(GLOBECOM),* pages 1–6, Dec. 2018.

[4] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "A Cloud-Based Environment-Aware Driver Profiling Framework using Ensemble Supervised Learning Applications," in *IEEE International Conference on Communications (ICC),* pages 1–6, May 2019.

[5] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "iRouteSafe: Personalized Cloud-Based Route Planning Based on Risk Profiles of Drivers," in *IEEE Global*

Communications Conference(GLOBECOM) Workshops, Dec. 2019 (accepted)

[6] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "Robust Data-Driven Framework for Driver Behavior Profiling Using Supervised Machine Learning," in *IEEE Transactions on Intelligent Transportation Systems,* (accepted)

[7] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "Towards Robust Environment-Aware Driver Profiling Using Ensemble Supervised Learning," in *IEEE Transactions on Intelligent Transportation Systems,* (Submitted)

Acknowledgements

In the name of God, the most gracious, the most merciful. Praises and thanks are due to God who bestowed upon us endless blessings, and the faculties of seeing, thinking, and learning. It is only with His guidance and help that this work was accomplished.

To my awesome wife Tasneem, you are the one who endured the hardships of this journey with me. During the last four years you were always beside me with your support and prayers. Despite all the hardships you were facing yourself, you spared no effort in creating the perfect atmosphere for me to progress. I would have never completed this work without your encouragement, patience and support. I love you from all my heart.

My sincere gratitude and appreciation are due to my supervisor Professor Hossam S. Hassanein. Your guidance, patience, and support throughout my PhD journey has led to the completion of this work and my development on personal and academic levels. I will always be indebted to you.

My gratitude and appreciation is extended to my co-supervisor Dr. Najah Abu-Ali. I would like to thank you for your guidance and constructive feedback. It has been a great pleasure working with you, and I hope to stay in touch.

I would like to thank the School of Gradute Studies (SGS) and the Electrical and Computer Engineering (ECE) department at Queen's University for their continuous

support and efforts. Special thanks to Debie Fraser for her help and support since the first day I joined my PhD program. I also would like to express my gratitude and appreciation to Basia Palmer from the School of Computing for her helpful feedback and efforts to address all my concerns.

Many thanks to my friend Mohamed Adel who helped me during my first days in Canada and who has been around throughout my whole PhD journey. I wish you all the success in your PhD program. Many thanks to my friend Fathi Souissi. I am so lucky to have a friend like you. I also like to thank my friend Dr. Yehia Elshater for the beautiful times we spent together.

I would like to thank all the people I was surrounded by during my program. Thanks Dr. Hisham Farahat for the awesome times we spent together which helped relieving the research stress. Many thanks to Dr. Ramy Atawia for bringing joy to our lab with your awesome personality and sense of humor. Thanks Anas Mahmoud for the fruitful and intellectual conversations we had together.

Many thanks to all my friends in the Telecommunication Research Lab (TRL): Amr El-Wakeel, Dr. Wenjie Li, Amir Ibrahim, Ashraf Alkhresheh, Ahmad Nagib, Faria Khandaker, Sara Elsayed, Galal Hassan, Samad Razaghzadeh-Shabestari, and Saadeldin Moustafa.

To my lifetime friend Tariq Fahmy, I was blessed to have someone like you in my life. During the harshest times, one call was enough to relieve all the stress. I wish you and your awesome family all the best.

To my parents Sahar and Ibrahim, my appreciation and gratitude to you is beyond the capacity of words. I owe you everything I achieved and will achieve in life. I will never fulfill your rights no matter what I say or do. May God reward you in this life

and in the hereafter.

Finally, to my beautiful daughter Deema, you brought joy and meaning to my life. I ask God to bless and protect you.

List of Abbreviations

GDP	Gross Domestic Product
RSS	Road Safety Strategy
ITS	Intelligent Transportation Systems
ADAS	Advanced Driver Assistance Systems
PHYD	Pay-How-You-Drive
UBI	Usage-Based-Insurance
ML	Machine Learning
SHRP2	Strategic Highway Research Program 2
OBD	On-Board Diagnostics
FoM	Figure of Merit
GPS	Global Positioning System
TD	Toronto Dominion
RNN	Recurrent Neural Network

SVM Support Vector Machine

ANN Artificial Neural Network

RF Random Forest

CAN Controller Area Network

DTW Dynamic Time Wrapping

VTTI The Virginia Tech Transportation Institute

DAS Data Acquisition System

IoT Internet of Things

WHO World Health Organization

IoIV Internet of Intelligent Vehicles

PCA Principal Component Analysis

MSE Mean Square Error

KNN K-Nearest Neighbors

ROC Receiver Operating Characteristic

DT Decision Tree

DNN Deep Neural Network

SGD Stochastic Gradient Descent

ELM Extreme Learning Machine

OLS Ordinary Linear Least Squares

AUC The Area Under the Curve

MAE Mean-Absolute Error

IDE Integrated Development Environment

IQR Inter Quartile Range

CEDP Cloud-based Environment-aware Driver Profiling

SRR Short Range Radar

HPC High-Performance Computing

EMWA Exponentially Moving Weighted Average

V2C Vehicle-to-Cloud

NRN National Road Network

LIP Linear Integer Programming

R.O.O. Revised Regulations of Ontario

NHTSA The National Highway Traffic Safety Administration

VDoT Virginia Department of Transportation

EDR Electronic Digital Recorder

LSTM Long-Short-Term-Memory

Contents

Dedication i

Abstract ii

Co-Authorship iv

Acknowledgements vi

List of Abbreviations ix

Contents xii

List of Tables xv

List of Figures xvii

Chapter 1: Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objectives . . . 3
- 1.3 Research Statement and Thesis Contributions . . . 4
- 1.4 Thesis Organization . . . 7

Chapter 2: Background and Overview 8
- 2.1 Driver Profiling . . . 10
  - 2.1.1 Definition and Potential uses . . . 10
  - 2.1.2 Behavior Detection Techniques . . . 13
  - 2.1.3 Risk Scoring and Profiling . . . 15
- 2.2 SHRP2 NDS Dataset . . . 16

Chapter 3: Data-Driven Profiling Based on Behavioral Patterns 19
- 3.1 Introduction . . . 20
- 3.2 Driver Profiling Framework . . . 21
- 3.3 Data Filtering and Pre-processing . . . 25

3.3.1 Feature Engineering . . . 25
3.3.2 Data Filtering . . . 28
3.3.3 Feature Scaling . . . 29
3.3.4 Dependent Variable (output) Selection . . . 30
3.4 Selection and Customization of Algorithms . . . 31
3.4.1 K-Nearest Neighbors (KNN) . . . 32
3.4.2 Support Vector Machines (SVM) . . . 33
3.4.3 Decision Tree (DT) and Random Forest (RF) . . . 34
3.4.4 Deep Neural Networks (DNNs) . . . 35
3.4.5 Extreme Learning Machines (ELMs) . . . 37
3.5 Performance Assessment Metrics . . . 38
3.5.1 Classification Models . . . 38
3.5.2 Regression Models . . . 39
3.6 Results and Discussion . . . 40
3.6.1 Training and Testing Splitting Methodologies . . . 40
3.6.2 Classification results . . . 41
3.6.3 Regression results . . . 45
3.7 Cloud-based Profiling System . . . 51
3.8 Summary . . . 52

Chapter 4: Cloud-Based Environment-Aware Driver Profiling Framework 54
4.1 Introduction . . . 55
4.2 Environment-Aware Profiling Framework . . . 58
4.2.1 Device Level: In-vehicle Behavior Detection . . . 61
4.2.2 Edge/Fog Level: Risk Prediction and Recommendation Modules 64
4.2.3 Cloud Level: Scoring and Profiling Processes . . . 68
4.3 Data Pre-processing and Model Selection . . . 73
4.3.1 Data Pre-processing . . . 74
4.3.2 Model Selection . . . 75
4.4 Performance Evaluation and Discussion . . . 76
4.4.1 Risk Prediction . . . 76
4.4.2 Driver Scoring . . . 79
4.5 Illustrative Example . . . 83
4.6 Summary . . . 84

Chapter 5: iRouteSafe: Personalized Cloud-Based Route Planning Based on Drivers' Risk Profiles 87
5.1 Introduction . . . 88
5.2 Background and Related Work . . . 90

5.3 iRouteSafe: System Architecture . . . 91
5.3.1 Overview . . . 92
5.3.2 Road risk prediction model . . . 94
5.3.3 Individualized drivers' profiles database . . . 95
5.3.4 Per segment risk indexing . . . 96
5.4 Personalized Safety-Based Routing . . . 98
5.5 Case Study . . . 100
5.6 Summary . . . 103

Chapter 6: Profiling Based on Fault Inference During Risky Events 105
6.1 Introduction . . . 105
6.2 Problem Statement . . . 107
6.3 System Overview . . . 108
6.4 Fault determination . . . 110
6.4.1 Conflicts with leading vehicles (type 1) . . . 111
6.4.2 Conflicts with following vehicles (type 2) . . . 113
6.5 Fault Inference Profiling System . . . 118
6.5.1 Notational Conventions . . . 118
6.5.2 Data Filtering and Pre-processing . . . 119
6.5.3 HMM-based Formulation . . . 122
6.6 Results and Discussion . . . 126
6.7 Summary . . . 127

Chapter 7: Conclusions and Future Work 129
7.1 Summary . . . 129
7.2 Future Work . . . 131
7.3 Concluding Remarks . . . 134

Bibliography 136

List of Tables

3.1 Summary of driving behaviors 23
3.2 SVM adopted Hyper-parameters 34
3.3 DT and RF adopted hyper-parameters 35
3.4 DNN adopted hyper-parameters 37
3.5 ELM adopted hyper-parameters 37
3.6 Classification performance results using the general splitting approach 43
3.7 Classification performance results using 10-fold cross-validation 46
3.8 Prediction performance results using general splitting approach 46
3.9 Prediction performance results using 10-fold cross-validation 47
3.10 Comparison between performance results of two RF models using conventional and extended FoMs 47
3.11 Test case for driver 1 49
3.12 Test case for driver 2 51

4.1 Summary of Notations 60
4.2 Risk Severity Levels 67
4.3 Summary of Environmental Conditions 72
4.4 Contingency table for the number of risky and non-risky events . . . 75
4.5 Hyper-parameters of RF Model 76

4.6 Summary of the RF Model Results 79
4.7 Confusion matrix for training set compliance classification 82
4.8 Confusion matrix for validation set compliance classification 83
4.9 An illustrative example of trip scoring for an *sd* using proposed risk scoring system 86

5.2 Optimization parameters of the case study 102
5.1 Features of driving environments 104

6.1 Behavioral classes of an *sd* involved in a conflict with a leading vehicle. 112
6.2 Behavioral classes of an *sd* driver involved in a conflict with a following vehicle. 115
6.3 Summary of Notations 119
6.4 HMM Hyper-parameters 125
6.5 Confusion matrix for classification under type 1 conflicts 126
6.6 Confusion matrix for classification under type 2 conflicts 127

List of Figures

1.1 High-risk driving statistics 2

2.1 Driver behavior classification and scoring. 11

3.1 Block diagram of the adopted data filtering and pre-processing on SHRP2 raw data. 22

3.2 Block diagram of the proposed driver's risk profiling system. 22

3.3 Weighting function of the risk profile. 25

3.4 Histogram distribution for the number of captured events for drivers in the SHRP2 dataset. 29

3.5 An example of a DNN with two hidden layers. 36

3.6 ROC Curves for DT, SVM, DNN, ELM, KNN and RF classifiers. . . 42

3.7 Precision-Recall Curve for DT, SVM, DNN, ELM, KNN and RF classifiers. 43

3.8 Whisker plot for accuracy, F1-score and ROC AUC performances using 10-fold cross-validation. 45

3.9 Whisker plot for MSE, MAE and R^2 performances using 10-fold cross-validation. 48

3.10 Predicted vs. true risk probabilities for a sample of 100 drivers using RF regressor. 49

3.11 RF models' performances using conventional vs. proposed predictors. 50
3.12 Uplink: A driver's smartphone sends the collected OBDII, radar and its inertial measurements to the cloud for processing. Inside the cloud, behaviors are classified using sequence modeling and inputted to the proposed driver scoring model. Downlink: A trip score is issued to the driver on a per-trip basis. 51

4.1 Proposed Cloud-based Environment-aware Driver Profiling Framework. 59
4.2 A single time frame of collecting and offloading data. 63
4.3 The normalized absolute error histogram for the training set using the developed RF risk prediction model. 77
4.4 The normalized absolute error histogram for the validation set using the developed RF risk prediction model. 78
4.5 Whisker plot for the MAE performance of RI using $10-fold$ cross-validation 80
4.6 Whisker plot for the mean absolute event score error using $10-fold$ cross-validation. 83

5.1 iRouteSafe: proposed personalized safety-based route planning system. 92
5.2 Driver profiling update after each driving trip. 95
5.3 Route planning case study in Ontario, Canada. 100
5.4 Road network as a graph. 101

6.1 A conflict with a leading vehicle in a divided roadway. 109
6.2 Proposed Fault Inference System. 110

6.3 A set of observations showing a faulty *sd* during a conflict with leading vehicle . . . 113
6.4 A set of observations showing a skilled *sd* during a conflict with leading vehicle . . . 114
6.5 A set of observations showing a faulty *sd* during a conflict with a following vehicle . . . 116
6.6 A set of observations showing a non-skilled *sd* during a conflict with a following vehicle . . . 117
6.7 A set of observations showing a non-faulty *sd* during a conflict with a following vehicle . . . 118
6.8 HMM-based architecture for fault inference during risky events. . . . 123

Chapter 1

Introduction

1.1 Motivation

According to Canada's Road Safety Strategy (RSS) 2025 statistics [1], approximately 2,000 fatalities and 165,000 injuries ($\sim$ 10,000 serious injuries) occur annually in Canada due to traffic-related accidents. This, in turn, costs society an estimated $ 37 billion each year which is approximately 2.2% of the total Canadian Gross Domestic Product (GDP). According to Canada's RSS 2025, there are key contributing factors that cause collisions. Among these factors, high-risk driving is considered the primary cause of road accidents. High-risk driving refers to certain driving behaviors that are attributed to high collision rate. This includes exceeding speed limits, alcohol and drug impaired driving, distracted driving, aggressive driving, and driving while fatigued. Figure 1.1 depicts the percentage of total collisions in Canada attributed to each of the aforementioned behaviors.

The recent advancements in vehicular sensing and predictive modelling enabled the deployment of various Intelligent Transportation Systems (ITS) applications that have the potential to lower the currently high crash rates. For instance, many car

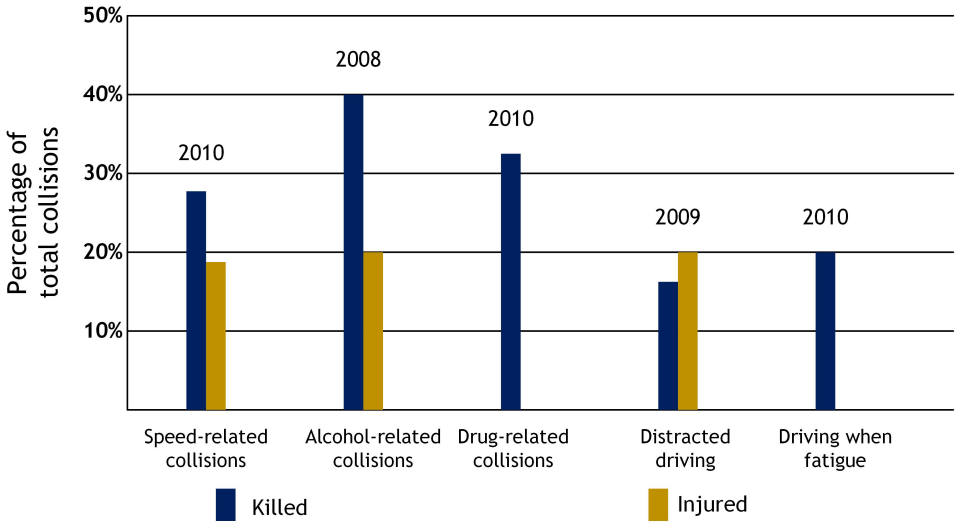


Figure 1.1: High-risk driving statistics

manufacturers are incorporating predictive models in *Advanced Driver Assistance Systems* (ADAS) to predict risky maneuvers before they occur and warn drivers accordingly [2]. Another emerging ITS application is driver risk profiling. Driver risk profiling is based on the detection of risky events after their occurrence and providing drivers with risk scores that reflect their driving behavior.

Driver behavior characterization and profiling has a growing relevance in many fields. For example, in the area of fleet management, fleet administrators are usually interested in keeping track of the driving patterns of their fleet drivers to warn them if unacceptable behavioral attitude is detected. In addition, the classification of drivers based on their driving competencies is used by some insurance companies to adapt the car insurance premiums of drivers by what is known as Pay-How-You-Drive (PHYD) or Usage-Based-Insurance (UBI) [3]. In PHYD, drivers are incentivized through reduced insurance premiums if they are avoiding risky driving maneuvers.

Current risk scoring systems have two main problems. The first is the problem

of *subjectivity* while the second is the problem of *generality*. The *subjectivity* refers to the widely deployed risk scoring functions that pre-assume associated risks for different driving risk factors based on subjective opinions or insufficient data. The problem of *generality*, on the other hand, reflects the fact that current risk scoring functions ignore the differences in the personal driving styles of drivers since such functions are based on absolute behaviors without considering the context of these behaviors. Although some behaviors such as harsh braking and aggressive driving may be attributed to high risk rate from a holistic perspective, they may be safe on a personal level, as drivers vary in their responses to driving situations based on the differences of the personal traits of drivers.

1.2 Objectives

The objectives of the research work presented in this thesis is summarized below:

1. The investigation of data-driven risk prediction and classification models to address the problems of *subjectivity* and *generality* in the current risk scoring functions. This is achieved through the utilization and analysis of large-scale naturalistic driving data-sets.

2. Demystify the terminologies that are used in the context of driver profiling through proposing a detailed driver profiling framework.

3. Proposing a personalized saftey-based route planning system that incorporates personal risk profiles of drivers in the suggesting optimal routes.

1.3 Research Statement and Thesis Contributions

In this thesis, we address the problems of *subjectivity* and *generality* in the current risk scoring functions through the use of large-scale naturalistic driving data-sets. Specifically, we address the following four research questions:

1. Are the long-term driving behavioral patterns good predictors for driving risk?

2. Are driving behaviors together with their environmental context good predictors for measuring risk probability? and how to develop a complete risk profiling framework that takes into account the joint effect of driving behaviors and the environmental context?

3. How to utilize the environment-stamped risk profiles of drivers towards building a personalized safety-based route planner?

4. Can we automatically infer the fault contribution of drivers during risky events?

The first and second questions are targeting the problem of *subjectivity*. The first question aims to provide a data-driven and simplistic approach for profiling drivers on a per-trip basis while considering the predicted risk probability of their behavioral patterns. The second question pushes for providing a complete driver profiling framework that involves predicting the associated risk of driving behaviors coupled with their environmental context to issue "*on the spot*" recommendations/warnings for drivers during their driving trips. The third question is concerned with the utilization of the personal environment-stamped risk profiles of drivers to suggest personalized routes that would minimize their individual driving risk. The fourth question is targeting the problem of *generality* in the current risk profiling functions through adding

another level in the hierarchy of profiling in which drivers are profiled based on the actual risk events they are involved in and their fault contribution in such events.

The contributions of this thesis are summarized as follows:

1. To address the first question, a novel data-driven robust profiling system that assigns risk scores for drivers based on the predicted risk of their behavioral patterns is proposed. In the proposed system, the frequency of each detected behavior during a certain driving trip of a subject driver and the trip length are used through a multi-stage prediction process to forecast the driving risk probability. Different Machine Learning (ML) algorithms are selected in an initial selection phase and their hyper-parameters are optimized for the risk prediction problem. The algorithms are tested on the Strategic Highway Research Program 2 (SHRP2) large-scale Naturalistic Driving Study (NDS) data-set which is the largest data-set of its kind to date. A variety of performance metrics are adopted to reflect the performance of the utilized algorithms. A driving risk score is then assigned as a function of the predicted risk. The proposed system provides a reliable data-driven risk scoring function that can be used in different industrial domains including telematics insurance companies.

2. To address the second research question, we propose a comprehensive driver profiling framework that comprises the different computational stages of the profiling process from the in-vehicle data acquisition to the cloud-based data processing. In this framework, environment-stamped detected behaviors are utilized to build an environment-aware risk profiling database for drivers. A risk profile of a subject driver is computed as a function of the predicted risk of the different environment-stamped behaviors as well as the driver's compliance

to warnings. The performance of the overall scoring system is evaluated using SHRP2 NDS data-set. This system lays the foundation for a novel personalized safety-based route planning system that utilizes the personal risk of drivers in suggesting potential routes.

3. To address the third research question, we propose iRouteSafe, a novel safety-based route optimizer. The proposed optimizer uses environment-stamped risk profiles of drivers to suggest routes based on the individual skill levels of the driver in different driving environments. We use graph theory concepts to define the routing problem which is formulated as a combinatorial joint optimization problem where the objective is to find the optimal route that minimizes cost function composed of a route's travel time, expected risk, and the personal driver-specific risk in such driving routes. We present a real-world case study from Ontario, Canada to highlight the significance of the proposed route planning system.

4. Finally, the last research question is addressed through proposing a Hidden Markov Model (HMM) approach for inferring the fault contribution of drivers during their involvement in risky events. Adding another level in the hierarchy of risk profiling that considers the individual risk rate of drivers and their fault contribution in risky events should mitigate the problem of generality in the current risk profiling systems. Such a problem is attributed to ignoring the personal differences between drivers in dealing with different driving situations. The proposed sequence modelling approach is investigated and results show that this approach can achieve a promising classification accuracy.

1.4 Thesis Organization

In this chapter, we highlighted the motivations of this work, stated the research problems, and discussed our contributions. The remainder of this thesis is organized as follows.

In chapter 2, we provide a background on driver profiling and the state-of-the-art research techniques on behavior detection, risk scoring and profiling. Also an overview on the utilized SHRP2 data-set is provided.

In chapter 3, we propose a robust data-driven profiling approach in which drivers are profiled based on the expected risk of their behavioral patterns. A thorough analysis on different selected and customized supervised ML techniques for the risk prediction problem is conducted. The analysis comprises an initial selection phase for a set of candidate ML algorithms, feature extraction and selection, data filtering, and performance evaluation.

Chapter 4 proposes a complete cloud-based environment-aware driver profiling framework. The framework architecture is discussed followed by an analysis on the performance of the developed prediction model and scoring function.

In chapter 5, a novel safety-based route planner that utilizes the individualized risk profiles of drivers in providing routes that minimize their personal expected risk is investigated. The proposed safety-based optimizer is applied to a real-world routing scenario.

In chapter 6, we propose a fault inference system that classifies the behavior of drivers during risky events using customized HMM models. Models are evaluated using a large scale NDS. Lastly, chapter 7 presents a summary of the work presented in this thesis, the potential future directions and some concluding remarks.

Chapter 2

Background and Overview

The Internet of Things (IoT) is gaining increasing relevance in many applications due to the recent advancements in communications, identification and sensing technology [4, 5]. IoT enables objects to sense and communicate information in real-time which facilitates information exchange, analysis and decision making [6]. According to the Gartner report in [7], it is expected that 20 billion IoT devices will be connected by 2020. This new wave of technology has gained its significance in a wide range of applications such as in smart homes [8, 9], connected wearables [10], and ITS applications [11] including driver risk profiling.

According to the World Health Organization (WHO) global status report on road safety, it is anticipated that road crashes will be the seventh leading cause of death in 2030 unless serious actions are taken [12]. Recently, researchers have been utilizing the Internet of Intelligent Vehicles (IoIV) technology, with attention on ensuring safe driving [13]. IoIV technology refers to the dynamic mobile communication between vehicles (V2V), vehicles and road infrastructures (V2I), vehicles and humans (V2H) or vehicles and cloud (V2C) with the primary objective of minimizing driving risk and ensuring a better driving experience.

Driver risk profiling is an emerging V2C driving application which has particular significance in the fleet management and car insurance telematics domains [14]. In fleet management, fleet administrators are keen on tracking the behavior of their drivers to ensure the safety of their fleets and the roads. Likewise, car companies are adopting the new PHYD insurance paradigm in which insurance premiums are adapted according to the real-time behavior of drivers. In both domains, data that reflect a subject vehicle's (sv) behavior is collected using smartphones' embedded sensors and/or On-Board Diagnostics (OBD or OBDII) units and is then sent to the cloud for analysis. In the cloud, different Figures of Merits (FoMs) are typically calculated for each trip using collected data and a driver's risk score is provided accordingly.

Modeling the actual risk score based on the detected FoMs is viewed by many as an intricate problem. The reason is that the process of designing efficient scoring models necessitates the existence of enough and reliable data, which is not always available. Consequently, different insurance companies have been adopting several scoring models that assign different weights to each FoM [15]. Although several insurers are viewing the number of harsh braking events as the best risk predictor, there is no common agreement about the statistical significance of such measure.

Among the different data collection approaches, NDSs have recently prevailed [16, 17, 18]. NDSs provide researchers with the opportunity to study the behavior of drivers, explore the different driving patterns, and provide data-driven approaches for calculating the risk associated with several driving behaviors [19]. For instance, SHRP2 NDS dataset offers an unprecedented amount of driving context data for almost 9,000 recorded crash and near-crash events and more than 20,000 balanced

base-line events (i.e., normal driving events proportional to the total driving per driver) for more than 3,000 drivers [20]. The collected data gives not only the opportunity to study the prevalence of behavioral factors during risky events but also their prevalence through normal driving episodes, which enables the conduction of statistically sound studies. This dataset is considered by far the largest of its kind. Consequently, the efficient utilization of such dataset can lead to a formulation of more robust driving risk models and can provide more insights into the significance of each risk predictor.

In the next two sections, we first cover the definition of driver risk profiling and the relevant research work. Then, we provide a background on the utilized SHRP2 NDS dataset.

2.1 Driver Profiling

In this section, we first cover the definition and potential uses of driver profiling. Then, we discuss the different behavior detection techniques, risk scoring, and profiling approaches in literature.

2.1.1 Definition and Potential uses

In the literature, the term "driver behavior profiling" has been used to describe different behavioral characterization processes, which may have caused some confusion since some of the literature used "driver profiling" interchangeably with "behavior classification or detection." Although behavior classification is the building block in the driver profiling hierarchy, other processes such as risk scoring and profiling are as important as behavior classification. A complete profiling system that includes

behavior detection, risk scoring, and profiling is still primitively presented in the literature to date.

Driver profiling is based on acquiring a continuous stream of information about the behavior of an *sv* through the use of unobtrusive devices such as OBDII units and smartphones [21]. This data is then processed and classified into driving behaviors which are inputted, along with other FoMs such as trip duration, to a scoring function, as shown in Figure 2.1. A scoring function is a model that can take different forms and assigns weights to the FoMs according to their risk impact [15]. Conventionally, there are four driving behavioral FoMs that are utilized as risk quantification measures (i.e., risk predictors) to calculate a risk score for a certain driver [15, 14]. These FoMs are:

1. **Braking**: number of harsh braking events.
2. **Speeding (relative or absolute)**: number of excessive speeding events whether more than the speed limit or relatively higher than surrounding vehicles.
3. **Cornering**: number of events when turning at a higher than the posted speed.
4. **Acceleration**: number of hard acceleration events.

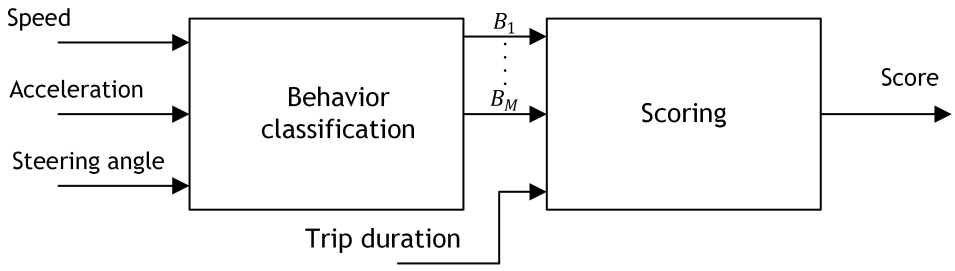


Figure 2.1: Driver behavior classification and scoring.

Several industrial products and research frameworks have been implemented and

proposed. For instance, car insurance companies have developed different smartphone applications that are compatible with IOS and Android operating systems and are capable of detecting and evaluating the behavior of drivers by utilizing smartphones' sensors such as: accelerometers, magnetometers and Global Positioning Systems (GPS). Examples include Toronto Dominion (TD) TDMyAdvantge, Aviva RateMyDrive and State Farm DriverFeedback applications [22, 23, 24]. The aggregated scores over many trips are used to adjust the drivers' car insurance premiums. Drivers with high scores (i.e., safe drivers) are incentivized through receiving a significant reduction in their car insurance premiums.

Research in the field of behavior detection and risk profiling has taken two main directions:

1. Driver behavior detection and classification, this includes the detection of certain events such as aggressive acceleration, and aggressive lane change. [25, 26, 27, 28, 29, 30, 31].

2. The development of risk prediction and scoring functions that accurately reflect the risk rate given the detected behaviors [3, 32, 33].

While the first direction contains many contributions; proposals and frameworks, the second has very few. The choice of scoring functions has been very subjective due to the absence of a frame of reference, which is due to the lack of large-scale and reliable datasets.

Large-scale driving datasets are necessary to develop a reliable data-driven risk prediction model that can infer the statistical dependence between detected behaviors and the expected driving risk (e.g., crash and near-crash probability, where a crash is any contact that the subject vehicle makes with an object, a vehicle, a pedestrian, a

cyclist, or an animal either moving or fixed. Also includes inadvertent departures of the roadway, and a near-crash is any driving conflict that requires an evasive action to avoid a crash). This model is crucial to provide drivers with fair risk scores based on the risk potential of their different behavioral patterns. The developed scoring function can be used within a smartphone application or in a cloud server after the detection/classification of driving behaviors during a specific driving trip.

2.1.2 Behavior Detection Techniques

Driver behavior detection has been extensively researched in the literature. Authors in [34] utilized variations of Recurrent Neural Network (RNN) models to detect seven distinct types of behaviors using smartphone sensors. Authors in [35] evaluated the performance of four static supervised machine learning algorithms such as Support Vector Machine (SVM) and Artificial Neural Network (ANN) in detecting seven different driving maneuvers. Authors concluded that Random Forest (RF) algorithm is superior over other algorithms in the detection of such events. In [36] the authors proposed the "DriveSafe" iPhone application that is capable of detecting drowsy and distracted driving behaviors by utilizing the iPhone's built-in rear-camera, microphone, inertial sensors, and GPS. Authors in [37] utilized the DriveSafe application to provide a large-scale naturalistic driving dataset (UAH-DriveSet) in two road types (i.e., highways and secondary roads). With 500 minutes of publicly available ND data, UAH-DriveSet is expected to facilitate the research in the field of driving behavior detection/classification.

In [28], the authors proposed an HMM-based model to detect abrupt and normal driving maneuvers in both longitudinal and lateral directions. Events were detected

using smartphones, and authors claimed to have a classification accuracy of $\sim 95\%$. Authors in [21] proposed an application called MobiDriveScore that acquires data from a smartphone and a vehicle's network (i.e., Controller Area Network bus (CAN-bus)) to detect risky events. A smartphone application called CarSafe was proposed in [38] to detect dangerous behaviors. Authors utilized smartphones' dual cameras to detect a number of dangerous events. The smartphone was mounted on the dashboard of the car. They used the front camera to detect drowsiness and distraction, whereas the rear camera was utilized to detect tailgating and unintentional lane changes. A fuzzy logic-based smartphone application was proposed in [3]. A driving behavior detection system was proposed and discussed and four unique driving events were detected with high accuracy by fusing the smartphone's accelerometer, gravity, magnetic, and GPS data. Moreover, the authors used two different smartphones with different sampling rates and resolutions and compared their detection performances, which were found to be consistent. Similar work was presented in [39] whereby authors used accelerometer, gyroscope, and magnetometer sensors of a smartphone to detect sharp turning, aggressive acceleration and abrupt lane changing, and sudden braking.A Dynamic Time Warping (DTW) algorithm was implemented to compensate for the varying time of events, and maneuvers were then classified according to their risk level using a binary Bayesian classifier. Other proposals such in [40] aimed to predict the driving behavior at signed intersections using a two-state HMM model.

Other works that are based on advanced discriminative and generative modeling approaches have also been proposed. References [26, 41] propose two algorithms that are based on SVM and HMM to predict the behavior of drivers at intersections. The problem is formulated as a binary classification problem where the output is the driver

being compliant or non-compliant. A large naturalistic data-set is utilized for models training and evaluation, and promising results were obtained. Sathyanarayana et al. in [42] proposed two HMM-based modeling approaches namely, bottom-to-top and top-to-bottom to identify different driving maneuvers. Oliver et al. utilized HMMs and their extension (Coupled HMMs) to classify seven different maneuvers [43]. Other works [44, 45, 46, 47, 48] also used HMMs to identify risky maneuvers or human behaviors.

2.1.3 Risk Scoring and Profiling

Notable research in the context of driver scoring and profiling is the work presented in [3]. Authors in this paper have made a clear distinction between behavior detection and driver profiling. Following the detection of different behaviors, a scoring function was introduced to reflect the overall driving trip score given the detected behaviors. Despite the proposals and findings of the paper, the scoring function was very primitive, since it did not reflect the statistical correlation between actual risk and detected behaviors. Moreover, it did not show how to find an overall driving profile as a function of many trips. In other words, it did not elaborate on how the individual trip scores will be used towards building a driver's profile.

Despite the aforementioned research effort in event detection and driver behavior classification field, contributions in formulating reliable scoring functions are still in their infancy [3], which motivated the formulation of reliable data-driven scoring models presented in this thesis.

2.2 SHRP2 NDS Dataset

Human error contributes to approximately 90% of crashes [49]. In order to examine the influence of different driving behaviors on the crash rate, different approaches have been proposed including the NDS data collection approach [16]. NDS data collection methodology provides three important advantages over other data collection methods [17]:

1. Detailed information about the behavior of a driver prior to a crash or near-crash events.

2. Exposure data, which provides vital information about the frequency of occurrence of different driving behaviors during normal driving episodes.

3. The amount and reliability of collected data allow statistically sound studies to be conducted.

The Virginia Tech Transportation Institute (VTTI) has been pioneering this approach since the beginning of this century with two large-scale data collection projects, the 100-CAR NDS and more recently the SHRP2 NDS. In SHRP2 NDS, 3542 drivers were recruited in six different sites in the United States, and their vehicles were equipped with unobtrusive Data Acquisition Systems (DAS) containing mainly forward radar sensors, video cameras, OBD units to acquire the vehicle's CAN bus information, and GPS. Participants were then asked to use their vehicles in their normal day-to-day driving routine. Data were continuously recorded which resulted in more than 35 million miles of driving data.

Data reductionists were then able to extract almost 9,000 risky events which are comprised of crash and near-crash events. Moreover, normal driving events were

randomly chosen for each driver to offer exposure information. These episodes are called balanced baseline events as their number is balanced with the total driving time of a driver.

The overall raw data contains detailed information of more than 29,000 driving events. Detailed information includes behaviors that are apparent within seconds before risky events or during captured normal driving episodes. Behaviors in the context of this work are different from the *in-vehicle* distractions. They are *vehicle-kinematic* observations that can be noticed from outside the vehicle such as aggressive driving and speeding. In addition to driving behaviors, SHRP2 NDS has the environmental contextual information at which these behaviors happened. Environmental information can be categorized into three types:

1. Static: This refers to long-term environmental features, such as road curvature, number of lanes, traffic flow direction, etc.

2. Quasi-Static: Environmental features that slowly change over the course of time. Road lighting is an example.

3. Dynamic: This refers to the environmental features that rapidly change over the course of time. It includes features such as traffic density.

The operational definitions of different event types in SHRP2 NDS can be found in [50, 20] and are briefly described as follows:

1. Crash: Any contact that the subject vehicle makes with an object, a vehicle, a pedestrian, a cyclist, or an animal either moving or fixed. Also includes inadvertent departures of the roadway.

2. Near-Crash: Any driving conflict that requires an evasive action to avoid a crash.

3. Crash-Relevant: Any driving conflict that requires a non-rapid evasive maneuver.

4. Non-subject Conflict: Any risky event, captured on video but does not involve the subject vehicle.

5. Balanced Baseline Events: Epochs of data selected to provide exposure information. They are 21 seconds long and their frequency is proportional to the total driving time for each driver.

Chapter 3

Data-Driven Profiling Based on Behavioral Patterns

This chapter presents a robust data-driven framework for calculating driver risk profile measured in terms of the additive inverse of the predicted risk probability. SHRP2 NDS dataset is utilized to build the risk prediction models. Crash and near-crash events are used to quantify riskiness whereas balanced baseline driving events (i.e., events captured during normal day to day driving episodes) are used to reflect total exposure or driving time per driver. Thirteen mutually exclusive behavioral risk predictors are identified, and the feature matrix is formulated. A sensitivity analysis is then performed to find the best number of balanced baseline events below which drivers are filtered out. Different machine learning models are selected, customized, and compared to achieve best risk prediction performance. Finally, the utilization of the proposed prediction model within an envisioned driver profiling cloud-based framework is briefly discussed.

3.1 Introduction

This chapter presents a novel robust data-driven framework for evaluating drivers' risk scores and the incorporation of this framework in a cloud-based driver profiling system.

The main contributions of this chapter can be summarized as follows:

1. We provide a practical, robust data-driven framework for calculating drivers' risk profiles (i.e., aggregated risk scores) as a function of the predicted risk probability of their behavioral patterns. This is achieved by the utilization of the behavioral context information during base-line, crash and near-crash events of SHRP2 dataset.

2. A comparative study between selected and customized machine learning algorithms is performed to determine the best performing algorithm for the risk prediction problem. Algorithms are compared in terms of their average performance and their performance consistency through various testing samples.

In this work, we utilized the information of 1836 crashes of all severity levels which represent $\sim 6\%$ of the overall number of events, 6881 near-crash events which constitute $\sim 24\%$ of the overall number of events, and 20179 baseline events which represent $\sim 70\%$ of the overall number of events. The number of baseline events reflects the total driving time of drivers over the period of SHRP2 study. Baseline events were used in this work to provide a "*snapshot* "of the behavioral patterns of drivers on the long-term. The detailed selection criteria of the number of baseline events per driver can be found in [50]. The dominant driving behaviors prior to crash/near-crash events or during baseline events were extracted and recorded from

the collected SHRP2 data by VTTI data reductionists.

The remainder of this chapter is organized as follows. Section 3.2 presents the proposed risk profiling framework. Also, the mathematical formulation of the risk prediction problem is introduced. In section 3.3, the adopted data filtering and pre-processing processes are discussed. In section 3.4, machine learning algorithms that are utilized to predict riskiness are presented. The selection process of these algorithms and the customization of their hyper-parameters for the presented risk prediction problem is motivated. In section 3.5, performance assessment metrics that are employed to measure the performance of the utilized models are discussed. Results and discussion are then presented in section 3.6. An envisioned cloud-based profiling system based on the developed risk prediction model is highlighted in section 3.7 and a summary is finally presented in section 3.8.

3.2 Driver Profiling Framework

In this section, the mathematical formulation of the proposed driver risk profiling framework is presented. Figure 3.1 depicts the block diagram of the adopted offline data filtering, pre-processing, and risk prediction model selection processes. The figure shows the logical sequence of processes applied to the SHRP2 raw data towards a robust risk prediction for different behavioral patterns. Data filtering and pre-processing process consists of merging the raw SHRP2 contextual driving behaviors to increase their importance, feature and output engineering, and filtering out unrepresentative data, whereas the risk prediction model selection phase is composed of the training, testing, and selection of the risk prediction models. Figure 3.2 shows the online risk profiling process which is composed of the online risk prediction, driver

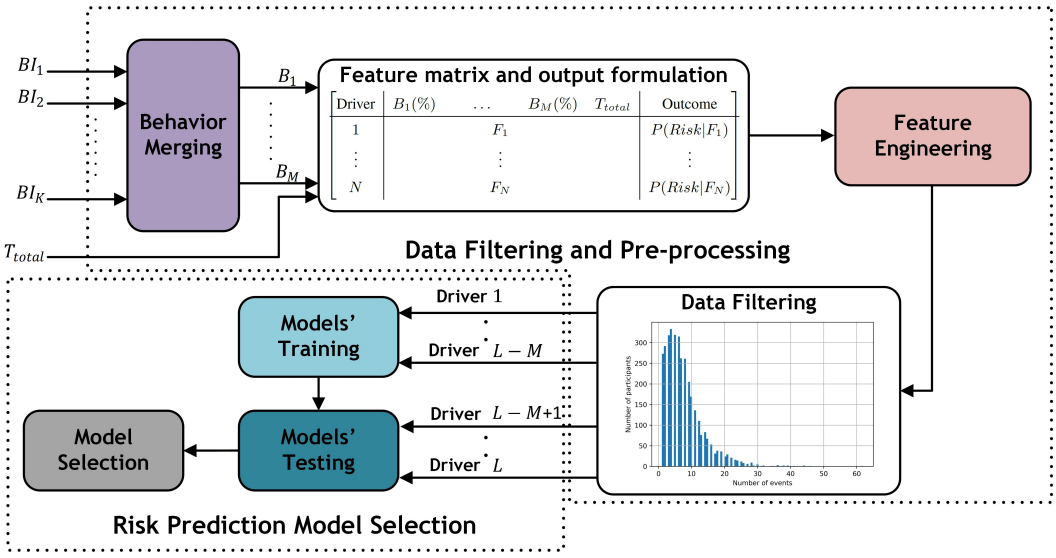


Figure 3.1: Block diagram of the adopted data filtering and pre-processing on SHRP2 raw data.

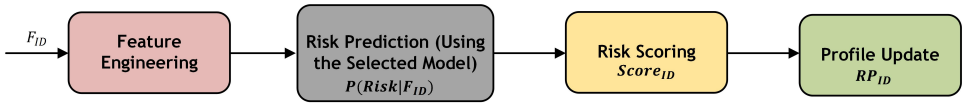


Figure 3.2: Block diagram of the proposed driver's risk profiling system.

scoring and profiling. The specifics of the system's individual components are explained in sections IV and V.

In the proposed framework, the long-term predicted risks of different behavioral patterns are used to reflect the short-term per trip risk scores. To predict the long-term driving risk, each driver is represented by a feature vector denoted by "F_{ID}" which is expressed as:

$$F_{ID} = \begin{bmatrix} B_1(\%) & \dots & B_M(\%) & T_{total} \end{bmatrix} \tag{3.1}$$

where the vector entries "$B_i(\%)$" represent the frequency of occurrence of each identified behavior with respect to other behaviors and T_{total} is a categorical variable that

reflects the total exposure (driving) time for a driver represented here in terms of the total number of base-line driving events. In this work, thirteen mutually exclusive driving behaviors have been identified as risk predictors as will be detailed in the following section. The identified behaviors are depicted in table 3.1 with a brief description of each. The risk prediction is formulated as both a classification and a

Table 3.1: Summary of driving behaviors

Index	Behavior	Description
1	Excessive speeding	Exceeding safe speed/speed limit
2	Inexperience or unfamiliarity	Apparent general inexperience driving, unfamiliarity with a vehicle or a roadway
3	Avoiding an object	Avoiding a vehicle, pedestrian, or an object
4	Sudden braking	Sudden or improper stopping on a roadway
5	Right-of-way error	Right-of-way error due to decision or recognition failures, or an unknown cause
6	Driving slow	Driving slowly in relation to other traffic or below speed limit
7	Improper reversing	Improper backing up due to inattentiveness or other causes
8	Illegal or unsafe lane change or turn	Any improper or illegal lane change or turn
9	Aggressive driving	Such as aggressive acceleration or aggressive lane changing
10	Signal or sign violation	Violation action at traffic signs or signals
11	Safe	No evidence/presence of risky behavior
12	Fatigue	Drowsiness, sleepiness, and fatigue
13	Negligence	Includes improper or failure to signal, and driving past dusk without lights

regression problem as will be discussed in the following section. The risk prediction

is initially defined as the process:

$$\mathcal{F} : F_{ID} \rightarrow P(Risk|F_{ID}) \tag{3.2}$$

where $P(Risk|F_{ID})$ is the probability of driver ID being involved in a risky event given his/her feature vector F_{ID}. $P(Risk|F_{ID})$ is governed by the summation of the crash (C) and near-crash (NC) conditional probabilities as shown in equation 3.3:

$$P(Risk|F_{ID}) = P(C|F_{ID}) + P(NC|F_{ID}) \tag{3.3}$$

These conditional probabilities are expressed herein in terms of crash, near-crash, and captured baseline events for each driver as follows:

$$P(C|F_{ID}) = \frac{NC_{ID}}{NT_{ID}} \tag{3.4}$$

$$P(NC|F_{ID}) = \frac{NNC_{ID}}{NT_{ID}} \tag{3.5}$$

where NC_{ID} and NNC_{ID} are respectively the numbers of recorded crash and near crash events for driver ID, and NT_{ID} represents the total number of recorded events for driver ID. A driver's score is then computed in terms of the additive inverse of $P(Risk|F_{ID})$ as shown in equation 3.6.

$$Score_{ID} = 1 - P(Risk|F_{ID}) \tag{3.6}$$

Practically, scores are calculated for each trip. In this context, a one-to-one mapping between the categorical variable T_{total} and the trip time should be performed. A

risk profile for a certain driver (RP_{ID}) can then be expressed in terms of the weighted average score over the last K trips:

$$RP_{ID} = \sum_{j=T_1-K}^{j=J_1} \alpha_j \times Score_{ID}(j) \tag{3.7}$$

where

$$\sum_{j=T_1-K}^{j=J_1} \alpha_j = 1 \tag{3.8}$$

α_j is the weight associated with the j_{th} trip of the last K trips and can take a shape of an exponentially moving average function to give more weight for recent trips as being depicted in Figure 3.3.

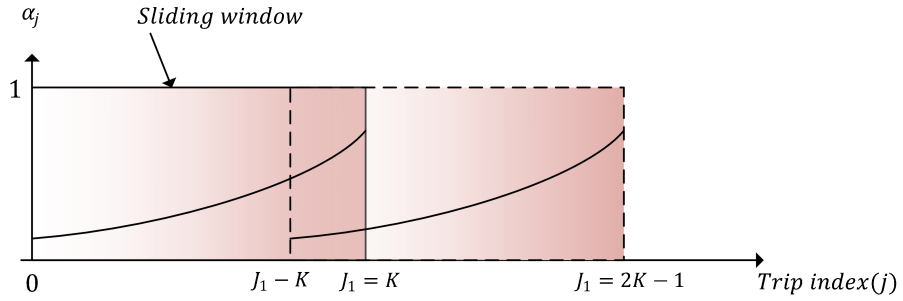


Figure 3.3: Weighting function of the risk profile.

3.3 Data Filtering and Pre-processing

3.3.1 Feature Engineering

As mentioned earlier, thirteen driving behaviors are identified and utilized to extract drivers' feature vectors F_{ID} to train and validate the proposed risk prediction models. Based on the adopted selection criteria, the selected behaviors are comprehensive and mutually exclusive in nature. They are chosen according to the following procedure:

1. In the SHRP2 dataset, driving behaviors are classified into 54 unique behaviors, spanning all possible driving behaviors. In the dataset, the three most identifiable behaviors inside the event time frame are recorded. For simplicity, only the most dominant behavior is chosen, which makes behaviors mutually exclusive for a given event ($P(B_i \cap B_j)_k = 0$, where $P(B_i \cap B_j)_k$ is the probability of the simultaneous occurrence of behaviors B_i and B_j at event k).

2. Behaviors that can be classed under the same category are combined to increase features' importance. Merging behaviors was an iterative process that included a compromise between reducing the models' over-fitting and avoiding the too broad generalization of behaviors resulting from merging too many behaviors in one "general" behavioral category. We initially attributed the over-fitting problem to there being a relatively small number of samples in some of the original behavior categories. Following our behavior merging process, which significantly enhanced the over-fitting performance, such behaviors - due to their rarity in the dataset - were proven to be a cause for over-fitting. At each *behavior merging* iteration, the classification/regression model is tested for over-fitting by comparing the model's train and test performances. As long as the model's performance is improving, additional behaviors with lower number of samples are merged with their corresponding "more general" behavioral categories. The "general" behavioral categories are chosen to avoid overlap and to avoid the broader generalization that makes such behavioral classifications meaningless (e.g., good/bad behaviors). For instance, excessive speeding behavior is clearly distinct from sudden braking, slow driving, improper reversing, etc. An example of merged behaviors is the merging of: "*Driving slowly: below the speed limit*"

and "*Driving slowly in relation to other traffic: not below the speed limit*" behaviors under the general behavioral category of "Driving slow". By following the same procedure for other behaviors, a total of 13 behavioral categories are identified.

The initial training and validating dataset is then formulated as shown in equation 3.9.

$$\left[\begin{array}{c|ccc|c} \text{Driver} & B_1(\%) & \dots & B_M(\%) \quad T_{total} & \text{Outcome} \\ \hline 1 & & F_1 & & P(Risk|F_1) \\ \vdots & & \vdots & & \vdots \\ N & & F_N & & P(Risk|F_N) \end{array}\right] \tag{3.9}$$

The initially formulated features are further processed to enhance the performance of the models. Third-order polynomial non-linear terms of the original features were added to increase model flexibility. Moreover, to capture the interactions between the initially formulated features (i.e., the joint effect of features on risk), features' third order interaction terms were generated. Considering only three original features (f_1, f_2, f_3), their third-order transformation is equivalent to:

$$(1, f_1, f_2, f_3, f_1^2, f_1.f_2, f_1.f_3, \dots, f_1.f_2.f_3) \tag{3.10}$$

With a large number of transformed features (i.e., 680), a feature extraction process was needed to reduce the feature space dimensionality. Such a process was crucial to enhance models' over-fitting performance and to minimize their training/testing processing time. For these reasons, the Principal Component Analysis (PCA) technique

[51] was applied. As a result, the set of features was significantly reduced to a new set of features (often called principal components) that were still able to represent most of the variability in the data.

3.3.2 Data Filtering

Feature engineering process was followed by data filtering. The purpose of the filtering process was to remove cases (i.e. , drivers) which did not have enough data to represent their behavioral patterns (i.e., insufficient number of baseline events). Such cases contributed to the models' irreducible error which is caused by the limitation in the dataset. A sensitivity analysis was applied to find the minimum number of events (E_{best}) a driver should have without being filtered out from the dataset. Threshold values, which represent different numbers of captured events for each driver, in the interval $[4, 10]$ were experimented and models' performances quantified in terms of Mean Square Error (MSE) were recorded for each threshold value. The trade-off was to find the best models' performance in terms of their MSE without losing too much data which can decrease a model's reliability. Having a marginal MSE enhancement in the proposed models' performance with a threshold value greater than 6, an $E_{best} = 6$ was adopted as a filtering criterion. Figure 3.4 depicts the histogram distribution for the number of captured events for all drivers contributed to the SHRP2 project.

After the filtering process, 29% of the cases were excluded. Despite a large number of filtered cases, there were still enough cases (i.e., 2007 cases) for the models' to be trained on and to be able to generalize with a high level of accuracy on test cases as shown in section 3.6. In real-life applications, the rate at which behaviors are detected is supposed to be high enough to represent the behavioral patterns of all

drivers. Detected behaviors during a certain trip will be augmented in the risk scoring function and drivers will be profiled according to the expected risk of their behavioral pattern.

Filtered data are highly skewed to the left as can be deduced from figure 3.4. In section V, different machine learning algorithms are investigated and compared to obtain the best modeling performance for such skewed data.

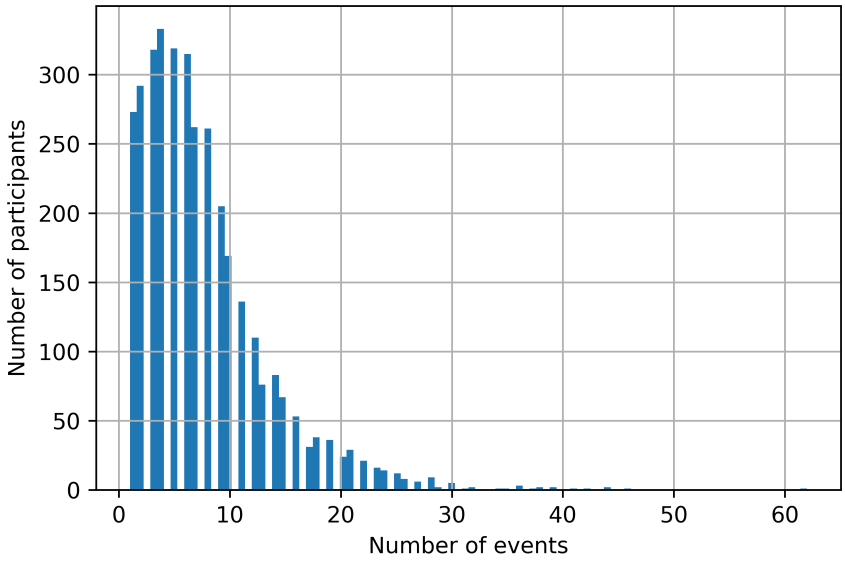


Figure 3.4: Histogram distribution for the number of captured events for drivers in the SHRP2 dataset.

3.3.3 Feature Scaling

Classification bounds for machine learning algorithms such as SVM and KNN are obtained by calculating the Euclidean distance between feature vectors. These algorithms will not work efficiently without feature normalization [52]. This is because if one of the features has a broader range of values than the others, the aforementioned

algorithms will be biased to this specific feature since the minimum distance will be governed by that feature. As a result, it is always a good practice to have the same range of values for all features. In this work, feature normalization was applied to the SVM, KNN, ELM, and ANN based models. The following normalization equation was adopted:

$$\hat{X} = \frac{X - \mu_x}{\sigma_x} \tag{3.11}$$

where X is the raw feature vector, $\hat{X}$ is the normalized feature vector, μ_x is the mean of X, and σ_x is the standard of deviation of X.

3.3.4 Dependent Variable (output) Selection

In this work, the risk prediction problem is formulated using two different approaches. Initially it is formulated as a binary classification problem according to the following expression:

$$Outcome_{ID} = \begin{cases} 1, & \text{if } P(Risk|F_{ID}^{th}) > p_{th} \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

where p_{th} is a threshold risk probability above which the driver is considered risky. The value of p_{th} can vary according to driving risk tolerance. In this work, a value $p_{th} = 0$ is adopted.

The problem is then formulated as a regression problem where $Outcome_{ID}$ takes the soft values of $P(Risk|F_{ID}^{th})$:

$$Outcome_{ID} = P(Risk|F_{ID}^{th}) \tag{3.13}$$

Each of these two risk prediction representations is important according to the domain in which driver profiling is applied. For instance, in the fleet management domain where drivers are warned if their behavior entails risky maneuvers, the binary classification would be more sensible. On the other hand, for insurance applications the adoption of the classification scenario may cause the loss of important information due to the generality that classification entails, since risk scores are averaged over several trips.

3.4 Selection and Customization of Algorithms

In order to tackle the risk prediction problem, a comparative performance study is performed on several selected and customized machine learning algorithms. In this section, we present the six machine learning algorithms selected and the choice of their hyper-parameters. The selection of the candidate algorithms was motivated by two main factors:

1. The non-linearity of the feature space which motivated the sole use of non-linear classifiers/regressors.

2. The inter-dependencies between the risk prediction features. Inter-dependencies are clearly present between the initial behavioral features ($i.e., (B_i(\%))$ because their values are complementary to each other. This occurs because they represent the rate at which each behavior occurs and they add up to one for each driver. So the increase/decrease in one feature will be reflected in the decrease/increase in other features. To show this mathematically, a vector that shows the correlation coefficients between the first and the rest of the features

is displayed below:

$$Corr_{1,i} = [1.0, -0.565, -0.481, \cdots, 0.114] \tag{3.14}$$

Note that the absolute values of the correlation coefficients are larger than zero, which reflects the inter-dependencies between features. This led us to exclude algorithms that assume features' independence such as the Naive Bayes algorithm.

The selected algorithms and the choice of the adopted hyper-parameters are presented next.

3.4.1 K-Nearest Neighbors (KNN)

Despite its simplicity, the KNN algorithm has been successfully applied in several classification and regression-based applications. The algorithm labels a new feature vector by applying a majority voting rule on the labels of its nearest neighboring samples, where neighbors are found by calculating their distances from the new feature vector. [53]. Distance is calculated using different measures such as the Chebyshev distance (L_1-*Norm*), the Euclidean distance (L_2-*Norm*), and more generally the Minkowski distance (L_p-*Norm*). In the context of the proposed framework, for a feature vector $F_{ID} \in \mathbb{R}^{M+1}$, the Minkowski distance between two feature vectors is defined as:

$$D(F_l, F_m) = \left(\sum_{i=1}^{M+1} |F_l(i) - F_m(i)|^p \right)^{(\frac{1}{p})} \tag{3.15}$$

The choice of the optimal number of neighbors (K) as well as the Minkowski distance parameter p depends on the data distribution and the feature space size.

This is usually considered a heuristic optimization problem and is beyond the scope of this chapter. However, for the choice of K, we tested odd values to avoid tied votes [54]. Also, we noticed that performance does not improve for $K > 5$ and a performance degradation occurs for $K > 11$. Consequently, a K value of 5 has been adopted. Concerning the other hyper-parameter p, large values are usually chosen if the feature space is large. Since the feature space of our problem is relatively low (only 13 features), the commonly used L_2-*Norm* distance (i.e., $p = 2$) has been used.

3.4.2 Support Vector Machines (SVM)

SVM is a very popular machine learning algorithm that has been applied in different classification and regression problems. For instance, it has been applied in bioinformatics, road anomalies and driver behavior classification, and a wide range of other applications [26]. The algorithm is based on the margin-maximization principle detailed in [55]. In order to achieve the best classification performance, different hyper-parameters need to be optimized. Grid search technique [56] is adopted in this work to find the best combination of hyper-parameters. We used the area under the Receiver Operating Characteristic (ROC) curve as a performance metric for grid search.

In this work, the optimization is performed over four hyper-parameters which are the regularization parameter C, the kernel function k, the polynomial degree p, and the sensitivity parameter γ. The parameter C is necessary to avoid the overfitting problem. It determines which training samples are considered as outliers. The k parameter specifies the kernel type. For instance, a linear kernel means that SVM will use linear separation hyperplanes. And finally the γ parameter is a sensitivity

parameter to measure the similarity between the feature vectors. For instance, if γ is large, feature vectors will be considered similar only if the Euclidean distance between them is small. A more detailed explanation of these hyper-parameters is found in [57]. Table 3.2 shows the investigated hyper-parameters and the best combinations are shaded.

Table 3.2: SVM adopted Hyper-parameters

Parameter	**Values**				
k	Linear	Polynomial	Gaussian radial basis	-	-
C	1	5	10	50	100
γ	0.01	0.05	0.7	0.1	0.2
p	2	3	4	5	6

3.4.3 Decision Tree (DT) and Random Forest (RF)

Unlike KNN and SVM, DT and RF classifiers (or regressors) do not rely on the minimum distance criterion. A decision tree finds a splitting point on the best predictor's histogram and incrementally builds a tree-structured classifier (or regressor) in a top-down fashion. Decision nodes in each level are chosen such that the entropy is minimized (or equivalently the information gain is maximized). For instance, the topmost decision node (the best predictor node) will have the highest homogeneity as it will maximize the information gain. RFs are very similar to DTs except they use a multitude of decision trees on random subsets of the data to reduce overfitting, which is a common DT problem. DTs and RFs are very intuitive and because of their trackable structure, the importance of each feature can be easily measured, which can be very insightful in many applications, see [58] for detailed information.

To achieve the best risk prediction performance, we optimize the maximum depth (MD) of the decision tree and the number of trees in the random forest estimator (n_estimators). Performance is also calculated in terms of the area under the ROC curve. Table III depicts the values that are used for the two hyper-parameters and the best combination is shaded. The values presented herein are the fine-tuned values of the last round of trials.

Table 3.3: DT and RF adopted hyper-parameters

Decision Tree					
Parameter	*Values*				
MD	12	14	16	18	20
Random Forest					
Parameter	*Values*				
MD	20	22	24	26	28
n_estimators	80	90	100	110	120

3.4.4 Deep Neural Networks (DNNs)

Using multiple sequential computational layers, Deep Neural Networks (DNNs) learn data representation through multiple levels of abstraction [59, 60, 61] as depicted in figure 3.5. Based on the original features from the input layer, each hidden layer creates more complex features based on interactions of features from a previous layer. DNNs do not need feature engineering since features with higher levels of abstraction are naturally extracted during back propagation. Using back-propagation algorithms such as Stochastic Gradient Descent (SGD), ADAM algorithm, RMSProp, Limited memory BFGS, etc., DNNs learn how the internal parameters between every two layers represented by the matrix $W^{(i)}$ should change to minimize a chosen aggregated

loss function $L(W)$, where $W = [W^{(1)}, W^{(2)}, \ldots, W^{(H)}]$ for a DNN with H hidden layers.

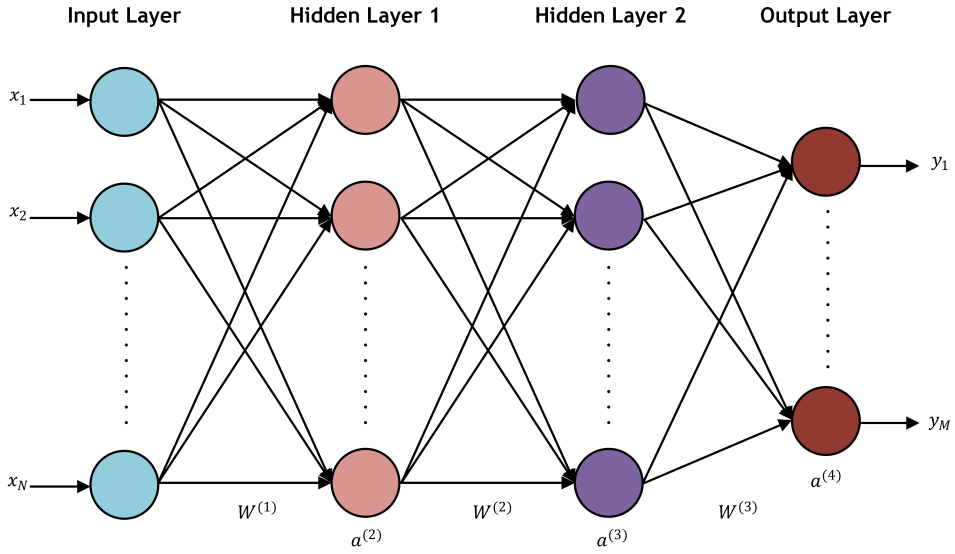


Figure 3.5: An example of a DNN with two hidden layers.

In this work, a customized **feed-forward** DNN was adopted. A feed-forward DNN rather than a convolutional DNN or a recursive DNN was utilized since the targeted problem does not require modeling image data nor a sequential data. The adopted DNN's hyper-parameters include the rate at which the weights are updated at each iteration (learning rate α), Momentum which helps in preventing oscillations around the cost function global minimum, the number of hidden layers, the number of hidden units per layer, the regularization parameter ($L2$ penalty) which helps in preventing over-fitting, the number of epochs, the optimization algorithm for updating the network's weights and the choice of the activation function. Due to the large number of hyper-parameters, Grid search was discarded as it is considered a computationally inefficient hyper-parameters' optimization technique in such cases. Thus, we applied the random search technique [56] to find the optimal set of hyper-parameters

which are displayed in table 3.4. The displayed hyper-parameters resulted in the best performance where we adopted the number of epochs as the stopping criteria.

Table 3.4: DNN adopted hyper-parameters

Parameter	**Adopted Value**
Learning rate (α)	0.001
Momentum	0.9
Number of hidden layers	5
Number of hidden units	5
$L2$ penalty	0.0001
Number of epochs	200
Optimization algorithm	Limitied-memory BFGS
Activation function	RELU

3.4.5 Extreme Learning Machines (ELMs)

A special case of Artificial Neural Networks is the Extreme Learning Machines (ELMs) [62, 63]. An ELM is a single layer feed-forward ANN with a random number of hidden neurons and Ordinary Linear Least Squares (OLS) algorithm applied to find the network weights' matrix through a single optimization step. ELMs take much less training time than ANNs trained through back-propagation and can give comparable results. The only two hyper-parameters in ELMs are the number of hidden units, and the activation function. Table 3.5 shows the chosen ELM's hyper-parameters which were found through grid search.

Table 3.5: ELM adopted hyper-parameters

Parameter	**Adopted Value**
Number of hidden units	100
Activation function	Sine

3.5 Performance Assessment Metrics

Different performance assessment metrics for classification and regression models have been adopted to quantify the quality of the algorithms presented in the previous section.

3.5.1 Classification Models

Accuracy

is one of the most often used measures for assessing the performance of machine learning algorithms. It measures the overall performance of a classifier and is expressed as:

$$Accuracy = \frac{T_p + T_N}{T_p + T_N + F_p + F_N} \tag{3.16}$$

where T_p, T_N, F_p, F_N are respectively referring to the number of true positive, true negative, false positive and false negative samples.

F1-Score

also called the harmonic mean of precession and recall. It gives an insight into the combined performances of precession and recall. It is defined as:

$$F1 = 2 \times \frac{Precession \times Recall}{Precession + Recall} = \frac{2T_p}{2T_p + F_p + F_N} \tag{3.17}$$

ROC Curves

reflect the classification performance of a binary classifier as we change a threshold on the classifier soft probability values. It is a comparison of the recall (i.e., the true positive rate) and the false positive rate as the threshold is altered. Using ROC curves, the performance of a classifier is measured mainly in terms of the Area Under the Curve (AUC), where the better the classifier performs, the closer the AUC gets to 1.

3.5.2 Regression Models

Let $\hat{Y}$ be a vector of N_T predictions containing the predicted risk probabilities for N_T drivers, and Y is the test vector that contains the true N_T risk probabilities.

Mean-Square Error (MSE)

MSE is defined as the squared sum of the averaged differences between predictions and true labels. It can be expressed as:

$$MSE = \frac{1}{N_T} \sum_{1}^{N_T} (Y_i - \hat{Y}_i)^2 \tag{3.18}$$

Mean-Absolute Error (MAE)

MAE is defined in terms of the absolute deviation between true and predicted values. This is mathematically written as:

$$MAE = \frac{1}{N_T} \sum_{1}^{N_T} |Y_i - \hat{Y}_i| \tag{3.19}$$

R^2 Value

also known as the coefficient of determination, is another important statistical measure for assessing the performance of prediction models. It measures how much variance in the test vector Y the model can describe. It is computed using this formula:

$$R^2 = 1 - \frac{SS_{Regression}}{SS_{Total}} \tag{3.20}$$

where $SS_{Regression}$ and SS_{Total} are, respectively, the squared sum of the regression error and the squared sum of the total error. They are mathematically expressed in equations 3.21 and 3.22:

$$SS_{Regression} = \sum_{1}^{N_T} (Y_i - \hat{Y}_i)^2 \tag{3.21}$$

$$SS_{Total} = \sum_{1}^{N_T} (Y_i - \bar{Y})^2 \tag{3.22}$$

3.6 Results and Discussion

This section presents the performance results of the algorithms described in section V. The algorithms were implemented in Spyder (Python 3.6) Integrated Development Environment (IDE) using the Scikit-Learn Library for Machine Learning and Data Mining.

3.6.1 Training and Testing Splitting Methodologies

Two training and testing splitting methodologies have been adopted to train and validate the models.

1. *General Splitting Approach*: this is the common method used for choosing a randomly selected portion of the dataset for training and leaving the remaining dataset for testing. The splitting ratio usually depends on the amount of collected data and the application. In this work, 70% of the dataset is utilized for training. As a result, $1,404$ training samples and 603 testing or validation samples are used.

2. *K-fold Cross-Validation*: in this approach, the entire dataset is randomly divided into K equally sized partitions. In each training/ testing cycle, a single partition is kept for testing and all the remaining partitions are used for training. Training and validation are performed K times with each of the single partitions used once for testing. The mean and standard of deviation of the results can then be obtained to have more a statistical reflection on the model's performance. This approach is superior over the first approach since all data samples are utilized for both training and testing. In this work, *10-fold cross-validation* is adopted for all models.

3.6.2 Classification results

ROC curves for the six aforementioned algorithms using the general splitting approach are depicted in figure 3.6. As this figure illustrates, the RF algorithm produces the best AUC results among all other classifiers followed by the DNN. Specifically, RF produces the highest true positive rate for low false-positive rates (i.e., $FP < 0.1$).

Another measure of performance is the precision-recall curve. It gives useful insight into a classifier's performance for unbalanced labels. Figure 3.7 shows the

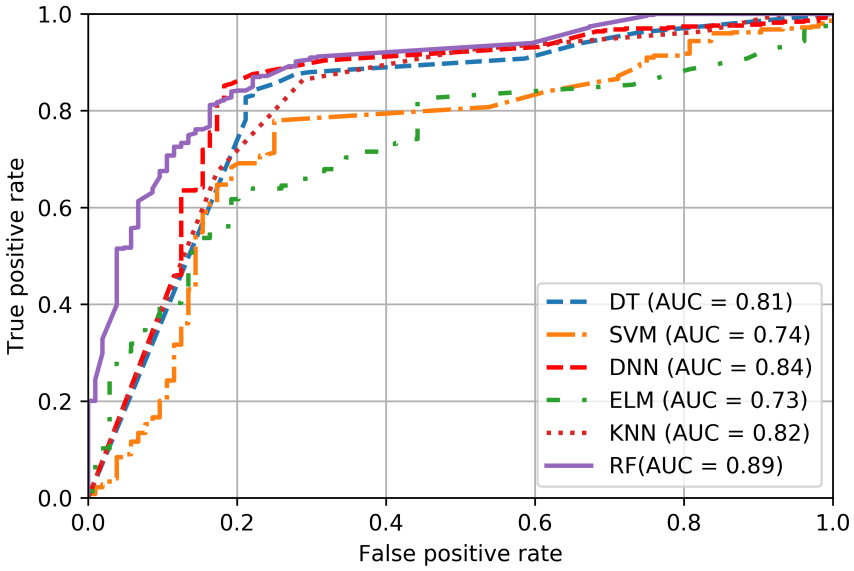


Figure 3.6: ROC Curves for DT, SVM, DNN, ELM, KNN and RF classifiers.

precision-recall curves for the six algorithms. Again, the RF classifier clearly outperforms all other classifiers with an average precision of 97%.

A summary of the remaining performance assessment results using the general splitting approach is shown in Table 3.6. The table shows consistency in performance superiority for RF classifier over the other five classifiers in all measures. RF achieves an accuracy of 87% and an F1-score of 0.93.

Figure 3.8 depicts the performance results using the 10-fold cross-validation approach. The shown figure displays the variation in performance metrics distributions for the six classifiers using whisker plots. Points outside whisker plots' range $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$ are considered outliers, where $Q1$ and $Q3$ are respectively the first and third quartile values of the whisker plot, and IQR refers to its interquartile range (i.e., $IQR = Q1 - Q3$). Figure 3.8(a) shows that the RF classifier has an accuracy that ranges between 88.5% and 92.2% with an average accuracy of

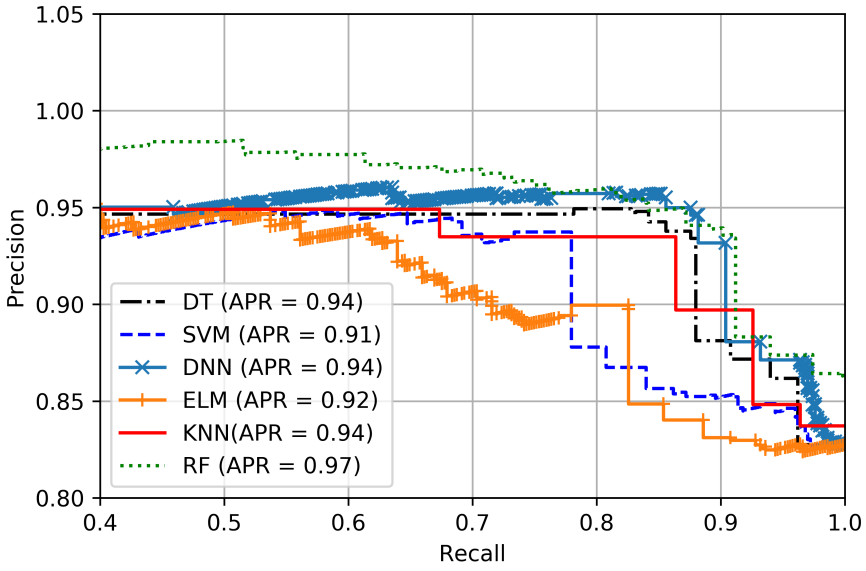


Figure 3.7: Precision-Recall Curve for DT, SVM, DNN, ELM, KNN and RF classifiers.

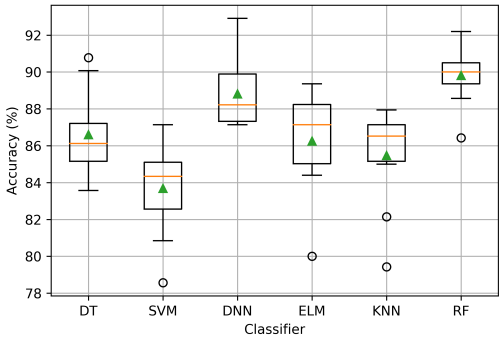
Table 3.6: Classification performance results using the general splitting approach

	Performance measure		
Algorithm	Accuracy (%)	F1-score	ROC curve AUC (%)
DT	84.1	0.910	81.3
SVM	81.5	0.896	72.3
DNN	85.4	0.917	84.2
ELM	81.6	0.900	73.6
KNN	81.9	0.895	80.7
RF	86.9	0.926	89

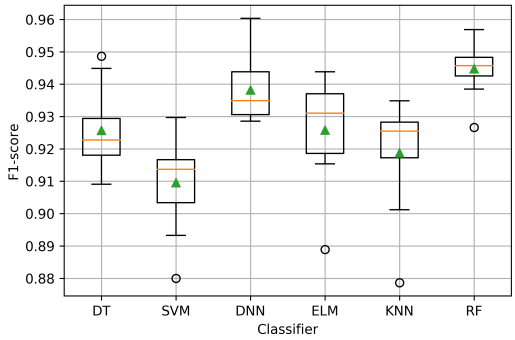
90%. The figure shows that the RF classifier outperformed the other six classifiers in the average sense and in its performance consistency over different training/testing samples. Similar conclusions can be drawn from figures 3.8(b) and 3.8(c) where the superiority of the RF classifier is consistently evident. A summary of the results is

shown in Table 3.7.

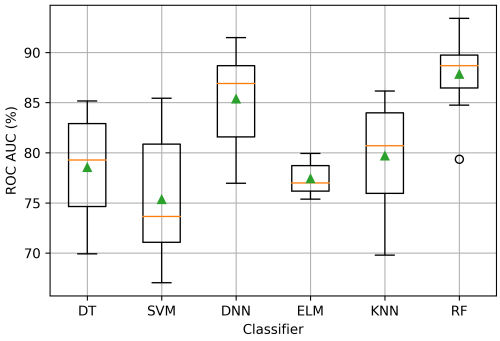
Two important observations can be made from the results. The first is the superior performance of the RF classifier over the DNN, whereas the second is the inferior performance of the SVM when compared to other classifiers. Concerning the first observation, despite the proven modelling power for DNNs, they seem to show their full potential when dealing with highly non-linear modelling problems with a large number of features and a very large number of training samples (big data). A possible reason of why the RF outperformed the DNN in this classification problem may be attributed to the size of the utilized dataset (intermediate size) and the relatively small feature space since only the 14 original features were used to train the DNN. With regards to the second observation, the poor performance of SVM in comparison to other classifiers is attributed to two main reasons. The first is the imbalanced classes in our classification problem since we have more positive labels, and secondly that the performance of SVM is highly dependent on the optimization of its hyper-parameters, especially the kernel function. Although different SVMs were trained on various kernels during the hyper-parameters' optimization as mentioned in section V, they still performed poorly when compared to other algorithms. From our experience, bagging algorithms (e.g., Random Forests) usually outperform SVM for intermediate data-sets with a relatively low number of features (e.g., the utilized data-set) unless a kernel that reflects the features' distribution is found, which is a computationally inefficient process (i.e., the computational complexity for training an SVM is between $O(n^2)$ and $O(n^3)$ where n is the number of training samples).



(a) Accuracy



(b) F1-score



(c) ROC AUC

Figure 3.8: Whisker plot for accuracy, F1-score and ROC AUC performances using 10-fold cross-validation.

3.6.3 Regression results

Comparison between regressors

We present herein the comparison results between DNN and RF regressors as they are the best two performing algorithms in the classification context. Table 3.8 shows the MSE, MAE and R^2 performance results for DNN and RF regressors using the general splitting approach. Similar to classification results, an RF regressor seems to outperform DNN regressor in all performance measures. Most importantly, the R^2

Table 3.7: Classification performance results using 10-fold cross-validation

	Performance measures		
Algorithm	Average accuracy (%)	Average F1-score	Average ROC curve AUC (%)
DT	86.6	0.926	78.6
SVM	84	0.910	75
DNN	88.8	0.938	85.4
ELM	86.2	0.926	77
KNN	85.5	0.92	80
RF	90	0.945	87.5

value for RF regressor is considerably higher with a difference gain of 25% over DNN regressor.

Table 3.8: Prediction performance results using general splitting approach

	Performance measures		
Algorithm	MSE	MAE	R^2
DNN	0.015	0.09	0.46
RF	0.008	0.05	0.71

Figure 3.9 shows the MAE, MSE and R^2 performance results of DNN and RF regressors. Again RF regressor outperforms DNN regressor in terms of consistency over different testing samples and in terms of its average performance. Particularly, DNN regressor seems to have very inconsistent R^2 results with a relatively small mean when compared to RF regressor. A summary of the results is shown in Table 3.9.

Figure 3.10 depicts the prediction vs. true $P(Risk|F_{ID})$ for a random sample of 100 drivers in the test set using RF regressor. The figure shows the ability of RF regressor to predict drivers' risk probabilities in most cases correctly.

Table 3.9: Prediction performance results using 10-fold cross-validation

	Performance measures		
Algorithm	Average MSE	Average MAE	Average R^2
DNN	0.018	0.105	0.41
RF	0.009	0.065	0.69

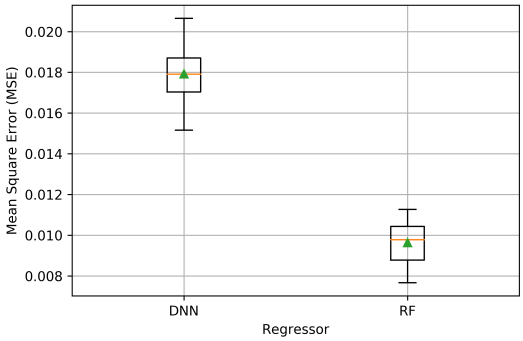
Conventional vs. proposed FoMs

We compare the performance of the RF model with the proposed predictors against its performance using the conventionally used FoMs that are usually adopted in the car insurance market, which are: excessive speeding, aggressive driving, sudden or improper braking and the total exposure time.

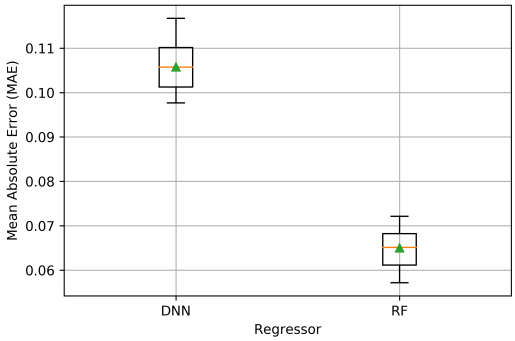
Figure 3.11 depicts the performance results of two RF models, one with the utilization of the proposed predictors (FoMs) and the other with the use of only the four conventional predictors that are used by car insurance companies. The results show a considerable difference between the two, where the model with the proposed FoMs is far superior. A summary of the comparison performance results is shown in Table 3.10.

Table 3.10: Comparison between performance results of two RF models using conventional and extended FoMs

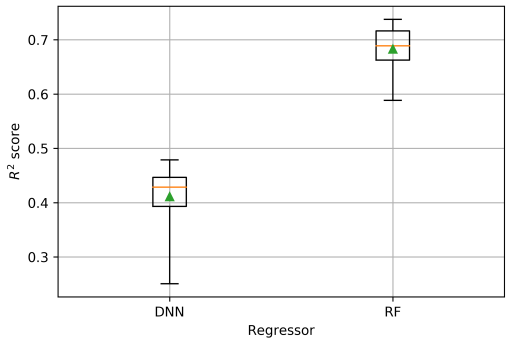
	Performance measures		
Algorithm	Average MSE	Average MAE	Average R^2
RF (Few FoMs)	0.018	0.097	0.43
RF (Extended FoMs)	0.009	0.065	0.69



(a) MSE



(b) MAE



(c) R^2 score

Figure 3.9: Whisker plot for MSE, MAE and R^2 performances using 10-fold cross-validation.

Test cases

Two test cases are presented in Tables 3.11 and 3.12. Table 3.11 shows that the relatively low percentage of safe driving (i.e., B_{11}) for driver 1 resulted in high risk probability of 0.655 specially when combined with highly risky behaviors such as: illegal or unsafe lane change or turn (B_8), fatigue or negligence (B_{12}), excessive speeding (B_1), and aggressive driving (B_9). In this case, the proposed RF regressor was able to predict the risk probability with a very low MSE of 0.0021.

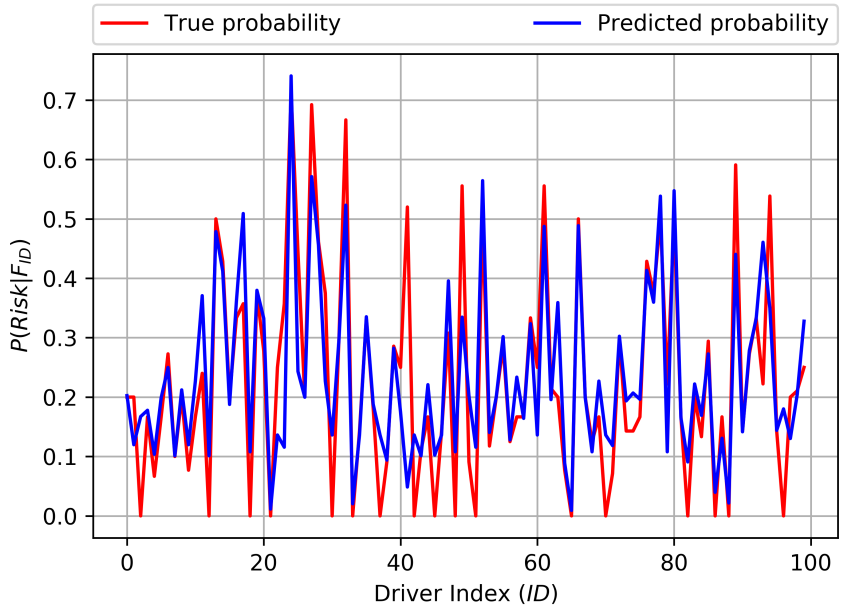


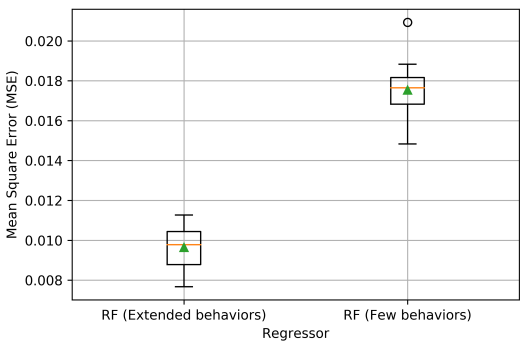
Figure 3.10: Predicted vs. true risk probabilities for a sample of 100 drivers using RF regressor.

Table 3.11: Test case for driver 1

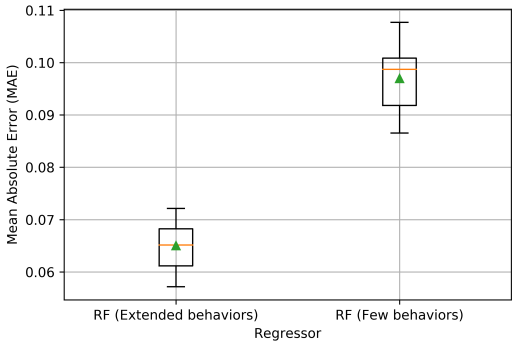
F_1	B_1 (%)	25
	B_8 (%)	10
	B_9 (%)	20
	B_{11} (%)	35
	B_{12} (%)	10
	T_{total}	7
$P(Risk)$	0.655	
$P(Risk\|F_1)$	0.662	

Table 3.12 shows that the very high percentage of safe driving for driver 2 (i.e., 87.5 %) was the dominant factor in having a low risk probability of 0.125. Similar to the first case, the MSE value here is negligible.

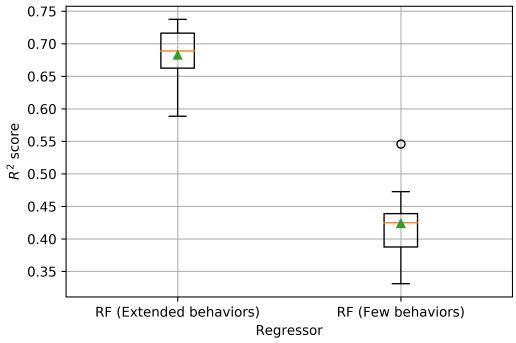
An important finding in the presented and many other cases is that driving risk can



(a) MSE



(b) MAE



(c) R^2 score

Figure 3.11: RF models' performances using conventional vs. proposed predictors.

be accurately predicted with only a few events captured with an appropriate sampling time (i.e., balanced base-line events denoted here as T_{total}). That is because given the relatively low rate at which the baseline events were taken in the SHRP2 dataset [50], the risk prediction models' irreducible error was insignificant and a snapshot of the behavioral pattern of different drivers was enough to predict their long-term risk. Therefore, there is no need for a continuous driving data acquisition to determine the associated risk of a certain driver. This has its relevance in minimizing the consumed power of offloading driving data to the cloud server in a cloud-based profiling system

Table 3.12: Test case for driver 2

F_2	B_8 (%)	6.2
	B_{11} (%)	87.5
	B_{12} (%)	6.2
	T_{total}	14
$P(Risk)$	0.125	
$P(Risk\|F_1)$	0.128	

and also in minimizing the computational cost for predicting driving risk.

Despite the insignificant models' irreducible error, more accurate results are anticipated given higher baseline events' sampling rate which should contribute to minimizing the models' errors.

3.7 Cloud-based Profiling System

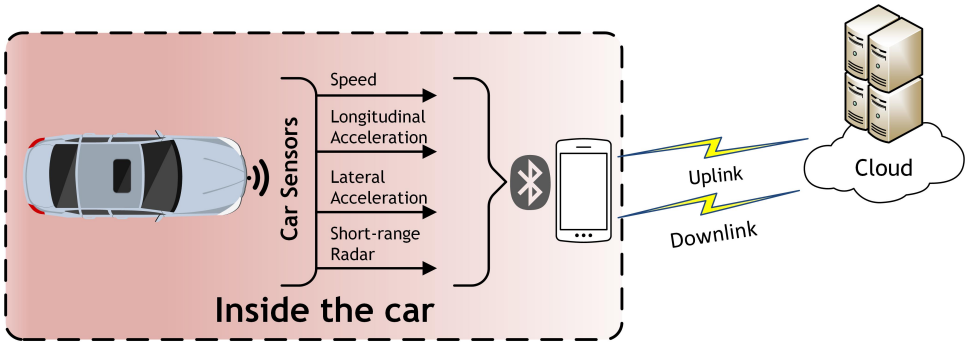


Figure 3.12: Uplink: A driver's smartphone sends the collected OBDII, radar and its inertial measurements to the cloud for processing. Inside the cloud, behaviors are classified using sequence modeling and inputted to the proposed driver scoring model. Downlink: A trip score is issued to the driver on a per-trip basis.

In real life profiling applications, the proposed risk profiling system can be hosted in a cloud as depicted in Figure 3.12. In the envisioned cloud-based profiling system, a

smartphone will serve as a hub in which real-time vehicle's network data (i.e., through OBDII units), the radar range data, and the smartphone inertial measurements are collected and forwarded to the cloud. On the cloud, such real-time data are leveraged to detect/classify driving behaviors through sequence modelling. Detected behaviors are then augmented in the proposed risk scoring function at the end of each driving trip. The calculated score is utilized to update the driver's risk profile and is sent back to the driver on a per-trip basis.

In the next chapter, an envisioned cloud-based profiling system that takes into account driving behaviors and their environmental context to predict risk is presented and validated.

3.8 Summary

In this chapter, a novel data-driven risk scoring framework for driver behavior profiling applications is proposed. Six machine learning algorithms are selected, customized and compared to achieve the best risk prediction performance. Algorithms are applied on SHRP2 NDS which is the largest NDS dataset collected to date. Results show the high-performance standards these algorithms can achieve in predicting risk probability with a performance advantage of the RF-based predictor. It was shown that the RF-based predictor could accurately model skewed data in which the histogram of the number of captured events per drivers is highly skewed to the left. This has practical significance in accurately predicting drivers' risk even for relatively short driving time. Good performance results are found consistent even with a relatively small number of captured events. This finding is very useful in a cloud-based profiling system to lessen the amount of consumed power caused by data off-loading, to reduce

the computational cost for predicting driving risk and minimizing the time needed before warning risky drivers.

A comparison between two customized RF regression models, one trained with only a few conventionally used predictors (FoMs) and the other trained with an extended set of proposed FoMs is established. The results show that the latter model outperforms the former in all performance measures as well as in performance stability over different sets of validation samples. Finally, given the successful results, the incorporation of the proposed system into a practical cloud-based driver profiling system is warranted. This system could be of great benefit to driver profiling companies in car insurance telematics and fleet administration domains.

Chapter 4

Cloud-Based Environment-Aware Driver Profiling Framework

Predicting expected risk based solely on the inclusion of detected behaviors-although more practical-ignores the environmental (e.g., weather and road conditions, traffic density level) context of detected behaviors. Coupling detected behaviors with their environmental context can be leveraged towards creating personalized risk profiles for drivers in each driving environment. These risk profiles can be utilized in various ITS applications including personalized safety-based route planning. In this chapter, a novel driver profiling environment-aware framework is presented. In the proposed framework, data processing is distributed over three computational layers to enhance the overall reliability of the system. With a developed risk notion, a risk prediction model is hosted in the edge/fog to determine the driving risk while considering the joint effect of the in-vehicle detected behaviors and their driving environmental context. Risk values along with a driver's compliance to warnings are both utilized to compute a driver's risk profile on the cloud. Using SHRP2 dataset, the development of a novel risk prediction model is presented herein with the underlying sub-processes

of data pre-processing, error analysis, and model selection. Then we analyze both the performance of the developed risk prediction model and the overall performance of the proposed system. Validation results for the developed randomized trees risk prediction model indicate a good trade-off between bias and variance with evidently high-performance results. Moreover, the results of the overall risk scoring model reflects its robustness and reliability in assigning accurate risk scores.

4.1 Introduction

The recent advancements in vehicular sensing, cellular communications, as well as cloud computing have enabled the deployment of various ITS applications [64, 65]. Given the high vehicle crash rates [12], these ITS applications are promising to lower these rates considerably. As mentioned in the previous chapter, an emerging safety-based ITS application is driver behavior profiling [66], which is applied in safety-based route planning, fleet management systems [67], and driver self-coaching systems [68].

Current risk scoring functions are not only subjective due to the absence of a valid risk measure (i.e., a risk measure quantified in terms of the actual risky events such as crash and near-crash), but they also ignored the environmental (e.g., weather and road conditions, traffic density level, etc.) effect on risk given the detected behaviors. For instance, an aggressive lane change in a highly dense driving environment could impose more risk than performing the same behavior in less dense traffic conditions. However, current profiling systems would equally penalize the subject driver in both scenarios regardless of where the behavior occurred since these systems only consider the behavior detection process [69, 70, 71, 35, 3].

NDSs have provided large-scale data about behavioral causes of risky events (i.e.,

crashes and near-crashes), as well as the environmental context of such behaviors (e.g., weather and road conditions, traffic density level, etc.). In addition, NDSs provide the same behavioral and environmental information during normal driving episodes, which enables the development of environmental-aware statistically significant risk prediction models [20].

The research question we address in this chapter is:

Are driving behaviors together with their environmental context good predictors for measuring risk probability?

To answer this question, the behavioral and environmental details of driving events presented in SHRP2 NDS are utilized to build a risk prediction model that can be incorporated in a complete cloud-based environment-aware driver profiling framework. The research contributions of this chapter are summarized as follows:

1. A novel Cloud-based Environment-aware Driver Profiling (CEDP) system is presented and discussed. The system provides a view on a "*next-generation*" driver profiling system in which drivers are profiled based on the expected risk of their environmentally-stamped driving behaviors and their compliance to warnings issued. The risk notion is mathematically developed and the terms: behavior detection, driving risk probability, driver scoring, and driver profiling, that are used interchangeably in the literature, are clearly distinguished and mathematically defined.

2. An ensemble supervised machine learning algorithm based on randomized trees is selected and customized to reflect the predicted driving risk probability while jointly considering the detected behaviors and their environmental context. The

model is proven to provide an acceptable compromise between bias and variance. The developed risk prediction model is trained and validated using an unprecedented amount of real driving data from SHRP2 NDS. This enhances the reliability and the practicability of the proposed system which is reflected in the performance results.

3. Given predicted risk probabilities, the performance of the overall risk scoring system is validated. Validation results show the robustness of the proposed system as it consistently provides accurate results over different training and validation samples.

To the best of our knowledge, no work in the literature has comprehensively considered a complete and detailed driver behavior profiling system that considers the sub-processes of behavior detection, risk prediction, driver's behavior scoring and profiling, and with consideration given to the driving environment. Although the environmental effect on risk has been comprehensively studied in the literature, the joint effect of driving behaviors and their environmental context on driving risk is presented in very few works, and not in the context of driver profiling [32]. In [72], the authors performed a statistical retrospective cohort study on the effect of traffic and road conditions on driving risk using the 100-CAR NDS. The authors in [73] used an NDS containing 1670 near-crash events to study the factors that are proportional to the increase in near-crash risk. They found that the road condition is one of the significant factors that affects driving risk.

In this chapter, an envisioned data-driven driver profiling system is introduced and discussed. We specifically targeted the problem of driving risk prediction by utilizing behavioral and environmental data of a large scale NDS (i.e., SHRP2). The

development of the risk prediction model is based on an error analysis of different supervised machine learning models to achieve the best bias-variance trade-off. The overall risk scoring system is then validated.

In this work, behaviors and the three environmental categories mentioned in chapter 2 are used as predictors to risk, quantified herein in terms of crash, near-crash, and crash relevant events.

The remainder of this chapter is structured as follows. Section 4.2 provides a detailed description of the envisioned CEDP system covering its in-vehicle, on edge/fog, and on cloud data processing. In section 4.3, the adopted pre-processing, error analysis and model selection processes for the risk prediction problem are described. In section 4.4, results are presented and analyzed. An illustrative example of the trip scoring process using the proposed framework is discussed in section 4.5. The chapter summary is presented in section 4.6.

4.2 Environment-Aware Profiling Framework

In this section, the proposed cloud-based environment-aware driver profiling framework is discussed. We cover the details of the complete driver profiling system, from the in-vehicle data acquisition to the cloud-based profiling. In short, acquired in-vehicle data is utilized to detect different driving behaviors. Detected behaviors are leveraged along with the environmental context in which they occurred to predict driving risk through a trained risk prediction model. If the predicted risk is higher than a pre-determined threshold, the subject driver (sd) is notified to change their behavior. Aggregated risk probabilities and the $sd's$ compliance to warnings throughout a certain driving trip are augmented in a scoring function to calculate the $sd's$

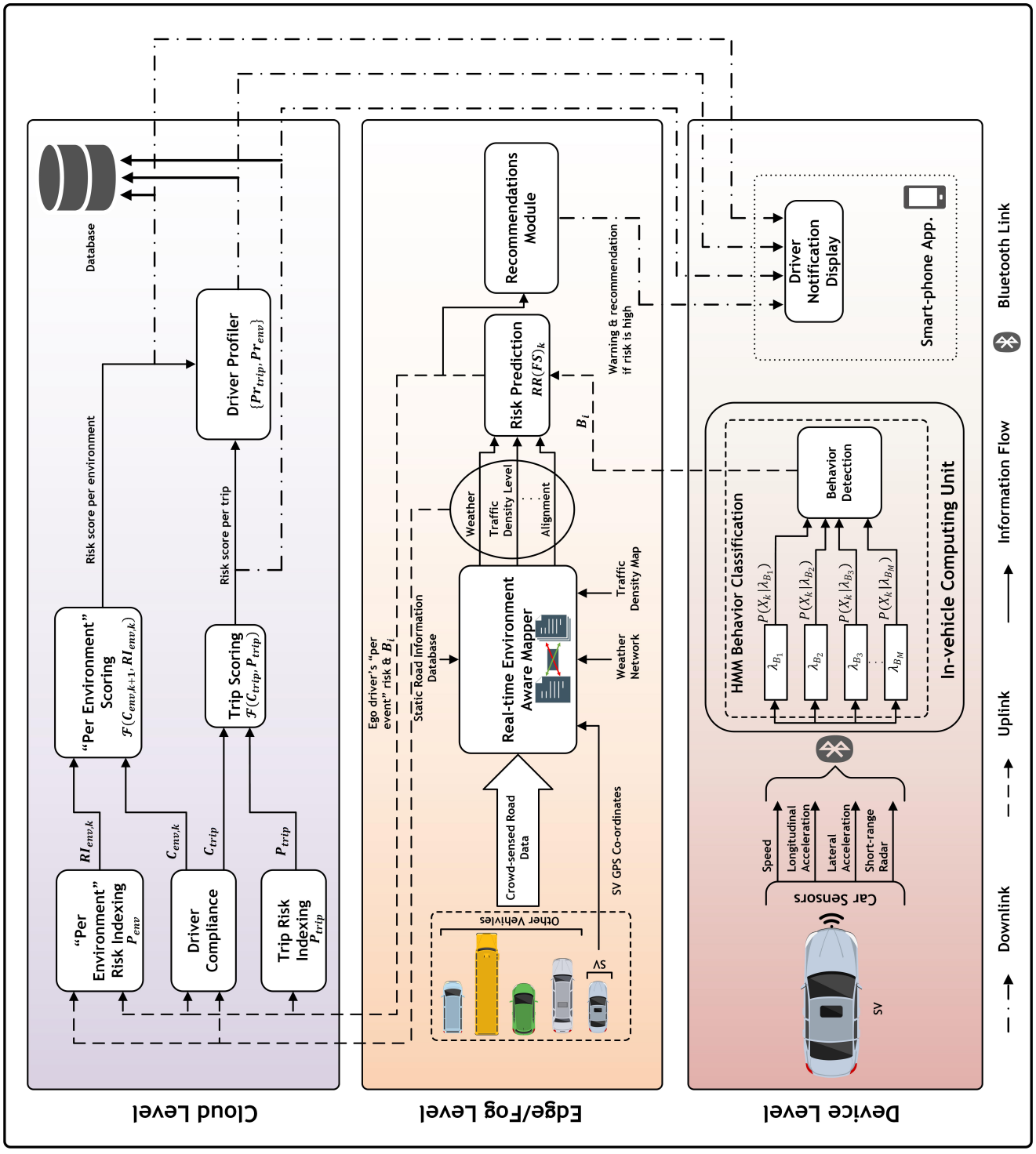


Figure 4.1: Proposed Cloud-based Environment-aware Driver Profiling Framework.

Table 4.1: Summary of Notations

Notation	Description
sd	Subject driver
sv	Subject vehicle
$\boldsymbol{x}_\tau$	In-vehicle feature vector at $t = \tau$
$\boldsymbol{X}$	In-vehicle feature matrix
R_s	Sampling rate of vehicular data
B_i	A detected driving behavior
T	A single in-vehicle time frame
λ_{B_i}	The sequence model representing the behavior B_i
F_l	Initial feature vector for risk prediction
FS_l	Engineered feature vector for risk prediction
$P(Risk\|FS_l)_k$	Predicted risk probability of an event k given FS_l
$P(Risk\|F_l)$	Calculated risk probability given F_l
env_j	A vector with extracted environmental attributes
$RR(F_l)$	The relative risk of F_l
$RI(k)$	The risk index of an event k
C_{trip}	Driver's overall compliance in $trip$
N	Total number of captured risky events per trip
P_{trip}	Average risk in $trip$
Sc_{trip}	$sd's$ score in $trip$
Sc_{env_j}	$sd's$ score in env_j
Pr_{trip}	$sd's$ risk profile after $trip$.
ξ	Weight of EMWA filter

trip score. The $sd's$ risk profile is then calculated as a weighted sum of different trip scores. Unlike other profiling systems, the proposed system is motivated by statistically significant results as will be shown in section 4.4. Figure 4.1 depicts the framework block diagram.

In the proposed framework, data processing is distributed over three computational layers based on the computational requirements of processes, delay, delivery, and accessibility requirements of processed data, and processed data size. Details are provided in the following section.

4.2.1 Device Level: In-vehicle Behavior Detection

The in-vehicle module contains data acquisition, pre-processing and modeling processes that occur inside the vehicle to detect different driving behaviors. In this module, collected data can be divided into two types:

1. *Type 1*: Data that reflects the longitudinal and lateral behavior of the vehicle. This data is collected through the vehicle's CAN bus and by utilizing the vehicle's OBD/OBDII port.

2. *Type 2*: Data that reflects the relative position of the subject vehicle to the surrounding vehicles and provides driving context-awareness. This is gathered using Short Range Radar (SRR) sensors.

Let $\boldsymbol{x}_\tau$ represents the feature vector that contains the collected vehicular data at time instant τ and expressed as:

$$\boldsymbol{x}_\tau = [v_\tau, a_{x,\tau}, a_{y,\tau}, R^F_{x,\tau}, R^F_{y,\tau}, R^R_{x,\tau}, R^R_{y,\tau}] \tag{4.1}$$

where v_τ represents the velocity of the sv, $a_{x,\tau}$ and $a_{y,\tau}$ represent the acceleration in the longitudinal and lateral directions of the sv, respectively, $R^F_{x,\tau}$ and $R^F_{y,\tau}$ are, respectively, the ranges between the sv and the closest forward object in the longitudinal and lateral directions, $R^R_{x,\tau}$ and $R^R_{y,\tau}$ are, respectively, the ranges between the sv

and the closest rearward object in the longitudinal and lateral directions, all at the time instant $t = \tau$. After τ_c seconds, collected data can be expressed in the following matrix notation:

$$\boldsymbol{X} = \begin{bmatrix} — & \boldsymbol{x}_1 & — \\ — & \boldsymbol{x}_2 & — \\ & \vdots & \\ — & \boldsymbol{x}_{(\tau_c \times R_s)} & — \end{bmatrix} \tag{4.2}$$

or equivalently:

$$\boldsymbol{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x}(1) & \boldsymbol{x}(2) & \dots & \boldsymbol{x}(Le) \\ | & | & & | \end{bmatrix} \tag{4.3}$$

where R_s stands for the data sampling rate and Le is the length of the feature vector $\boldsymbol{x}_\tau$) (i.e., seven in this case). Data is collected and sent from OBD and radar interfaces to the $sd's$ in-vehicle computing unit (e.g., smartphone) through a Bluetooth link. In the in-vehicle computing unit, the time-series vehicular data ($\boldsymbol{X}$) is acquired over a pre-determined time interval τ_c and sequence modeling for behavior classification (e.g., HMM-based Modeling) is applied. The behavior classification is defined as the process:

$$\mathcal{F} : \{\boldsymbol{x}(1), \dots, \boldsymbol{x}(Le)\} \rightarrow B_i \tag{4.4}$$

where B_i, $i = 1, \dots, M$ represents one of M output behaviors on which the sequence model is trained to detect.

A single time frame in the in-vehicle module is depicted in figure 4.2 and can be

expressed mathematically as:

$$T = \tau_c + \tau_p + \tau_o + \tau_I \tag{4.5}$$

where τ_p is the sequence model's processing time for behavior detection, τ_o is the time required for off-loading a detected behavior to the edge/fog, and τ_I is the idle time where no vehicular data is acquired.

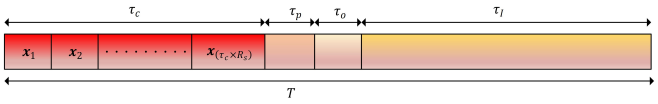


Figure 4.2: A single time frame of collecting and offloading data.

After the behavior B_i is detected, it is sent to the edge/fog, along with the GPS co-ordinates of the sv for analysis and processing.

In the proposed framework, behavior detection is performed inside the vehicle to ensure high detection accuracy and to minimize the cost of data off-loading. High levels of accuracy in behavior detection is essential given its importance for predicting risk. With the high rate at which vehicular data are sampled (on the scale of sub-seconds), performing behavior detection inside the vehicle should diminish data loss caused by off-loading data, and hence, should ensure high detection accuracy. Furthermore, transmitting vehicular data to the fog/cloud would incur a lot of transmission cost to drivers. To illustrate, the total amount of traffic data in a 1 hour trip with $\tau_c = 10s$ and $\tau_p + \tau_o + \tau_I = 10s$, and with a data rate of $1KB/s$ will be $1.8MB$ of transmitted cellular data.

Algorithm 1 shows a summary of the explained behavior detection process.

Algorithm 1: In-vehicle Behavior Detection

Input: Vehicular data: $\{\boldsymbol{x}\}_{\tau=1}^{\tau=\tau_c \times R_s}$, Data Collection Time: τ_c, Idle Times: $\{I_F, I_T\}$
Output: B_i

```
repeat
    for τ ← 1 to τ_c × R_s do
        X.append(x_τ)
    for k ← 1 to M do
        Calculate P(X|λ_{B_k})
        P.append(P(X|λ_{B_k}))
    i = arg max{P}
    Offload B_i & location co-ordinates
    if warning ='FALSE' then
        τ_I = I_F
    else
        τ_I = I_T
until trip = 'FALSE'
```

4.2.2 Edge/Fog Level: Risk Prediction and Recommendation Modules

On the edge/fog level, driving risk is predicted based on the detected behavior of the *sd* along with the environmental context in which the behavior was detected. The *sd* is warned and advised to change their driving behavior through a recommendation module if expected risk exceeds a pre-defined threshold. We assume the existence of a real-time environment aware mapper to which the $sv's$ GPS co-ordinates are inputted and the environmental road segment attributes, which the vehicle was subjected to during detected behavior, are returned. The envisioned mapper has access to the static and dynamic road information databases on the area the designated edge/fog covers. The mapper is hosted in the edge/fog level rather than in the cloud level to minimize the time required for pulling out the environmental information of the desired road segment. To explain, having a centralized road information database in

the cloud that contains the information of a large traffic network would increase the search time needed for extracting the information of a designated road segment, and hence will increase the time needed for predicting risk. Likewise, both risk prediction and recommendation modules are hosted in the edge/fog level to reduce the time required to calculate the expected risk of a captured event and to reduce the time latency between predicting risk and warning a risky driver.

Environmental attributes contain static information about the road characteristics, and the real-time road information such as density level, weather condition, traffic flow, and lighting conditions. In this framework, we utilized the following environmental attributes: weather condition (W), traffic density level (TD), road lighting conditions (L), traffic control (TF), road flow (RF), and road alignment (A). The returned environmental attributes vector env_j, where $j \in [1, \ldots, J]$, along with the $sd's$ detected behavior B_i form the initial feature vector F_l, $l = 1, \ldots, L$:

$$F_l = [B_i, env_j] \tag{4.6}$$

Feature extraction and selection is then performed on the initial feature vector. The engineered feature vector (FS_l) is then inputted to a trained risk prediction model.

The risk prediction model uses FS_l to predict the driving risk probability $P(Risk|FS_l)_k$, where the subscript k is an integer that represents an event index. The driving risk probability is expressed herein in terms of the crash and near-crash rate:

$$P(Risk|FS_l)_k = P(C|FS_l)_k + P(NC|FS_l)_k \tag{4.7}$$

where $P(C|FS_l)_k$ and $P(NC|FS_l)_k$ are, respectively, the conditional probabilities of crash and near-crash events (including crash relevant events) given the feature vector FS_l at event k. The conditional risk probabilities in different driving environments are calculated as:

$$P(Risk|F_l) = \frac{R_{F_l}}{R_{F_l} + NR_{F_l}} \tag{4.8}$$

where R_{F_l} and NR_{F_l} are, respectively, the number of risky and non-risky events, given F_l. In SHRP2 data-set, a non-risky event is either a non-subject conflict, or a balanced baseline event, as they are previously defined.

Once risk probability is predicted, a warning is issued to the subject driver. The level of warning severity changes according to the level of risk the detected behavior imposes. Since the risk probability is data-set dependent and is characterized by the sampling rate at which normal driving events are captured, the threshold between risk levels can be set using the following relative risk equation:

$$RR(F_l) = \frac{P(Risk|F_l)}{P(Risk|F_l')} \tag{4.9}$$

where $RR(F_l)$ is the relative risk of F_l, and F_l' is the complement of F_l (i.e., $[B_i, env_j]'$).

Based on the relative risk values, risk severity is assigned and warnings are issued accordingly. In this work, risk severity during a driving event belongs to the set $\{Severe, Critical, High, Normal, Low\}$ or equivalently to the integer set $\{4, 3, 2, 1, 0\}$, as shown in table 4.2.

As shown in table 4.2, risk severity levels are assigned depending on the relative risk of a captured event. A driving event with a relative risk of 1 possesses a risk probability equivalent to the average risk probability of events captured in other

Table 4.2: Risk Severity Levels

Risk Severity	Definition
Severe (4)	A driving event where $RR(F_l) > 4$. A warning is issued. A non compliance to warnings results in zero compliance score.
Critical (3)	A driving event where $3 < RR(F_l) \leq 4$. A warning is issued. A non compliant driver receives only a one-quarter compliance score.
High (2)	A driving event where $2 < RR(F_l) \leq 3$. A warning is issued. A non compliant driver looses half of his/her compliance score.
Normal (1)	A driving event where $1 < RR(F_l) \leq 2$. A warning is issued. A non compliance to warnings results in a one-quarter reduction in compliance score.
Low (0)	A driving event where $RR(F_l) \leq 1$. No warning is issued.

driving environments. Consequently, a relative risk of 1 was chosen as a threshold between low-risk events and other events. If a captured event imposes some risk, the *sd* will be notified and advised to change his/her behavior as to reduce risk. The *sd* receives a complete compliance score unless he/she does not change behavior to normal. If the *sd* is not compliant, the reduction of his/her compliance score will be directly proportional to the event risk severity.

The $sd's$ compliance to warnings along with a weighted sum of the aggregated risk probabilities over a certain trip are used to compute the final trip score Sc_{trip} as will be detailed in the next section.

4.2.3 Cloud Level: Scoring and Profiling Processes

On the cloud level, time-tolerant computationally intensive operations are hosted. On the cloud, the overall risk and compliance of drivers through their driving trips are computed. Risk and compliance are utilized afterwards to calculate trip scores or to update personalized competency levels of drivers in various driving environments. Based on risk and compliance scores, risk profiles of drivers are continuously updated after each driving trip and stored in a centralized database. Drivers are notified about their overall scores and regarding relevant updates to their driving profiles following the end of each driving trip. Processing this amount of data requires a High-Performance Computing (HPC) servers which are available on the cloud level. We will take the computation of driver compliance as an example to highlight the asymptotic time complexity of the on-cloud operations. In a time slot t, computing the compliance to warnings for M drivers in E events will be of the order of $O(M \times E)$. Repeating this process K times will incur a computational cost of $O(M \times E \times K)$. With such high computational cost, it is reasonable to host such operations in the cloud. Next, the logical flow of information in the cloud is detailed.

Following risk prediction of event k, predicted risk is offloaded to the cloud and inputted to the "Trip Risk Indexing" module. Based on the predicted risk severity level, the event k is assigned a risk index RI_k according to equation 4.10:

$$RI(k) = 0.25 * sl_k \tag{4.10}$$

where $sl_k \in \{0, 1, 2, 3, 4\}$ is the risk severity of event k and is one of the risk severity levels shown in table 4.2. Risk indices for all captured events during a driving trip

are computed and stored. The overall trip risk index P_{trip} can be simply calculated as the trip average risk, which is denoted by the following formula:

$$P_{trip} = \frac{1}{N} \sum_{k=1}^{N} RI(k) \tag{4.11}$$

where N is the total number of captured events during a trip.

The *sd* compliance to a warning following being involved in a risky behavior during event k is calculated through the "Driver Compliance" module during event $k+1$ (i.e., monitoring the driver behavior after issuing a warning). As shown in table 4.2, compliance is computed according to the risk severity of k. By way of explanation, the *sd* is given the full compliance score of 1 if the driver is compliant. If the driver is non-compliant, a deduction in compliance score is weighted according to risk severity during the event k. The binary variable c_{k+1} is defined as follows:

$$c_{k+1} = \begin{cases} 1, & \text{if } RI(k+1) > 0 \\ 0, & \text{if } RI(k+1) \leq 0 \text{ or } B_{i,k} \text{ is normal} \end{cases} \tag{4.12}$$

Then, compliance with a warning following a risky behavior in event k is expressed as:

$$C(k) = 1 - c_{k+1} * RI(k) \tag{4.13}$$

Similar to the overall trip risk index P_{trip}, the overall trip compliance, C_{trip}, is calculated as the average compliance throughout a driving trip. It is expressed mathematically as:

$$C_{trip} = \frac{1}{N-1} \sum_{k=1}^{N-1} C(k) \tag{4.14}$$

The above argument requires repeating the in-vehicle processes of data collection, behavior detection, and data offloading, as well as the cloud risk prediction process, each time after detecting a risky behavior. This repetition is to check if the driver complied to the warning. A simpler and more practical yet less accurate approach is to calculate the $sd's$ compliance based on their compliance probability distribution for events of different severity levels.

Under the assumptions of:

1. Independent sd compliances in different risky events.
2. Equally probable compliance rates in different driving environments and for events with the same risk severity level.

the probability of l compliances in N_{sl} risky events of severity level sl would follow a binomial distribution with parameter p_{sl}:

$$P(C_{sl} = l) = \binom{N_{sl}}{l} p_{sl}^{l} (1 - p_{sl})^{N_{sl} - l} \tag{4.15}$$

The overall compliance per trip C_{trip} would be the probability of being always compliant (i.e., $l = N_{sl}$, $\forall sl \in \{0, 1, 2, 3, 4\}$). Substituting equation 4.15 in equation 4.13, C_{trip} can be expressed as follows:

$$C_{trip} = \sum_{sl=0}^{sl=4} (1 + RI_{sl}).P(C_{sl} = N_{sl}) - RI_{sl} \tag{4.16}$$

This simplified formulation will require only calculating the probability parameters p_{sl}, $\forall sl \in \{0, 1, 2, 3, 4\}$ in a primary training phase, which is more practical in many situations. These probability parameters can be updated regularly to track the changes in a driver's compliance behavior.

The trip score is then computed as a function of the trip weighted sum of the risk index P_{trip}, and the driver's per trip compliance value C_{trip}:

$$Sc_{trip} = \mathcal{F}(C_{trip}, P_{trip}) \tag{4.17}$$

Given that $P_{trip} \in [0, 1]$ and $C_{trip} \in [0, 1]$, a normalized $Sc_{trip} \in [0, 1]$ can be written as:

$$Sc_{trip} = \gamma.C_{trip} + \alpha.(1 - P_{trip}) \tag{4.18}$$

where

$$\gamma + \alpha = 1 \tag{4.19}$$

The values of γ and α determine how much weight is given to C_{trip} and P_{trip}. For instance, if $\alpha = 1$, the overall trip score will be determined solely based on the value of P_{trip} (i.e., $\gamma = 0$).

Finally, a subject driver's profile after a certain trip (Pr_{trip}) can be computed using an Exponentially Moving Weighted Average (EMWA) filter applied on various trip scores to assign exponentially increasing weights for recent trips. This is expressed as:

$$Pr_{trip} = \begin{cases} Sc_1, & \text{if } \ trip = 1 \\ \xi.Sc_{trip} + (1 - \xi).Pr_{trip-1}, & \text{if } \ trip > 1 \end{cases} \tag{4.20}$$

where the value of ξ determines the number of trips which the filter will use to calculate Pr_{trip}.

The $sd's$ per-environment profile is updated using the same analogy for updating the $sd's$ per-trip profile. The "Per Environment Risk Indexing" module calculates

Table 4.3: Summary of Environmental Conditions

Environmental attribute	***Values***			
Traffic Flow	*Divided*	*Not Divided*	*No Lanes*	-
Traffic Density	*Stable*	*Stable With Flow Restrictions*	*Unstable*	-
Traffic Control	*Yes*	*No*	-	-
Weather Conditions	*No Adverse Conditions*	*Foggy*	*Rainy*	*Snowy*
Lighting Conditions	*Dark*	*Lighted*	-	-
Road Alignment	*Straight*	*Curved*	-	-

$RI_{env_j}(k)$ which is the risk index for event k taken into consideration the environmental context of the event, calculated for each env_j. $RI_{env_j}(k)$ is utilized to reflect the driving competency level of the sd in the driving environment env_j along with the compliance $C_{env_j}(k)$. Consequently, the score of the sd in env_j at event k is:

$$Sc_{env_j}(k) = \gamma.C_{env_j}(k) + \alpha.(1 - RI_{env_j}(k)) \tag{4.21}$$

An sd profile in env_j (Pr_{env_j}) can then be updated after each event captured in env_j. Similar to the per trip profile, Pr_{env_j} can be computed using an EMWA filter to assign exponentially increasing weights for recent captured events in env_j.

An important feature of the presented framework is the prediction of driving risk probabilities given the behavioral and environmental attributes. Non-accurate values of these probabilities can result in missed or false warnings as well as unreliable driving scores. The rest of the chapter contains the necessary steps for the development of the driving risk prediction model. Moreover, the effect of risk prediction results on the overall scoring performance is analyzed using SHRP2 naturalistic driving data.

4.3 Data Pre-processing and Model Selection

Raw data contains information about $\sim$ 29,000 driving events, each with a certain severity level. In the original dataset, event severity levels are exclusively contained in the following set: *Severity* $\in$ {*Crash, Near-Crash and Crash-Relevant, Non-Subject Conflict, Balanced Baseline*}. An event k in the dataset is represented by a vector that contains the captured driving behavior of the subject driver prior to a risky event (or during a normal driving event) (B_i), the environmental context in which these behaviors happened (env_j), and the event severity ($Severity$):

$$k = [[B_i, env] \xrightarrow{yields\ to} Severity] \tag{4.22}$$

Since our problem is to classify the risk level of an event given the behavior of the driver and the environmental context, the notion of risk is developed as shown in equations 4.7-4.9. The initial feature matrix is transformed from the original event-based matrix to the following matrix:

$$\left[\begin{array}{c|cc|c} \text{Feature Vector } (F_l) & B_i & env_j & \text{Outcome } (RR(F_l)) \\ \hline F_1 & B_1 & env_1 & RR(F_1) \\ F_2 & B_1 & env_2 & RR(F_2) \\ \vdots & \vdots & \vdots & \vdots \\ F_L & B_\mu & env_J & RR(F_L) \end{array}\right] \tag{4.23}$$

4.3.1 Data Pre-processing

Data Merging

In this work, *Crash, Near-Crash* and *Crash-Relevant* severity levels are put under the common severity level of *Risky*, whereas *Non-Subject Conflict* and *Balanced Baseline* events are used to represent the *Normal* level. Under each environmental category, similar features are merged to increase their importance in order to enhance the prediction model performance (e.g., under the road alignment category, whether curved to the right or curved to the left these road features are considered the same). Similarly, the 13 behaviors previously identified in chapter 3 are utilized. Identified behavioral and environmental features are shown in tables 3.1 and 4.3, respectively.

Data Filtering

Rows in the feature matrix are filtered out if their relative risk values ($RR(F_l)$) are not statistically significant. The $p-value$ is utilized to signify the statistical significance. Rows which possess a $p-value > 0.1$ are filtered out. The filtered feature matrix has L' rows. With the Contingency table shown in table 4.4, the $p-value$ is calculated for each row l using Fisher's exact ratio as:

$$p_l = \frac{\binom{a+c}{a}\binom{b+d}{b}}{\binom{n}{a+b}} \tag{4.24}$$

where $n = a + b + c + d$.

Table 4.4: Contingency table for the number of risky and non-risky events

	Risk	**No risk**
F_l	a	c
F_l'	b	d

Data Encoding

After data merging, the behavioral and environmental categorical variables are encoded to integers. Events with the same behavioral and environmental features are combined and the corresponding risk probability for each is calculated. To represent data in a meaningful way for the machine learning algorithms, we used the *one-hot encoding* technique.

4.3.2 Model Selection

After data is encoded, it is divided into training and development sets according to the ratio of 70% and 30%, respectively. Using MAE as a performance metric, an error analysis for a simple multiple linear regression model indicated a high bias (i.e., low training set performance). More complex structured SVM-based models, on the other hand, were able to model training data accurately, but were not capable of generalizing on the development set (i.e., high variance). To achieve a good bias-variance trade-off, we selected a customized random forest model. In the random forest algorithm [74], multiple decision trees are built, each from a sample of the training set. The best split in each tree is based on a random subset of the input features rather than the whole feature set. The average performance of the various trees is then used to reflect the forest performance. Although this approach theoretically causes a slight degradation

in the training set performance, it reduces over-fitting due to the averaging process. In this work, a customized random forest model resulted in the best bias-variance performance the regression context.

The adopted hyper-parameters of the selected model are shown in table 6.4, where N_{tot} represents the number of all behavioral and environmental features, and MSE is the mean square error.

Table 4.5: Hyper-parameters of RF Model

Hyper-parameter	**Classification**	**Regression**
Number of Trees	100	100
Split Criterion	Entropy	MSE
Max No. of Features per Tree	$\sqrt{N_{tot}}$	N_{tot}

4.4 Performance Evaluation and Discussion

In this section, the performance results of the Random Forests risk prediction model presented in section 4.3 are investigated. The model was implemented in Spyder (Python 3.6) IDE using the Scikit-Learn Library for Machine Learning and Data Mining. Results in the regression context are discussed along with the relevant risk index RI and the overall risk scoring results. Reported results are those obtained from the customized RF model after trying different random seeds. They represent the best obtained results.

4.4.1 Risk Prediction

The developed RF model is trained to predict the relative risk of a specific event given the driver's behavior and the environmental context. The model was trained

and validated according to the splitting ratio of 70 % and 30 %, respectively. The $10 - fold$ cross-validation was performed to reflect the average performance of the model over different training samples. The normalized absolute error histograms of the model for both training and validation sets are depicted in figures 4.3 and 4.4, respectively.

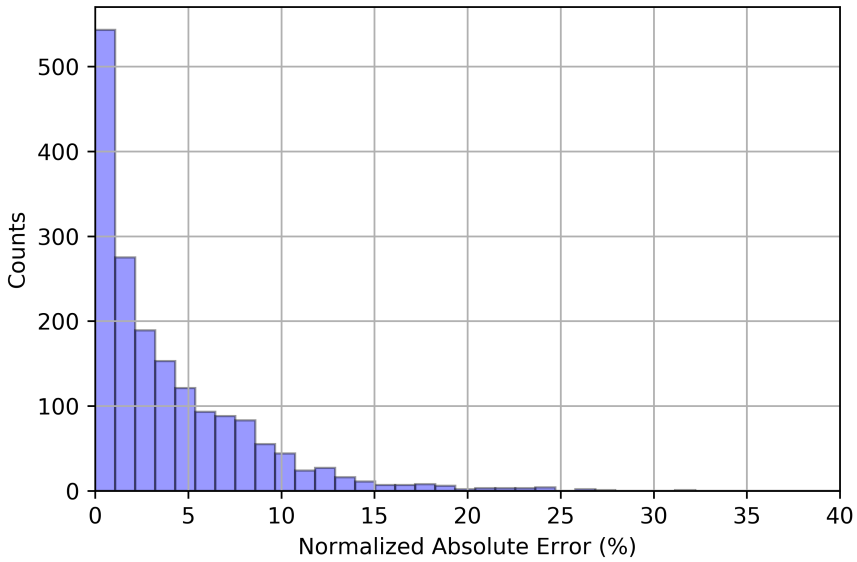


Figure 4.3: The normalized absolute error histogram for the training set using the developed RF risk prediction model.

The Normalized Absolute Error (NAE) percentage of a feature vector F_l is calculated according to equation 4.25:

$$NAE(F_l)\% = \frac{|RR_{act}(F_l) - RR_{pred}(F_l)|}{\max(RR_{act}) - \min(RR_{act})} \tag{4.25}$$

where $RR_{act}(F_l)$ and $RR_{pred}(F_l)$ are, respectively, the actual and predicted relative risk values for the feature vector F_l, and RR_{act} is the vector that contains the actual

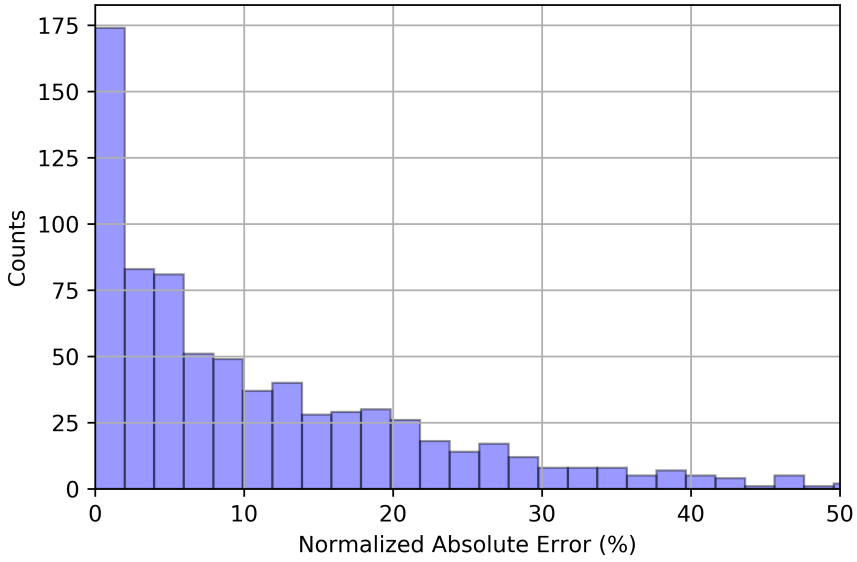


Figure 4.4: The normalized absolute error histogram for the validation set using the developed RF risk prediction model.

relative risk values for all the feature vectors in the data-set. Figure 4.3 shows that the sample count is exponentially decreasing as the NAE increases, with a maximum NAE of 27%. Similarly, the validation set NAE performance resembles an exponential distribution but with a higher normalized mean absolute error $NMAE$ as shown in figure 4.4. The summary of the model $NMAE$ and R^2 results is shown in table 4.6. The validation set results show the high-performance standards the developed model can achieve with an average $NMAE$ of only 10.7% and with an ability to explain most of the variability in the data output as shown from the coefficient of determination value (e.g., 0.66). Moreover, the training set performance indicates that the developed model has a very small bias with an $NMAE$ value of 4.25% and R^2 value of 0.95. Despite the good performance results for both training and validation sets, the validation set performance shows a 6.45% degradation in the $NMAE$ performance

when compared to the training set. Furthermore, a 0.29 difference in the R^2 value is noticed. This reflected that the developed model is slightly over-fitted. Although this bias-variance combination was the best achieved, an over-fitting was unavoidable which may be attributed to the data-set sample size.

Table 4.6: Summary of the RF Model Results

Performance Measure	Training Set (%)	Development Set (%)
NMAE (%)	4.25	10.7
Adjusted R^2	0.95	0.66

The actual and predicted risk indices are respectively computed from the actual and predicted relative risk values using equation 4.10, . The mean absolute error (MAE) metric is utilized to signify the performance. Figure 4.5 depicts the Whisker plot of the MAE for the risk index RI using $10-fold$ cross-validation. The average MAE for the training and validation sets is, respectively, 2% and 8.7%. Such negligible average errors highlight the accurate RI results, and hence, the accurate scoring results as will be shown in section 4.4.2.

4.4.2 Driver Scoring

In this section, the expected value of the deviation in an event score is derived and empirically calculated given the SHRP2 event-based dataset. As shown in section 4.2.3, the performance of a driver in an event k is calculated based on the risk index RI_k and the compliance C_k. The absolute error in the score of a driver given the feature vector F_k is defined as the absolute difference between the actual and predicted scores of a driver in an event k. It is denoted by $Sc_{error}(k)$ and can be expressed

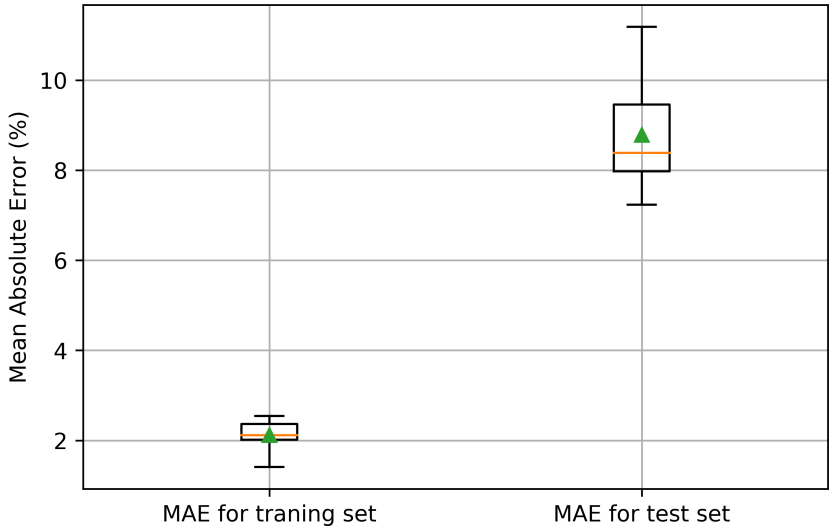


Figure 4.5: Whisker plot for the MAE performance of RI using $10-fold$ cross-validation

mathematically as:

$$\begin{aligned} |Sc_{actual}(k) - Sc_{pred}(k)| = \alpha.|RI_{act}(k) - RI_{pred}(k)| \\ + \gamma.|C_{act}(k) - C_{pred}(k)| \end{aligned} \tag{4.26}$$

where $Sc_{act}(k)$ and $Sc_{pred}(k)$ are, respectively, the actual and predicted risk scores of a driver in event k. The expected value of the absolute error Sc_{error} can then be expressed as:

$$\begin{aligned} Sc_{error} = \alpha.\,\mathbb{E}(|RI_{act} - RI_{pred}|) \\ + \gamma.\,\mathbb{E}(|C_{act} - C_{pred}|) \end{aligned} \tag{4.27}$$

where $\mathbb{E}(|RI_{act} - RI_{pred}|)$ and $\mathbb{E}(|C_{act} - C_{pred}|)$ are, respectively, the mean absolute errors for the risk index and the compliance scores. Denoting the feature vector at event k by F_l and the feature vector at event $k+1$ by F_j, Sc_{error} can be written as:

$$\begin{aligned}\mathbb{E}(Sc_{error}) = \alpha.\sum_{l=1}^{L'} P(F_l).|RI_{act}(l) - RI_{pred}(l)| \\ + \gamma.\sum_{j=1}^{L'}\sum_{l=1}^{L'} P(F_j|F_l).|C_{act}(l) - C_{pred}(l)|\end{aligned} \tag{4.28}$$

where $P(F_l)$ and $P(F_j|F_l)$ are the probability of F_l and the conditional probability of F_j given F_l, respectively.

The absolute deviation in compliance ($|C_{act}(l) - C_{pred}(l)|$) is calculated for four cases:

1. The model predicts that the driver is compliant given that the driver is actually compliant. In this case, $|C_{act}(l) - C_{pred}(l)| = 0$.

2. The model predicts that the driver is non-compliant while the driver is actually compliant. In this case, $|C_{act}(l) - C_{pred}(l)| = RI_{pred}(l)$.

3. The model predicts that the driver is compliant while the driver is actually non-compliant. The absolute deviation in compliance in this case is $RI_{act}(l)$.

4. The model predicts that the driver is non-compliant while the driver is actually non-compliant. The absolute deviation in compliance in this case is $|C_{act}(l) - C_{pred}(l)| = |RI_{act}(l) - RI_{pred}(l)|$.

Under the assumption of independent occurrences of F_l, $\forall l \in [1, L']$ and according to the four cases shown above, Sc_{error} can be written as:

$$
\begin{aligned}
Sc_{error} = \alpha.\frac{1}{L'}\sum_{l=1}^{L'}|RI_{act}(l) - RI_{pred}(l)| \\
+ \gamma.(P(NonC|C).\frac{1}{L'}\sum_{l=1}^{L'}RI_{pred}(l) \\
+ P(C|NonC).\frac{1}{L'}\sum_{l=1}^{L'}RI_{act}(l) \\
+ P(NonC|NonC).\frac{1}{L'}\sum_{l=1}^{L'}|RI_{act}(l) - RI_{pred}(l)|)
\end{aligned}
\tag{4.29}
$$

where $P(NonC|C)$, $P(C|NonC)$, and $P(NonC|NonC)$ are, respectively, the probability of the driver being classified as non-compliant given that the driver is actually compliant, the probability of the driver being classified as compliant given that the driver is actually non-compliant, and the probability of the driver being classified as non-compliant given that the driver is actually non-compliant. The mean of those probabilities is empirically calculated from the data-set for the training and validation sets using the confusion matrices shown in tables 4.7 and 4.8, respectively.

Table 4.7: Confusion matrix for training set compliance classification

Actual \ **Predicted**	Complaint	Non-compliant
Compliant	168	29
Non-compliant	4	1551

The expected value of the absolute score error is then computed using equation 4.29. Figure 4.6 depicts the $10-fold$ cross-validation Whisker plot for Sc_{error}, where α and γ are set to 0.5. The figure shows that the average Sc_{error} for the validation set is 9.5%, which means that, on average, the risk score of a driver in a captured

Table 4.8: Confusion matrix for validation set compliance classification

Actual \ Predicted	Complaint	Non-compliant
Compliant	67	28
Non-compliant	11	654

event will be deviated from the true value by 9.5%.

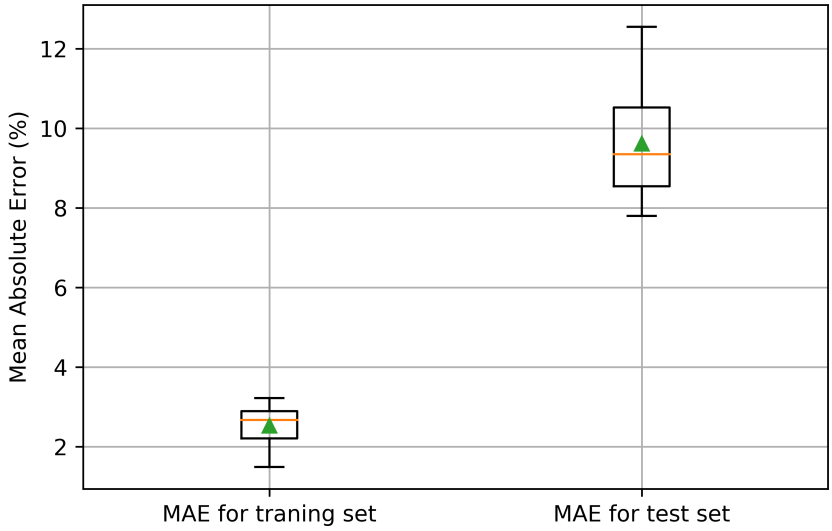


Figure 4.6: Whisker plot for the mean absolute event score error using $10 - fold$ cross-validation.

4.5 Illustrative Example

In this section, an explanation of the trip scoring process for a subject driver using the proposed risk scoring system is provided through an explanatory example. Table 4.9 displays the details of a driving trip composed of nine captured driving events. The two weighing factors α and γ are set in this example to 0.5, which means that

for a captured event, the risk score of the sd will be calculated by equally considering the risk index of the event and the driver's compliance to a warning.

For the first event, the driver's behavior is classified as "normal" and the driver is consequently assigned the full score of 1. The driver's score in the second event is calculated based on the event's risk index and the driver's compliance observed in the third event. The driver receives the full compliance score since he/she changed behavior to "normal."However, given the high risk imposed by the driver's behavior in the second event (i.e., $RI(k) = 0.75$), the overall score is calculated as: $\alpha.(1 - 0.75) + \gamma.1 = 0.5 \times 0.25 + 0.5 = 0.625$. The predicted score in this case coincides with the actual score with no error. During the fourth event, the driver was excessively speeding. In this case, there was a 25% deviation from the actual score given that the actual and predicted risk indices are 0.25 and 0.5, respectively. The driver was not compliant in this case since he/she did not change behavior to "normal" nor the risk index was zero during the following event. Consequently, the score was calculated solely based on the event risk index. The overall absolute deviation in the $sd's$ score during this trip is: $|0.79 - 0.75| \times 100\% = 4\%$.

4.6 Summary

In this chapter, a novel driver risk profiling framework was presented and discussed. The information flow among three different computational layers (i.e., the device, edge/fog, and cloud layers) in the proposed profiling system was investigated. The risk, scoring, and profiling notions were mathematically defined and explained. The chapter addressed the risk prediction problem by utilizing the behavioral and environmental contextual information of $29,000$ driving events, using the SHRP2 NDS.

Data pre-processing and model selection processes were performed to achieve the best possible prediction performance. By analyzing the mean absolute error of different models, a customized randomized trees model appears to give the best bias-variance trade-off. Results confirm that behavioral and environmental data are together good predictors of driving risk, which is measured in this chapter in terms of crash, near-crash and crash-relevant events. The developed model was then utilized to calculate the average error between predicted and actual risk indices and the average overall risk score error. An explanatory example of the risk scoring process using the proposed framework was provided. The results clearly show the robustness and effectiveness of the proposed profiling system in assigning accurate and representative risk scores for drivers.

In the next chapter, the personalized risk profiles of drivers are used in suggesting individualized safety-based routing options that aim to minimize the driving risk of individuals by considering their different skillfulness levels in various driving environments.

Table 4.9: An illustrative example of trip scoring for an *sd* using proposed risk scoring system

Event Index	Behavior	Traffic Flow	Traffic Density	Traffic Control	Weather	Lighting	Road Alignment	Actual Score	Predicted Score
1	Normal	Divided	Stable	No	No Adverse Conditions	Dark	Straight	1	1
2	Illegal or unsafe lane change or turn	Divided	Stable	No	No Adverse Conditions	Dark	Straight	0.625	0.625
3	Normal	Divided	Stable With Restrictions	No	No Adverse Conditions	Dark	Curved	1	1
4	Excessive Speeding	Divided	Stable	No	No Adverse Conditions	Lighted	Curved	0.75	0.5
5	Excessive Speeding	Divided	Stable	No	No Adverse Conditions	Lighted	Curved	0.875	0.75
6	Driving Slow	Not Divided	Stable	No	No Adverse Conditions	Lighted	Straight	1	1
7	Aggressive Driving	Divided	Stable With Restrictions	No	No Adverse Conditions	Lighted	Straight	0.25	0.25
8	Aggressive Driving	Divided	Stable With Restrictions	No	No Adverse Conditions	Lighted	Straight	0.625	0.625
9	Normal	Divided	Stable With Restrictions	No	No Adverse Conditions	Lighted	Straight	1	1
							Trip Score =	**7.9/10**	**7.5/10**

Chapter 5

iRouteSafe: Personalized Cloud-Based Route Planning Based on Drivers' Risk Profiles

Car accidents are one of the leading causes of human fatalities worldwide. Given the variation in capabilities of drivers in different driving conditions, a personalized safety-based routing - that considers the variation in driving skills - is a step towards minimizing drivers' individual and aggregate risk. In this chapter, we propose iRouteSafe, a novel cloud-based route planner that utilizes drivers' individualized risk profiles in suggesting routing options based on drivers' personal skillfulness levels. Using graph theory concepts, the routing problem is formulated as a combinatorial joint optimization problem where the objective is to find the optimal route that minimizes cost function composed of a route's travel time, expected risk, and the personal driver-specific risk in such driving routes. To highlight the significance of the proposed route planning, a case study is presented.

5.1 Introduction

Despite the recent safety measures that are being adopted by governments and car manufacturers to ensure safe driving, the road traffic death rate is still high. The 2018 global status report on road safety issued by the WHO indicated that 1.35 million people across the world are losing their lives every year due to road injuries [12]. Such a significant fatality number has made road injuries the eighth global cause of death in 2016. Moreover, the report dictates that most countries spend approximately 3% of their GDP to cover road crash expenses in the form of injury treatment, helping bereaved families, etc. [75]. With these alarming statistics, more innovative proposals are needed to minimize road crash rates.

Considering the effect that driving conditions can have on drivers, providing them with the choice to avoid driving in risky environments could certainly mitigate crash risk. Current navigation systems only provide route suggestions based on travel time or distance, hence, safety-based routing systems that suggest routes based on their expected risk are needed [76]. Safety-based routing terminology comes in different levels of abstraction. A general definition of such terminology is to find the safest route between a source and a destination among several potential routes based on the expected crash risk of each route. A common approach to predict such risk is through analyzing the crash records of roads with similar static (e.g., road alignment, traffic control) and dynamic (e.g., traffic density, weather conditions) environmental attributes. Although this approach covers the safety-based routing notion from a holistic perspective, it ignores the variation in the personal driving skill levels of drivers in the same driving conditions.

With the recent advancements in vehicular sensing technologies [77] and low-cost platforms such as OBD and smartphone sensors [78], the accurate detection of various driving behaviors and the ability to profile drivers has become affordable [3]. Furthermore, the recent developments in Vehicle-to-Cloud (V2C) [79, 80, 81], and cloud computing technologies have made it easy to send detected behaviors from vehicles to a cloud and couple the detected behaviors with the real-time environmental context as they occurred [82]. Such coupling paves the road to an environmental-aware driver profiling which measures the individual competence levels of drivers in different driving environments as discussed in chapter 4. With this information stored in the cloud, a personalized safety-based route planning that considers the individual risk profile of a driver is now possible.

In this chapter, the personalized safety-based route planning is formulated as a joint optimization problem in which the cost function is composed of the travel time of a route, and weighted general and personal expected risks taking such a route. The main contributions of this chapter are summarized as follows:

1. A novel personalized safety-based routing framework that is founded on the personal risk profiles of drivers in various driving environments is proposed with the underlying in-vehicle and on-cloud processes.

2. The personalized safety-based routing problem is formulated as a joint optimization problem and a possible solution is provided using a linear programming approach.

3. A real-world case study from the province of Ontario, Canada is presented and discussed to demonstrate the difference between current and proposed routing systems and to highlight the importance of the proposed framework.

The remainder of this chapter is organized as follows. In section 5.2, background and related work are presented. Section 5.3 discusses the proposed cloud-based routing framework with the underlying sub-systems. In section 5.4, the route planning optimization problem is formulated and discussed. Section 5.5 presents a real-world case study to demonstrate the proposed routing system. The summary is given in section 5.6.

5.2 Background and Related Work

Current popular route planning systems such as Google Maps primarily rely on the expected travel time in suggesting routes. Vehicle routing based on the estimated travel time problem has been extensively covered in the literature. The main objective in this problem revolves around finding the optimal route that has the minimum overall travel time among a number of potential routes given the static and dynamic attributes of the route [83]. Likewise, Waze application provides route planning choices based on which route has the shortest distance or travel time. Waze issues real-time traffic warnings such as car accidents based on information inputted by drivers [84].

Eco-route planning has been recently studied in the literature. In [85], authors proposed a cloud-based system that provides heavy duty vehicles with optimal routes that minimizes fuel consumption while satisfying a constraint on the maximum travel time. Moreover, the choice of the optimal route that jointly minimizes travel time and a driver's discomfort is presented in [86]. Discomfort was measured in terms of road roughness, anomalies, and the number of intersections. Recently, authors in [87] proposed a fuzzification route recommendation system that suggests a route based on the condition of its segments.

Safety-based route planning has also been studied in literature. In [88], the author discussed an envisioned IoT-based framework that is expected to facilitate the employment of safety-based routing. Authors in [89] proposed a risk prediction model that utilizes a large-scale road and crash dataset to predict crash rates in road segments based on eight static road features. Moreover, to compensate for the dynamic real-time factors (e.g., weather condition) that were not present in the dataset, authors introduced ad-hoc correction factors to be applied on the proposed prediction model. More recently, authors in [90] proposed SafeRNet, a safety-based routing system. The system is built on a customized Bayesian inference model that is able to predict the crash risk in a road segment based on both static and dynamic features.

Despite the research efforts mentioned above, to the best of our knowledge a safety-based route planner that takes into account the individual differences in driving competence levels among drivers in different driving environments is still missing. For instance, although curved roads with high traffic density and foggy weather conditions could be risky from a holistic standpoint, drivers may have various risk rates in such an environment depending on their personal competence levels. Using the individual risk profile of a driver in calculating his/her personal overall risk in different routes is presented and thoroughly explained next.

5.3 iRouteSafe: System Architecture

In this section, we present an overview of the personalized safety-based routing system followed by an explanation of the individual safety-based system's components.

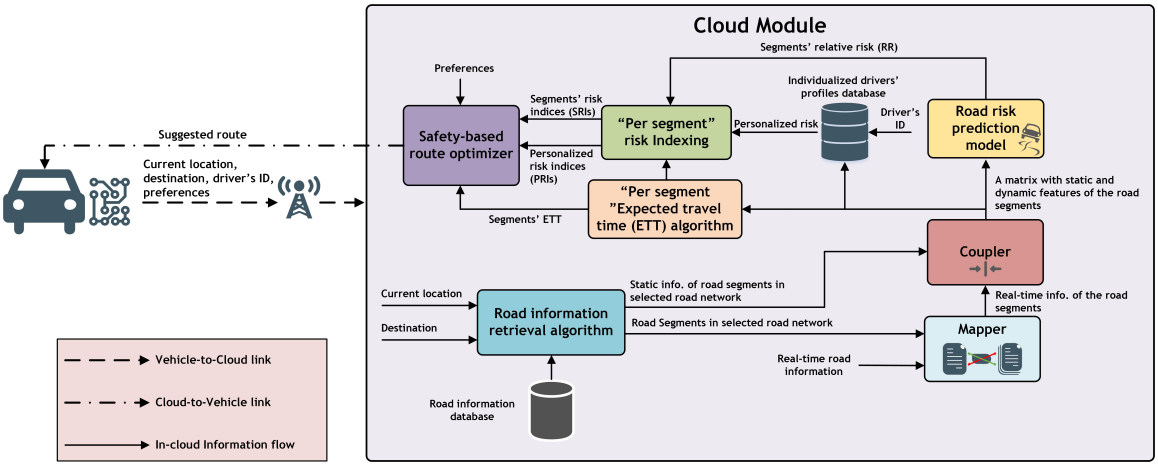


Figure 5.1: iRouteSafe: proposed personalized safety-based route planning system.

5.3.1 Overview

Figure 5.1 depicts the proposed iRouteSafe system's architecture. In the proposed iRouteSafe system, the route planning process is initiated by the subject driver (sd) who communicates his/her current GPS co-ordinates, desired destination, identification number, and desired personalized routing preferences to the cloud through a cellular wireless link. An sd can choose a route based on the minimum expected travel time (ETT), minimum risk (from both personalized and holistic perspectives), or based on the joint inclusion of these preferences in the route's optimization cost function.

On the cloud, the $sd's$ GPS current location (source) and desired destination are inputted to a road information retrieval module which retrieves the potential road segments (R) from source to destination as abstracted directed graph edges, and the segments' corresponding static features (Env_s) from the road information database (e.g., In Canada, the road information database is built through accessing the National Road Network (NRN) Canada, which includes road segment information

such as location, name, type, direction, address range, rank and class [91]). Then, with an access to the real-time road information, a mapping function f matches the potential road segments with their corresponding real-time information (Env_d) including their weather conditions, traffic density levels, and lighting conditions.

$$f : R \rightarrow Env_d \tag{5.1}$$

After that, the static and dynamic features of each potential segment are merged together through a coupler in a matrix structure ($Env = [Env_s, Env_d]$) with each row representing the overall features of one potential road segment. Given the static and dynamic features of potential road segments, a trained supervised risk prediction model predicts the relative risks (RRs) of the segments, where RR is calculated as the relative crash and near-crash risk probability as further explained in section III.B. Moreover, to retrieve the personalized competence levels of the sd for the potential road segments, Env is fed to a database containing updated risk scores of the sd in road segments with similar features to the potential road segments. The personalized risk scores of the sd in the potential road segments are extracted from the personalized driver profile database given the identification number of the sd. Also, Env is utilized to calculate the expected travel times of the road segments ($ETTs$). The information of the general RR, personalized risk scores, and $ETTs$ of potential road segments is then passed to the "per segment risk indexing" module which hosts two utility functions $\mathcal{F}(.)$ and $\mathcal{G}(.)$ that respectively assign two risk indices RI_{gen} and RI_{per} for each road segment corresponding to its general and personalized risks. The calculation of RI_{gen} and RI_{per} is detailed in section III.D. Finally, the segment $ETTs$, general and personalized risk indices are provided to the proposed joint safety-based route

optimizer which, based on the $sd's$ preferences, calculates the optimal route and sends it back to the sd.

5.3.2 Road risk prediction model

Road risk in the context of this chapter refers to the general risk imposed on a driver when exposed to a certain road environment, regardless of the personal driving skills of that driver. By definition, such risk does not vary between drivers as it solely depends on the road's architecture and dynamic features.

In this chapter, SHRP2, a large-scale naturalistic driving study [50], is utilized to develop the road risk prediction model. Using the environmental context during such events as risk predictors, the risk prediction is defined as the process:

$$\mathcal{F} : Env_i \rightarrow RR(Env_i) \tag{5.2}$$

where $RR(Env_i)$ is defined as the relative risk of Env_i in SHRP2 dataset and is mathematically expressed as:

$$RR(Env_i) = \frac{P(Risk|Env_i)}{P(Risk|Env_i^{'})} \tag{5.3}$$

where $P(Risk|Env_i)$ is the risk probability given the exposure to driving environment i, and $P(Risk|Env_i^{'})$ is the risk probability in all other environments except i. Risk probability in a certain driving environment is calculated in terms of the number of crash, near-crash, and baseline events as:

$$P(Risk|Env_i) = \frac{C_i + NC_i}{B_i + C_i + NC_i} \tag{5.4}$$

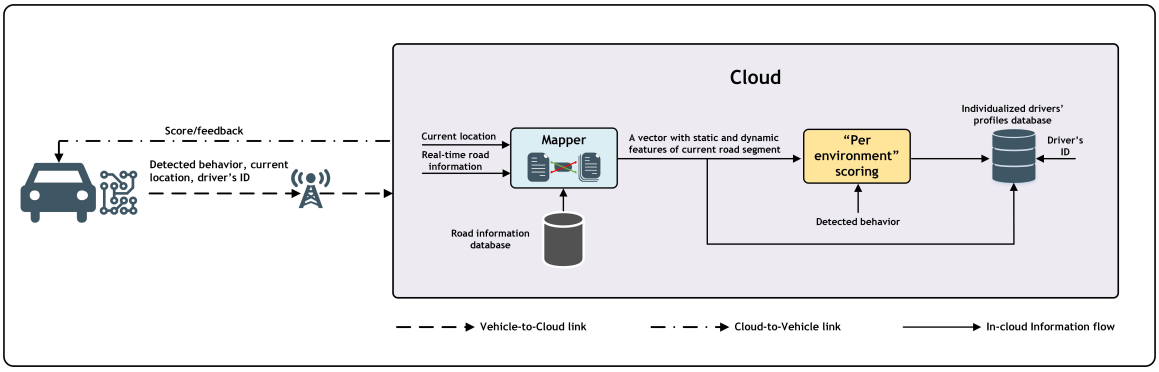


Figure 5.2: Driver profiling update after each driving trip.

where C_i, NC_i, and B_i are respectively the number of crash, near-crash, and baseline events captured in driving environment i. Since $P(Risk|Env_i)$ depends on the sampling rate at which the baseline events are taken, the relative risk probability rather than risk probability has been adopted as a risk measure.

In this work, an RF regressor with 100 decision trees and MSE splitting criterion is trained using SHRP2 data samples to reflect the relative risk of different driving environments [82]. Table 5.1 depicts the considered environmental road features.

5.3.3 Individualized drivers' profiles database

As discussed in chapter 2, driver profiling is a dynamic personalized process that targets the detection of a driver's competencies based on his/her driving behaviors. Figure 5.2 depicts a summary of the driver profiling process which starts by communicating a detected behavior, $sd's$ current location and identification number to the cloud. Behavior detection is usually performed inside the vehicle by utilizing the in-vehicle sensors such as smart-phone sensors (e.g., accelerometers, gyroscopes, GPS) or OBD units. Using such data, behaviors are categorized using a multi-class classifier.

On the cloud, static and real-time features of the road segment where the behavior is detected are extracted and fed to the "per environment" scoring module. This module hosts a trained risk prediction model which predicts risk based on the joint effect of the detected behavior and its environmental context. The scoring module also hosts a feedback sub-module which compares the relative risk of detected behavior to an ad-hoc threshold. A warning is issued to the sd during a driving trip if the relative risk of the detected behavior is high. Based on the average relative risk of different detected behaviors and the $sd's$ compliance to warnings, the $sd's$ "per environment" profile is updated by the end of each trip. The risk score (RS) of sd in a driving environment Env_i is expressed mathematically as:

$$RS_{sd}(Env_i) = \max\left(RR_{sd}(Env_i) - \beta.C_{sd}(Env_i), 0\right) \tag{5.5}$$

where $RR_{sd}(Env_i)$ and $C_{sd}(Env_i)$ are respectively the average relative risk and compliance of driver sd in driving environment Env_i, and β is a weighting factor the system administrator chooses to specify the importance of $C_{sd}(Env_i)$ in the calculation of $RS_{sd}(Env_i)$ [82].

5.3.4 Per segment risk indexing

Two risk indices, SRI and PRI, are assigned to a road segment r based on the RR of the segment and the $sd's$ personal RS in that segment, respectively. SRI is assigned based on two factors. The first is the ETT of the segment which reflects how much time the sd will be exposed to the risk imposed by r, while the second is the RR

value of the segment as expressed in equation 5.6.

$$SRI(r) = \mathcal{F}(ETT(r), RR(r)) \tag{5.6}$$

Since the risk of the segment has a positive relationship between its ETT and RR values, the SRI utility function can in the form:

$$SRI(r) = (ETT(r) \times RR(r))^{n_1} \tag{5.7}$$

$ETT(r)$ can be viewed as a factor that weighs the risk of the segment based on the $sd's$ exposure to that risk. n_1 is an integer chosen by the system administrator that determines the effect of the risk of individual route segments on the choice of the overall optimal route. For instance, considering two potential routes $R1$ and $R2$. $R1$ may have a smaller sum of weighted relative risks compared to $R2$ but still be avoided if n_1 is large in case that $R1$ contains a segment or more with very high weighted relative risk.

Similarly, PRI is assigned based on the ETT of a segment and the $sd's$ personal risk score RS in that segment (or in another segment with similar environmental features). The truthfulness (TR) of the RS score is another weighting factor that is utilized to calculate PRI. TR value depends herein on the total exposure time of the sd driver in a driving environment similar to the segment's environment. $TR \in [0, 1]$, with a value of 1 indicating full truthfulness. The mathematical expression of the personal risk index PRI for sd is shown in equation 5.8.

$$PRI_{sd}(r) = (ETT(r) \times TR_{sd}(r) \times RS_{sd}(r))^{n_2} \tag{5.8}$$

5.4 Personalized Safety-Based Routing

In this section, the proposed route planning problem is formulated using graph theory and Linear Programming (LP). Planning a route between a source and destination in a road network can be modelled as a digraph where nodes in the graph resemble road network intersections while edges represent road segments.

A digraph is formally represented by the tuple G, where $G = (\mathcal{N}, \mathcal{E})$. $\mathcal{N}$ and $\mathcal{E}$ are respectively representing the set of all nodes in the graph, and the set of all edges (i.e., ordered pairs of nodes) in the graph.

In our proposed route planning problem, each edge ε_{n_i,n_j}, where $\varepsilon_{n_i,n_j} \in \mathcal{E}$, is uniquely characterized by the 3-tuple $(t_{\varepsilon_{n_i,n_j}}, SRI(\varepsilon_{n_i,n_j}), PRI_{sd}(\varepsilon_{n_i,n_j}))$, where $n_i, n_j \in \mathcal{N}$ are any two consecutive nodes in the graph with a direct path, t_{n_i,n_j} is the expected travel time between n_i and n_j, $SRI(\varepsilon_{n_i,n_j})$ is the segment risk index of ε_{n_i,n_j}, and $PRI_{sd}(\varepsilon_{n_i,n_j})$ is the personalized risk index of sd in ε_{n_i,n_j}.

A path from source to destination in the digraph is a sequence of edges (road segments in our problem) starting from source and ending to destination. Let P denote a matrix containing all paths, where P_l is a vector of nodes that form a possible path in P. The personalized safety-based routing problem is formulated as a combinatorial joint optimization problem as shown in equation 9:

$$\min_{l} \sum_{i=0,j=i+1}^{i=M-1,j=M} t_{\varepsilon_{P_l(i),P_l(j)}} + \gamma_1 . SRI(\varepsilon_{P_l(i),P_l(j)}) + \gamma_2 . PRI_{sd}(\varepsilon_{P_l(i),P_l(j)}) \tag{5.9}$$

where $P_l(i)$ is the node that corresponds to the i_{th} index of P_l, M is the last node in

path P_l which is the destination node, γ_1 and γ_2 are weighting factors which reflect how much importance is given to the safety terms. So the problem is to find the integer l which corresponds to the optimal path P_l.

The problem is further formulated as a Linear Integer Programming (LIP) problem. The binary variable x_{n_i,n_j} is defined as follows:

$$x_{n_i,n_j} = \begin{cases} 1, & \text{if } \varepsilon_{n_i,n_j} \text{is a segment in the optimal path} \\ 0, & \text{otherwise} \end{cases} \tag{5.10}$$

And let $C(\varepsilon_{n_i,n_j})$ be the cost of travelling in edge ε_{n_i,n_j} which is expressed as:

$$C(\varepsilon_{n_i,n_j}) = t_{\varepsilon_{n_i,n_j}} + \gamma_1.SRI(\varepsilon_{n_i,n_j}) + \gamma_2.PRI_{sd}(\varepsilon_{n_i,n_j}) \tag{5.11}$$

So the LIP problem can be formulated as:

$$\text{Minimize} \quad \sum_{\forall \varepsilon_{n_i,n_j} \in \mathcal{E}} C(\varepsilon_{n_i,n_j}).x_{n_i,n_j} \tag{5.12}$$

$$\text{subject to} \quad \sum_{\forall \varepsilon_{n_0,n_k} \in \mathcal{E}} x_{n_0,n_k} = 1 \tag{5.13}$$

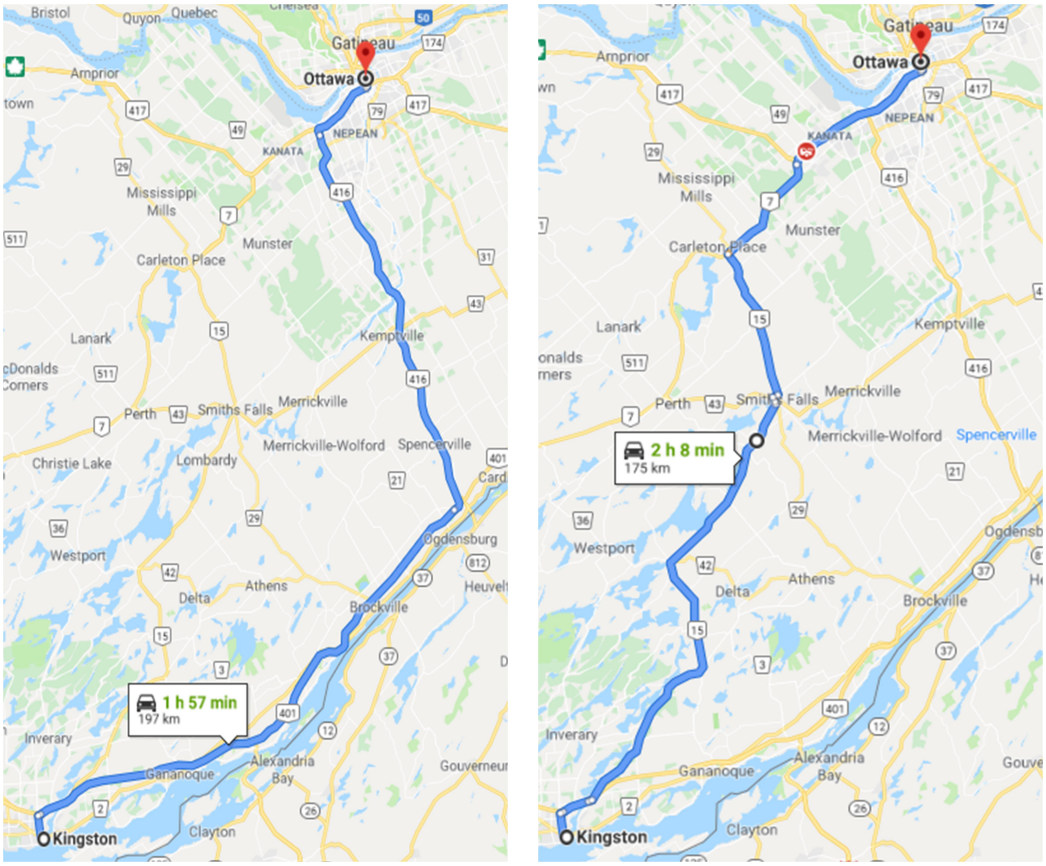
$$\sum_{\forall \varepsilon_{n_i,n_j} \in \mathcal{E}} x_{n_i,n_j} - \sum_{\forall \varepsilon_{n_j,n_m} \in \mathcal{E}} x_{n_j,n_m} = 0 \tag{5.14}$$

$$\sum_{\forall \varepsilon_{n_k,n_M} \in \mathcal{E}} x_{n_k,n_M} = 1 \tag{5.15}$$

$$\sum_{\forall \varepsilon_{n_i,n_j} \in \mathcal{E}} \gamma_1.SRI(\varepsilon_{n_i,n_j}) + \gamma_2.PRI_{sd}(\varepsilon_{n_i,n_j}) < s_{th} \tag{5.16}$$

where the constraints in equations 13 and 15 are necessary to ensure that there is only one arc leaving the source node and only one arc arriving to the destination node, respectively. The constraint in equation 14 is important to ensure the path continuity where n_j is any intermediate node in the graph (i.e., $n_j \neq n_0$ and $n_j \neq n_M$). The constraint in equation 16 defines a user-specific safety constraint for which a route is avoided if its total risk is above s_{th}.

5.5 Case Study



(a) Optimal route according to Google Maps. (b) Optimal route according to iRouteSafe.

Figure 5.3: Route planning case study in Ontario, Canada.

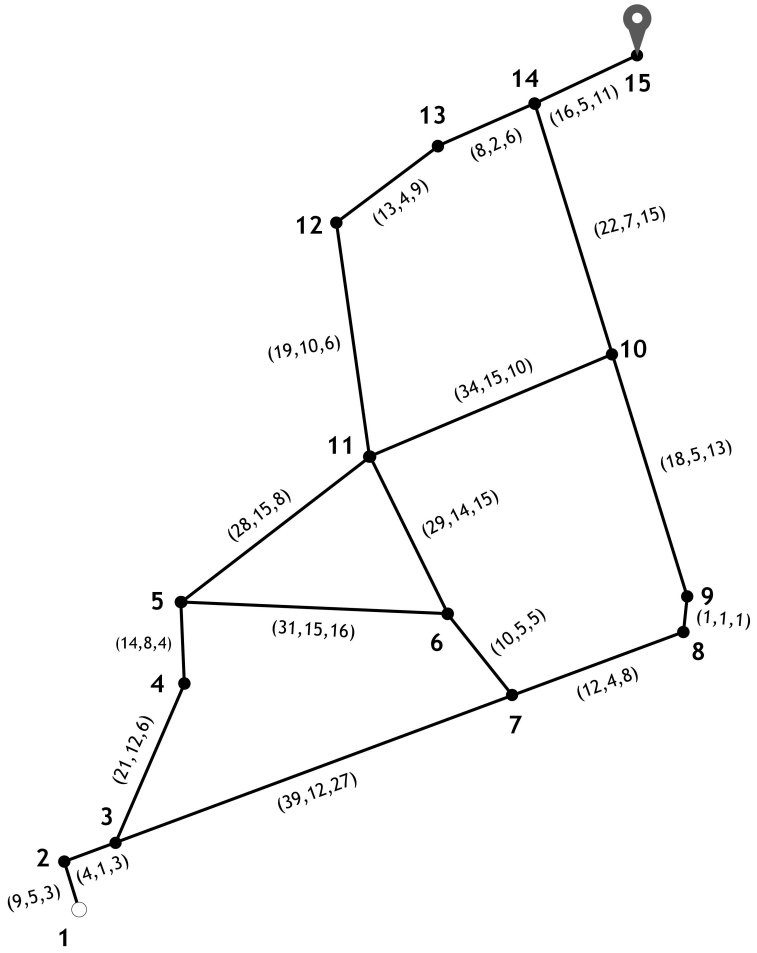


Figure 5.4: Road network as a graph.

In this section, a route planning case study which highlights the effectiveness of the proposed route planning scheme is discussed. The case study is from the province of Ontario in Canada where the requested route is from the city of Kingston to Ottawa. The trip request was performed on Sunday, June 2nd at 10:15 PM EDT.

Figure 5.3(a) depicts the proposed Google Maps route. Considering the real-time traffic and road conditions, the selected Google Maps route resulted in the minimum expected travel time. Figure 5.4 shows the extracted graph that represents the road network. In this figure, the 3-tuple presented on each road segment represents the

expected travel time of the segment, general risk index, and personalized risk index, respectively. General and personalized risk indices were generated using equations 5.7 and 5.8 considering both the static and real-time environmental features shown in table 5.1. The nodes in this figure resemble the major road intersections.

To choose the optimal route which jointly considers the travel time and risk, we used Gurobi optimizer [92] to solve the optimization problem in equations 12 through 15. The optimization parameters that are used in this case study are presented in table 5.2.

Table 5.2: Optimization parameters of the case study

Optimization Parameter	**Value**
γ_1	1
γ_2	2
n_1	1
n_2	1
TR	$\overrightarrow{1}$
s_{th}	210

In table 5.2, the values of γ_1 ad γ_2 shows that more weight was given to the personalized risk index of the subject driver than the general segment risk index. Also road segments in this case study are linearly penalized for their SRI and PRI values as indicted form n_1 and n_2 values. The optimal iRoutesafe route follows Figure 4 node sequence 1-2-3-4-5-11-12-13-14-15 and is depicted in Figure 5.3(b).

5.6 Summary

In this chapter, a novel cloud-based route planning framework was presented. In the proposed framework, the system selects the route which jointly minimizes the expected travel time and the risk from a holistic and personalized perspectives. Using static and dynamic environmental attributes, a customized regressor was trained to reflect the expected relative risk of road segments. The novelty in the proposed framework appears in the incorporation of the personalized drivers' risk profiles in the calculation of the overall route risk. Taking to account such variation in drivers' skillfulness levels in the same driving environment is certainly crucial to minimize the aggregate risk.

Using graph theory and linear programming, the problem was formulated as an LIP problem. To highlight the effectiveness of the proposed system, Gurobi optimizer was utilized to solve a real route planning problem from the province of Ontario in Canada.

Table 5.1: Features of driving environments

Static features			Dynamic features		
Traffic flow	**Traffic control**	**Alignment**	**Weather**	**Traffic density**	**Lighting**
Divided	No traffic control	Straight	Normal	Free flow	Darkness; lighted
Not-divided	Traffic signal	Curved left	Fog	Flow with some restrictions	Darkness; not lighted
One-way traffic	Traffic sign	Curved right	Mist	Unstable flow	Dawn
No lanes	-	-	Rain and fog	Forced traffic flow conditions	Daylight
-	-	-	Rain	-	Dusk
-	-	-	Sleet	-	-
-	-	-	Snow	-	-
-	-	-	Snow/sleet and fog	-	-

Chapter 6

Profiling Based on Fault Inference During Risky Events

Proposed risk profiling frameworks in chapters 3 and 4 are based on assigning risk scores for drivers given the expected risk of their driving behaviors. Another approach is to profile drivers based on the actual risky events (i.e., crashes or near-crashes) they were involved in and their fault contribution in such events.

In this chapter, an additional level of classification in the hierarchy of profiling is proposed. Using the 100-CAR NDS data-set, five different Hidden Markov Models are trained to determine the fault responsibility of a subject vehicle in crash or near-crash events. Two specific driving situations, which are conflicts with leading and following vehicles, are investigated in this study. Results show that these models can achieve a reasonable classification accuracy.

6.1 Introduction

Profiling drivers solely based on the expected risk of their driving behaviors without considering the actual risky events they were involved in and whether they were faulty

or non-faulty can be misleading. That is because, from a personal perspective, drivers may vary in their responses towards driving conflicts and in their skillfulness levels. For instance, some drivers show an extremely careful attitude even in non-risky events (e.g., no crash or near-crash). This, in turn, causes them to do more frequent harsh braking than other drivers. Although harsh braking behavior may be associated with high risk probability from a holistic perspective, it may not be risky for these specific drivers with such a careful attitude. These drivers will be unfairly profiled as they will have similar risk profiles to careless and inattentive drivers. Consequently, adding another level in the hierarchy of profiling that considers the detection of the **actual risky events** (e.g., crash or near-crash events) and the fault contribution of the subject driver in such events is important. That is achieved through modified techniques that incorporate driving scene variables, which reflect the actual behavior of individual drivers in risky situations as is explained in this chapter.

The main contributions of this chapter are as follows:

1. A novel driving fault inference profiling system is proposed. The system compromises the data filtering and pre-processing, risky event detection, time windowing, semantic analysis of detected risky events to classify the subject driver's behavior, and risk profiling.

2. Fault inference is formulated as a sequence modeling problem and tackled using HMM-based modelling approach.

3. HMM models are trained and validated using a large-scale NDS (i.e., 100 CAR NDS). Two specific driving conflicts are studied. Results show the effectiveness of our approach in classifying drivers.

Despite the efforts in driving behavior classification (see section 2), models that are capable of capturing drivers fault contribution in creating risky events are still a void. To this end, the work in this chapter is meant to be a step towards a comprehensive novel classification approach in which drivers are classified based on their fault contribution in different risky driving situations.

In this chapter, an HMM-based modeling approach is deployed to determine the fault responsibility of a subject driver during crash or near-crash events. Five unique HMMs, representing five classes of behaviors, are trained and validated using the 100-CAR NDS data set [93]. Three of these models are utilized to classify a subject driver when he/she is involved in conflicts with following vehicles, while the other two are used to classify conflicts with leading vehicles, in normal road and weather conditions. The rest of this chapter is structured as follows: first, the driving fault determination problem is discussed in Section 6.2. A system overview is then presented in section 6.3. In section 6.5, the fault determination classification approach is detailed and examples are provided to elaborate the idea. Section 6 explains the details of the fault inference profiling system. It comprises the data filtering and pre-processing, and the HMM-based formulation of the fault inference problem. The experimental results are then provided in section 6.6. Finally, the chapter summary is shown in section 6.7.

6.2 Problem Statement

In this chapter, we consider a traffic environment in which traffic flow is either in one or in two directions (i.e., one-direction, divided, or non-divided roadway). The objective is to automatically determine the fault responsibility of a subject driver

during risky conflicts of different types (e.g., conflicts between subject driver and following or leading vehicles) using sequential modelling. This is achieved via the analysis of the pre-incident maneuver that caused this risky event, and the re-action of the *sd* in response to that event. Figure 6.1 depicts an example of a conflict between an *sd* and a leading vehicle in a divided roadway.

By analyzing vehicle-related signals (e.g., longitudinal and lateral acceleration) as well as range related signals (i.e., radar range and range rate sequential data), different behavioral classes are assigned to the *sd*, depending on the conflict type. The classification process is detailed in section IV. This classification is in agreement with the Revised Regulations of Ontario (R.R.O.) 1990, Regulation 668, and fault determination rules, under "*Rules for Automobiles Traveling in the Same Direction and Lane*" section [94].

The accurate determination of the *sd*'s behavior during risky driving conflicts lays the foundation for more advanced profiling techniques that can be utilized by insurance companies as well as fleet administrators.

6.3 System Overview

In this section, an overview of the proposed fault inference system is provided. Figure 6.2 depicts the system architecture.

In the proposed system, the time-series data that is collected by the subject vehicle during a certain trip is off-loaded to the cloud on a per-trip basis. Collected data contains the longitudinal and lateral behaviors of the subject vehicle as well as its relative motion information with surrounding vehicles and objects. Once the data is

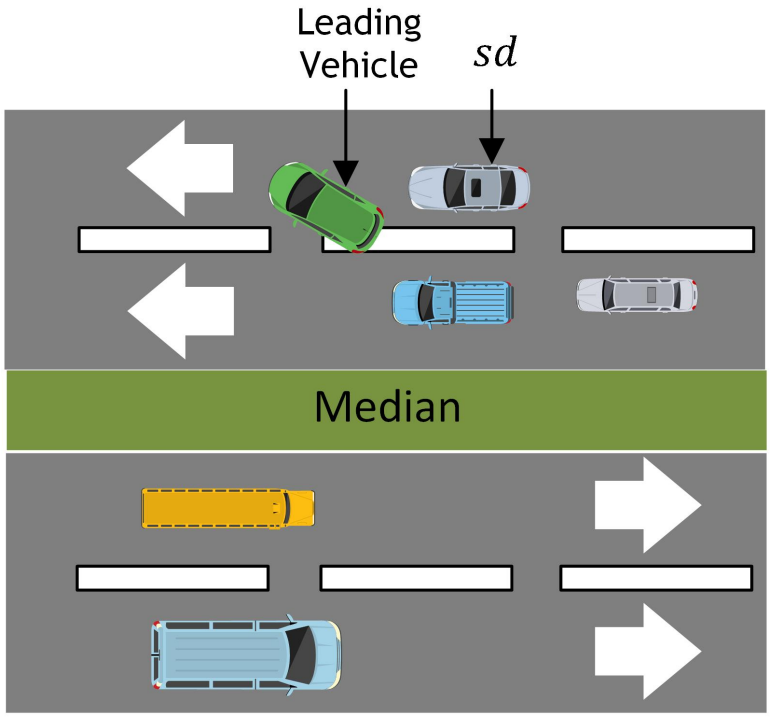


Figure 6.1: A conflict with a leading vehicle in a divided roadway.

off-loaded, it is pre-processed to remove outliers and to interpolate missing data. Afterwards, risky events of different types are detected from the data time-series stream through different triggers. For instance, data reductionists in the 100-CAR study have performed a sensitivity analysis to find the best set of triggers that leads to the best confusion matrix performance [93]. Once a risky event is detected, a pre-incident time-series data are inputted to different HMM-based models each corresponds to a unique behavioral class to infer the fault contribution of the subject driver. In this work, four distinct behavioral classes are identified as is explained in section 6.4. The choice of the pre-incident time in which the time-series data are analysed is a system's tunable hyper-parameter, and so are the number of states in the HMM models and

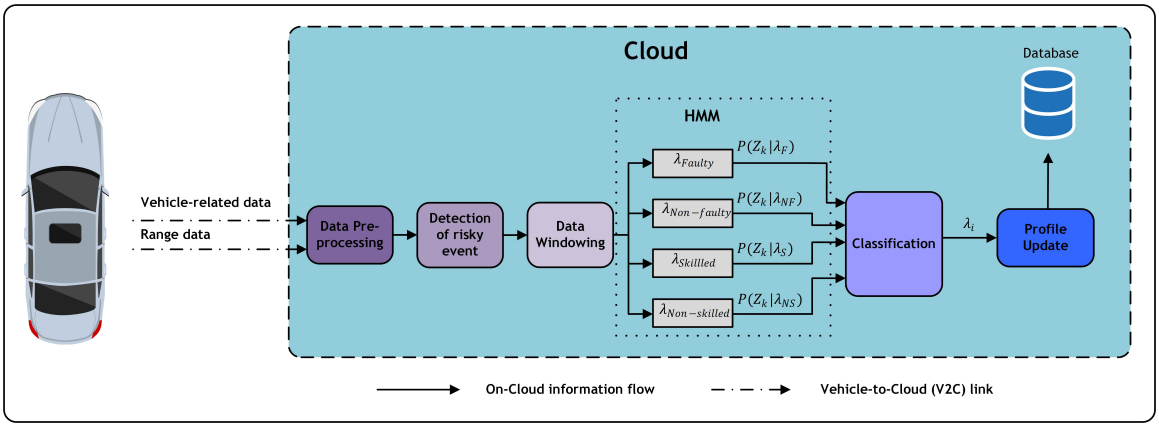


Figure 6.2: Proposed Fault Inference System.

the sampling time. Each of the HMM models outputs a posterior conditional probability ($P(Z_k|\lambda_i)$) of the observation sequence (Z_k) given the model (λ_i). Based on the posterior probabilities of the models, the driver is assigned to one of the four behavioral classes (the one with the highest posterior probability). In the next section, an explanation of the four behavioral classes in two types of conflicts (i.e., conflicts with following and leading vehicles) is provided.

6.4 Fault determination

The knowledge of longitudinal and lateral behavior as well as the relative position of the *sd* prior to and during a risky event, can provide an insight to the driver's contribution in creating such an event. This leads to a more accurate and fair classification that takes into account the whole driving scene. The classification process depends on the driving conflict type in which the *sd* is involved. In general, drivers are classified as one of the following four classes: *Faulty, Non-faulty, Skilled, and Non-skilled*. Each of these classes represents a set of unique and distinguishable observation sets.

Faulty and non-faulty classification modeling is done in accordance with the insurance act rules in the case of crash occurrence. The other two classes represent a finer classification based on the driver's re-action during the event. The latter two classes, although they are conventionally considered part of the non-faulty category, they have unique characteristics that single them out from the non-faulty class. In the rest of this section, we discuss two of the most common driving conflicts which are conflicts between the *sd* and leading/following vehicles. The devised models are built under three assumptions

1. Normal weather and road conditions.
2. Vehicles are in a one way or divided roadways.
3. Vehicles are not at an intersection.

6.4.1 Conflicts with leading vehicles (type 1)

Consider a situation where an *sd* is involved in a crash/near-crash event with a leading vehicle. Table 6.1 summarizes the description of the different *sd*'s behavioral classes. The 100-CAR data-set does not contain events that represent the non-skilled and non-faulty classes for this conflict type and under the aforementioned assumptions. The *sd* driver is classified as non-faulty when another driver cuts in too close in front of him/her, making it impossible to avoid crashing (given the current velocity and the forward range of the *sd*). On the other hand, the *sd* is non-skilled when an avoidable crash occurs due to the *sd*'s poor judgement or inattentiveness. Hence, the *sd* in this class is still considered as non-faulty since he/she did not initiate the faulty maneuver. The other two classes will be explained using two sets of real observations obtained from the data-set.

Table 6.1: Behavioral classes of an *sd* involved in a conflict with a leading vehicle.

Class	Description
Faulty	Being involved in a crash/near-crash event with a leading vehicle due to the *sd* inattentiveness/ aggressive behavior.
Non-skilled	Being involved in an avoidable crash event where the leading vehicle is faulty. The *sd* could safely take a corrective reaction to avoid the crash.
Non-faulty	Being involved in an unavoidable crash event where the leading vehicle is faulty.
Skilled	Avoiding a crash with a faulty leading vehicle by decelerating or changing lanes.

Faulty class

An *sd* is considered faulty in a conflict with a leading vehicle when he/she is not keeping enough forward distance prior to the conflict. Figure 6.3 depicts four time-series observations during a near-crash event. Figure 6.3 shows that the *sd* has kept a constant speed (i.e., zero longitudinal acceleration) even with a decreasing forward range and range rate (i.e., the *sd* is aggressively speeding). The driver had to perform a harsh braking at the last moment (i.e., at time = 6s on the Figure) to avoid rear-ending the leading vehicle.

Skilled class

The skilled behavioral class comprises *sd*s who reacts skillfully to avoid a crash with a leading vehicle that cuts in too close in front of him/her (i.e., faulty lane change maneuver). Figure 6.4 depicts the time-series data of a skilled driver during a risky

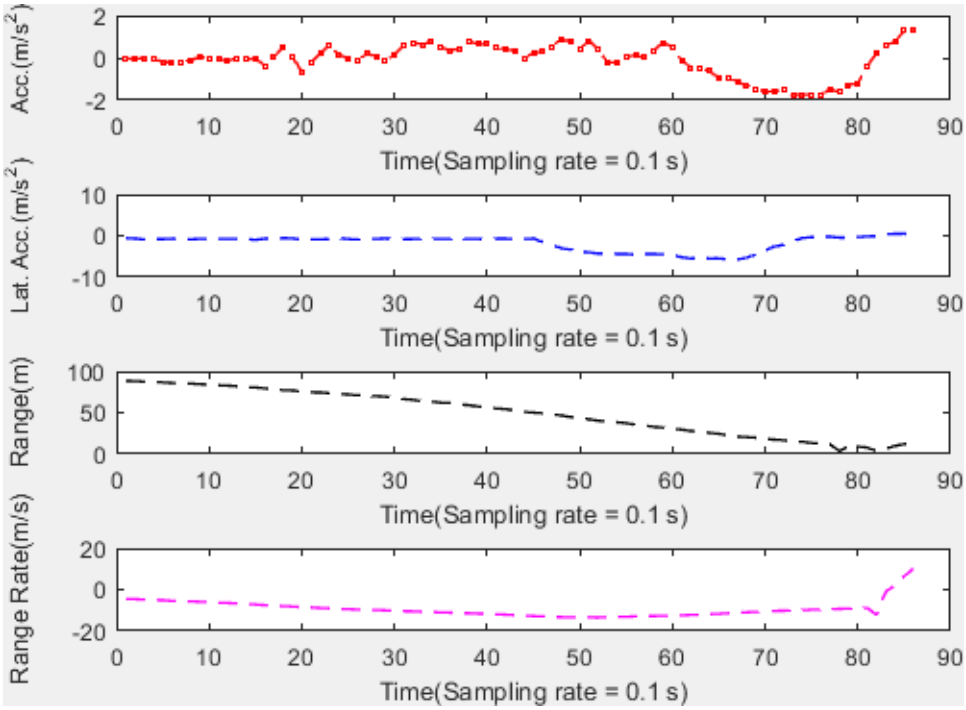


Figure 6.3: A set of observations showing a faulty *sd* during a conflict with leading vehicle

conflict with a leading vehicle. Observations show a steep change in the forward range of the *sd* at time $\approx$ 5s. The range is almost constant when it suddenly and significantly decreased. This reflects the unsafe lane change maneuver performed by a leading vehicle. The sudden change in the lateral acceleration signal shows that the *sd* has avoided the crash by immediately making a left lane change maneuver followed by a smooth longitudinal braking.

6.4.2 Conflicts with following vehicles (type 2)

In table 6.2 , a brief description of different behavioral classes of an *sd* involved in a risky conflict with a following vehicle is presented. Based on the event narrative data dictionary of the 100-CAR data set, and the description of each class, the skilled

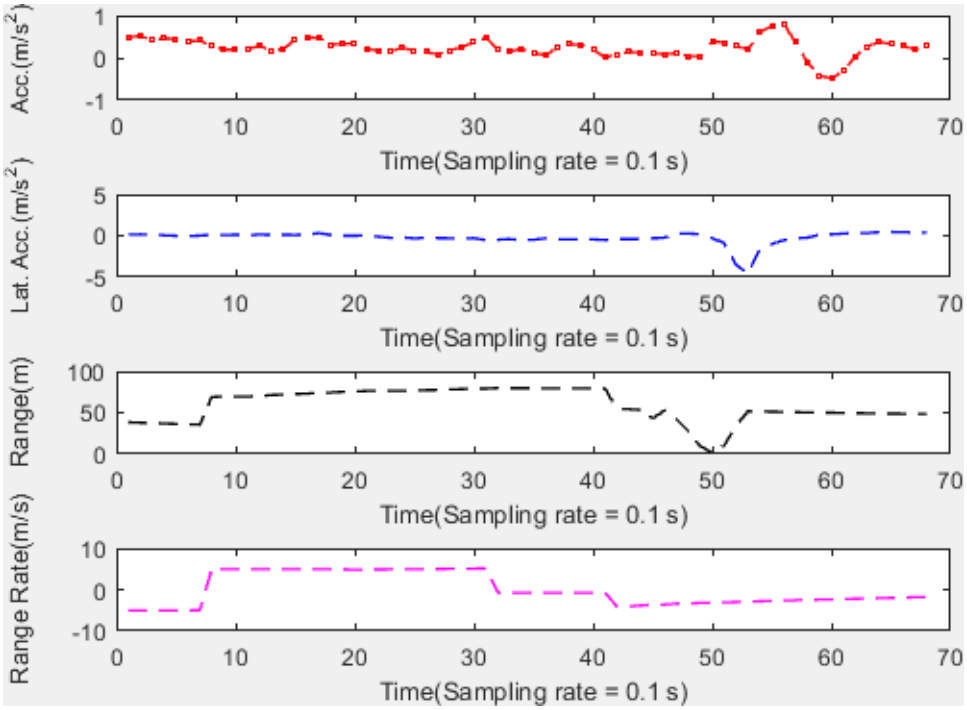


Figure 6.4: A set of observations showing a skilled *sd* during a conflict with leading vehicle

behavioral class is missing in the available data. An *sd* is considered skilled in this type of conflicts when he/she reacts to an inattentive or aggressive following vehicle by safely accelerating or changing lanes. The other three classes are detailed in the following sections.

Faulty class

An *sd* is classified as faulty in this category of conflicts when he/she changes lanes without keeping a safe gap with the following vehicle in the new lane. Figure 6.5 shows an example of a set of observations that reflects this behavior. As can be deduced from this figure, the change in the lateral acceleration at time $\approx$ 4s is followed by a steep change in the rearward range (i.e., from $R_{Rearward} \approx$ 43m to $R_{Rearward} \approx$ 5m).

Table 6.2: Behavioral classes of an *sd* driver involved in a conflict with a following vehicle.

Class	Description
Faulty	Cutting too close in front of a following vehicle causing an unsafe rearward range or a negative range rate high in magnitude.
Non-skilled	Braking too hard in front of a following vehicle causing an unsafe rearward range or a negative range rate high in magnitude.
Non-faulty	Braking smoothly. Unsafe rearward range due to the reckless behavior of the following vehicle.
Skilled	Avoiding a crash with a following vehicle by accelerating or changing lanes.

This is interpreted as a faulty lane change maneuver.

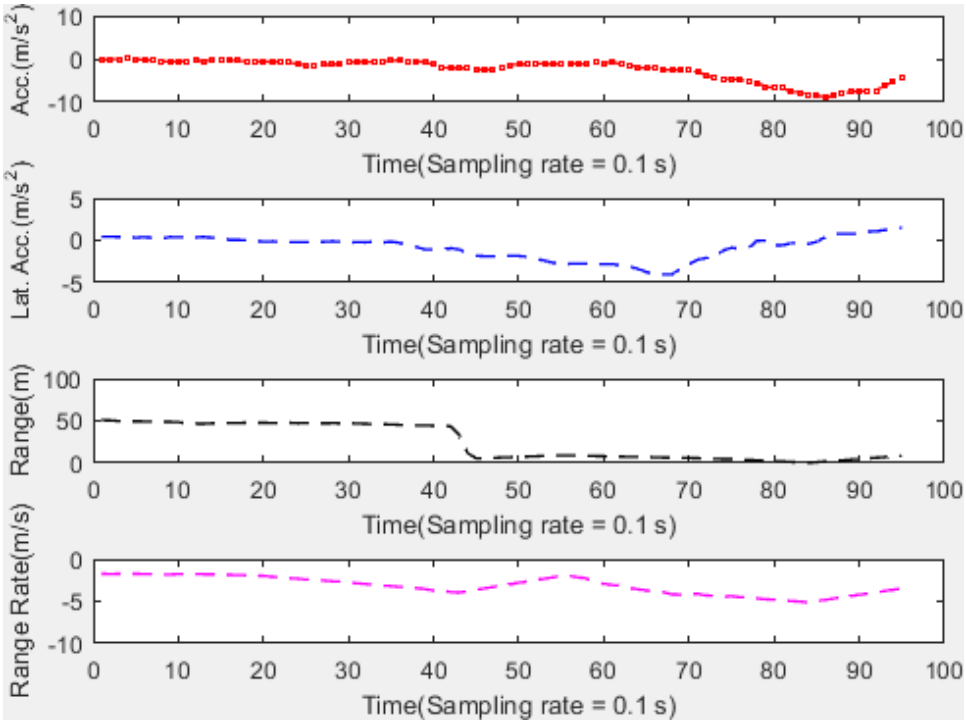


Figure 6.5: A set of observations showing a faulty *sd* during a conflict with a following vehicle

Non-skilled class

In this class, the *sd* is in the same lane of the following vehicle when he/she makes an unnecessary harsh braking, causing a crash or near-crash event with the following vehicle. Although the following vehicle is still classified as faulty, since it has to keep a safe rearward range, the conflict could be avoided if the *sd* did not make this unnecessary action. Figure 6.6 depicts a set of observations showing an unskilled behavior. The *sd* in this example performed an aggressive deceleration event at time $\approx$ 3s. This has led the following vehicle to nearly hit the *sd* in the rear as can be shown from the rearward signal. In this incident, the forward range was big enough that the *sd* did not have to take such aggressive action.

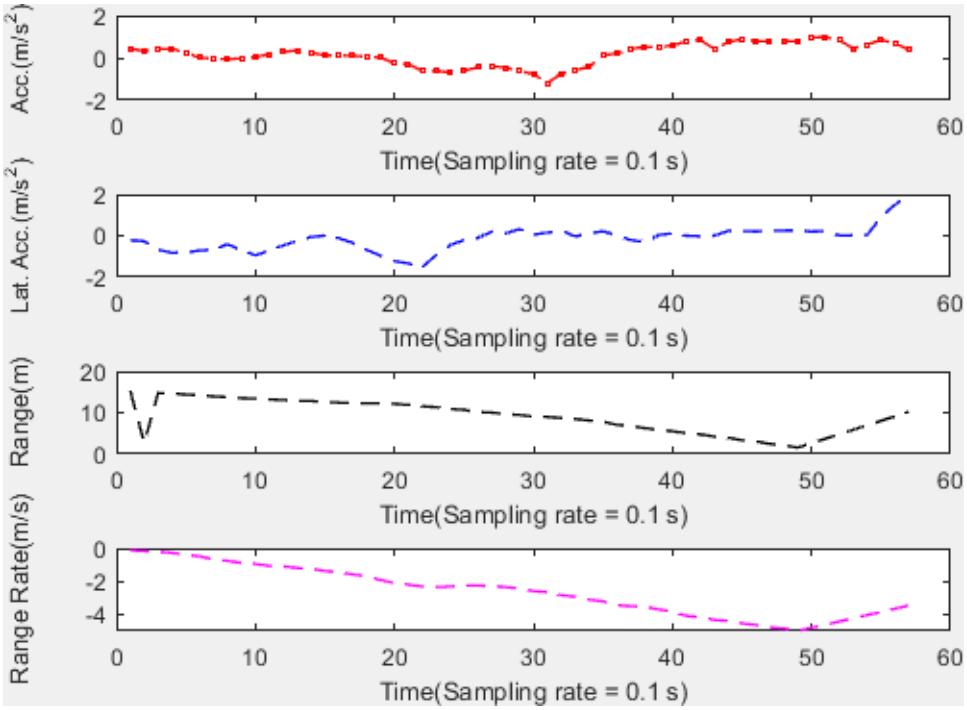


Figure 6.6: A set of observations showing a non-skilled *sd* during a conflict with a following vehicle

Non-faulty class

In this conflict type, despite of the smooth behavior of the *sd*, the inattentiveness or aggressiveness of the following vehicle causes a risky event. In figure 6.7, an *sd* is involved in a near-crash event due to the in-attentiveness of the following vehicle. As can be noticed, although the *sd* is keeping a constant speed, his rearward range started to decrease gradually at time $\approx$ 2.5s. A near-crash event occurred at time $\approx$ 5s when the *sd* smoothly decelerated at time $\approx$ 4.5s.

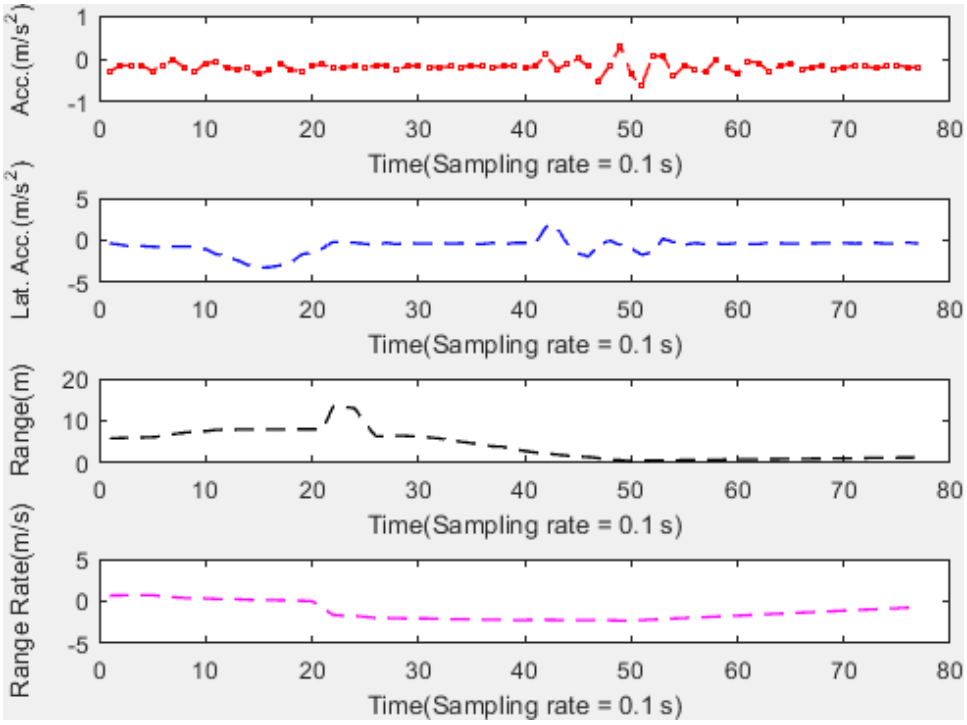


Figure 6.7: A set of observations showing a non-faulty *sd* during a conflict with a following vehicle

6.5 Fault Inference Profiling System

In this section, the details of the fault inference profiling system are provided. The details comprise the notational conventions that are utilized, the data filtering, preprocessing, and feature selection, and the HMM-based formulation of the driver behavior classification problem.

6.5.1 Notational Conventions

The mathematical notations that are used in this section are displayed in table 6.3.

Table 6.3: Summary of Notations

Notation	Description
sv	Subject vehicle
N	Number of HMM model states
M	Number of possible observations per state
A	The HMM transition matrix
B	The HMM emission matrix
π	The HMM initial state distribution array
TTC	Time-to-Collision
$R_{Forward}$	Range between sv and the nearest vehicle/object/pedestrian in the forward direction
$RR_{Forward}$	Range rate between sv and the nearest vehicle/object/pedestrian in the forward direction
$R_{Rearward}$	Range between sv and the nearest vehicle/object/pedestrian in the rearward direction
$RR_{Rearward}$	Range rate between sv and the nearest vehicle/object/pedestrian in the rearward direction
Acc_{Long}	Acceleration in the longitudinal direction
Acc_{Lat}	Acceleration in the lateral direction
Z_k	The k_{th} observation matrix
$Z_{k,t}$	The observation vector that belongs to Z_k at time t
S_j	An HMM state corresponding to the driving behavioral mode j
$P(S_j\|S_i)$	An HMM state transition probability
$P(Z_{k,t}\|S_j)$	An HMM emission probability
$P(Z_k\|\lambda_i)$	The posterior probability of the observation sequence Z_k given the HMM model λ_i

6.5.2 Data Filtering and Pre-processing

In this study, the 100-CAR NDS data is used for the training and validation of the HMM models. The 100-CAR NDS project is a large-scale data collection project sponsored by the National Highway Traffic Safety Administration (NHTSA) and the

Virginia Department of Transportation (VDoT) [93]. In the 100-CAR NDS, 241 primary and secondary drivers were recruited over a period of 1 year to collect large-scale driving data. Recruited drivers used approximately 100 cars instrumented with a set of sensors including forward and rearward radar sensors, OBD units, accelerometers, gyroscopes, five channels of digital video, and GPS. The sampling rate of the acquired data ranged from 1 Hz to 10 Hz. Data was recorded using Electronic Digital Recorders (EDR) resulting in a significant amount of driving data which is approximately 43,000 hours of data [93].

Data reductionists identified a total of 69 crash and 760 near-crash events based on different types of triggering signals (e.g., acceleration or deceleration ≥ 0.5 g coupled with a time-to-collision (TTC) of 4 seconds or less). The detailed identification process could be found in [93]. Only 176 events that match the scope of this study have been used for model training and validation purposes. For each of these events, a file that contains a time-series data spanning 30 seconds before and 10 seconds after the event is available. Time-series data include 31 variables (e.g., speed, acceleration, etc.) that describe the sv behavior during each event. The detailed narrative of each event was extracted and documented using the installed digital video cameras. These narratives are used in this work for labelling the events.

During the data collection process, sensors failed to capture the values of some variables at some time instants, causing some gaps in the data. In this work, missing data are approximated using linear interpolation. Only a few variables are initially selected as candidate features for model training. They can be categorized into two types:

1. Type 1 variables: variables that reflect the longitudinal and lateral movements

of the *sv*. These variables are gas pedal position, speed (CAN-bus), speed (GPS), yaw rate, vehicle heading (GPS), longitudinal and lateral accelerations.

2. Type 2 variables: variables that reflect the relative motion of the *sv* to leading and following vehicles. These variables are: the forward range and range rate, and the rearward range and range rate.

The forward range is the distance between the *sv* and the nearest leading vehicle/object/pedestrian, while the rearward range is the distance between the *sv* and the nearest following vehicle/object/pedestrian, at any point in time. They are expressed mathematically as:

$$R_{Forward}(t) = |x_i(t) - x_{i+1}(t)| \tag{6.1}$$

$$R_{Rearward}(t) = |x_i(t) - x_{i-1}(t)| \tag{6.2}$$

where $R_{Forward}(t)$ and $R_{Rearward}(t)$ are the forward and rearward ranges at time t, respectively, $x_i(t)$, $x_{i+1}(t)$, and $x_{i-1}(t)$ are the positions of the *sv*, leading vehicle, and following vehicle at time t, respectively. The range rate is the rate of change of the distance between the *sv* and following or leading vehicles. It possesses a negative value when the distance decreases over time. Forward and rearward range rates can be expressed mathematically as:

$$RR_{Forward}(t) = \frac{x_{i+1}(t+\Delta) - x_{i+1}(t)}{\Delta} - \frac{x_i(t+\Delta) - x_i(t)}{\Delta} \tag{6.3}$$

$$RR_{Rearward}(t) = \frac{x_i(t+\Delta) - x_i(t)}{\Delta} - \frac{x_{i-1}(t+\Delta) - x_{i-1}(t)}{\Delta} \tag{6.4}$$

where $RR_{Forward}(t)$ and $RR_{Rearward}(t)$ are the forward and rearward range rates, respectively, and Δ represents the time step.

Feature Selection

The statistical dependencies among "type 1" variables are obtained through the calculation of their correlation matrix. A fixed correlation coefficient threshold of a value ≥ 0.8 is adopted to indicate redundancy between two variables. Feature space is reduced by removing all redundant variables, resulting in only three "type 1" candidate variables which are speed (CAN-bus), longitudinal acceleration, and lateral acceleration. Finally, the speed variable is removed for more simplification since it does not improve the classification accuracy based on the experimental results.

6.5.3 HMM-based Formulation

First-order time homogeneous discrete HMMs are utilized to classify drivers under the two different conflict types. Hidden Markov modeling is one of the best modeling approaches for modeling time-series systems [95]. Unlike conventional Markov models, where each state corresponds to a fully observed physical event, HMM observations are stochastically related to the hidden states. An HMM is formally defined by the five tuple $\Omega(N, M, A, B, \pi)$, where N is the number of states, M is the number of possible observations per state, A is the transition matrix, B is the emission matrix, and π is the initial state distribution array. Transitions between states are annotated with probabilities that compose the transition matrix (i.e., $P(S_j|S_i)$), whereas the emission matrix includes the emission probabilities of observation symbols given the presence at a specific state (i.e., $P(Z_{k,t}|S_j)$). The initial state distribution matrix contains the initial probabilities of each state (i.e., $P(S_j)$). For convenience, HMMs are usually described using the following simplified notation: $\lambda = (A, B, \pi)$. Figure 6.8 depicts the architecture of the utilized HMM-based models.

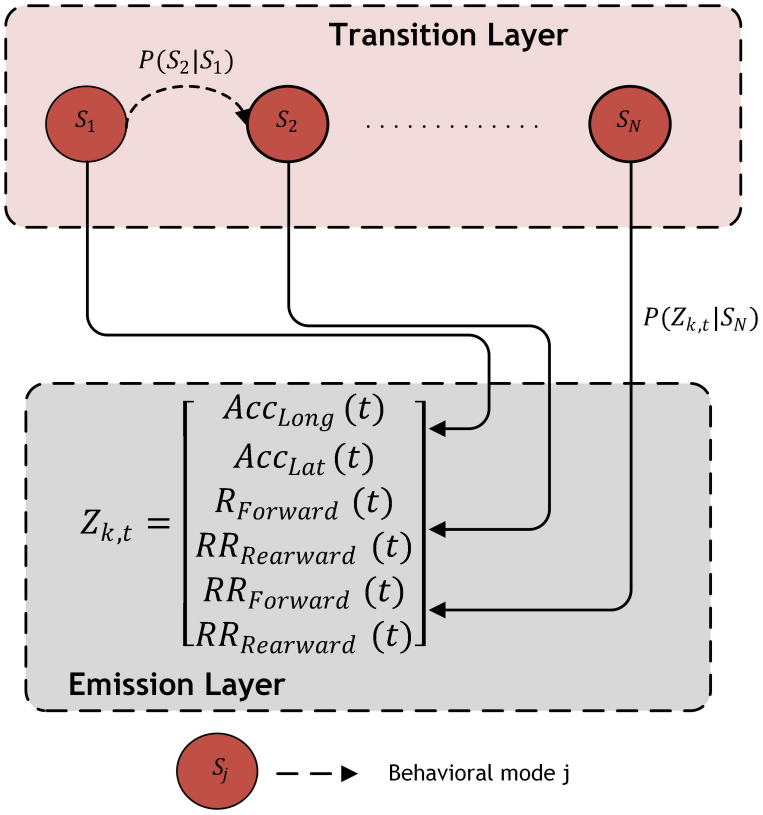


Figure 6.8: HMM-based architecture for fault inference during risky events.

In our problem, the number of hidden states is a tunable parameter that represents the possible driving behavioral modes, and the observations are the six time-series signals previously discussed in section 6.5.2. Through an iterative process, each of the six observation sequences is quantized into six possible quantization levels which take values from the set $Q = \{1, 2, 3, 4, 5, 6\}$ according to their level and then normalized where their normalized values $\in [0, 1]$. For each risky event, the *sv* driver is characterized by these sequences, which are mapped into a 1-dimensional emission array. Some of the emission arrays are used for models training. Consequently, the Baum Welch algorithm, aka forward-backward algorithm, is used to solve the HMM learning problem. Baum-Welch algorithm is an iterative update procedure in which

A, B, and π are re-estimated in each iteration to maximize the likelihood of a given observation sequence (i.e., $\max_{A,b,\pi} P(Z_k)$). This is done by utilizing the following two update equations:

$$\xi_t(i,j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}, \tag{6.5}$$

and

$$\gamma_t(i) = \sum_{j=1}^{N}\xi_t(i,j), \tag{6.6}$$

where $\alpha_t(i)$ and $\beta_{t+1}(j)$ are the forward and backward HMM variables for states i and j, at time instants t and $t+1$, respectively, a_{ij} is the transition probability from state i to state j, and $b_j(Z_{k,t+1})$ is the emission probability of the observed symbol at state j and time $t+1$. Readers are referred to [95] for a detailed explanation of this algorithm.

Five unique HMMs, representing the five behavioral classes discussed in the previous section, are trained using Baum Welch algorithm. L_i observation sequences are used to train each model, where i refers to the model index. Baum Welch algorithm does not guarantee the convergence to a global maximum. As a result, many initializations of the transition and emission matrices are tested to improve the converged values over different trials. A tolerance value of $1e^{-4}$, which represents the difference in the forward log probabilities between two successive iterations, is utilized as a convergence criterion. Model evaluation is performed using Z validation sequences. The evaluation process utilizes the well-known Forward algorithm. For type 1 conflicts, two probabilities are computed for each sequence: $\mathrm{P}(Z_k|\lambda_{Faulty})$, and $\mathrm{P}(Z_k|\lambda_{Skilled})$. Similarly, three forward probabilities are computed for each sequence resulted from a

conflict of type 2: P($Z_k|\lambda_{Faulty}$), P($Z_k|\lambda_{Non-skilled}$), and P($Z_k|\lambda_{Non-faulty}$). In each of these types, the *sv* driver is assigned to the class with the highest forward probability. The number of states (N), sampling time (T_s), and the time window (T_w) in which the observations are analysed are all HMM tunable hyper-parameters. The number of states represents the number of behavioral modes the model can capture. The choice of the number of states is data-dependent as it depends on the size of the training set. Increasing the states is a small or intermediate size data-sets could result in a model's over-fitting. The sampling time T_s depends on the dynamic nature of the problem. Since driving is a highly dynamic process, choosing sampling time on the scale of sub-seconds is important. Finally the window size T_w defines how much time before and after the risky incident is needed to infer the skillfulness level of the *sv* driver during the event. The utilized best combination of such these hyper-parameters is shown in table 6.4.

Table 6.4: HMM Hyper-parameters

Hyper-parameter	**Value**
N	6
T_s	10 ms
T_w	4s

where the start of T_w is 2s before the risky incident.

A risk score of a subject driver in this system will be characterized by the number of risky events the driver was involved in. For instance, a risk score of the subject driver *sd* in a driving trip, *trip*, can be calculated using:

$$Risk_{trip,sd} = [N_{Faulty}, N_{Non-faulty}, N_{Skilled}, N_{Non-skilled}] \tag{6.7}$$

where N_{Faulty}, $N_{Non-faulty}$, $N_{Skilled}$, $N_{Non-skilled}$ are, respectively, the number of risky

events the subject driver was classified as faulty, non-faulty, skilled, and non-skilled.

Following the development of this vector, the developed risk score of the subject driver in sections 3 or 4 can be adjusted in many different ways. For instance, a detection of only one risky event in which the subject driver was classified as *Faulty* may be used to override his/her risk score to a zero value.

6.6 Results and Discussion

HMM algorithm is implemented using MATLAB R2015b statistics toolbox. A total of 248 risky sequences are used for the training and validation of the HMM models. A total of 214 of these sequences correspond to risky events of type 1 (i.e., conflicts with leading vehicles). Half of these sequences are used to train the two aforementioned models. On the other hand, only 34 events are available for the case of conflicts with a following vehicle. Similar to the first conflict type, 50% of the data is utilized for models training. As mentioned in section 6.5.3, the original sequences are cropped to include $2s$ before the occurrence of the risky event. This has resulted in the best experimental classification results. As previously mentioned, the data is sampled at a rate of 10 Hz. The overall accuracy of driver classification under each conflict type

Table 6.5: Confusion matrix for classification under type 1 conflicts

		Predicted	
		Faulty	Skilled
Actual	Faulty	77	18
	Skilled	3	9

is calculated using the following formula:

$$Accuracy = \frac{T_P}{T_P + F_P} \tag{6.8}$$

where T_P and F_P are the numbers of true positive and false positive predictions for the considered class, respectively. The overall accuracy of type 1 and type 2 conflicts is 80.37% and 72.2%, respectively. Tables 6.5 and 6.6 depict the confusion matrices of the different classes under the two conflict types. An important observation is that for conflicts of type 2 developed HMMs were unable to make a clear distinction between the Non-skilled an Non-faulty classes. That is attributed to the slight difference between the two behavioral classes and the subjectivity the labelling process entailed. It is also expected that using a larger data set would lead to better performance results.

Table 6.6: Confusion matrix for classification under type 2 conflicts

		Predicted		
Actual		Faulty	Non-skilled	Non-faulty
	Faulty	6	0	1
	Unskilled	0	7	2
	Non faulty	0	2	0

6.7 Summary

This chapter introduced a novel HMM-based classification approach for determining the fault responsibility of drivers involved in risky driving conflicts. The chapter focused on two types of driving conflicts which are conflicts with leading and following vehicles, under normal weather and road conditions. A total of 124 training

sequences were used to train five unique HMMs that represent five distinguishable behaviors. The models were successfully validated using 124 evaluation sequences. Model training and evaluation were performed using Baum welch and Forward algorithms, respectively. Overall classification accuracy of 80.37 % is achieved for conflicts of type 1, whereas an accuracy of 72.2 % is achieved for conflicts of type 2. These promising results can be improved by using a larger data set with more training and validation sequences.

Chapter 7

Conclusions and Future Work

7.1 Summary

In this thesis, two problems in the field of driver risk profiling, not covered in current literature, were tackled. The first is the *subjectivity* of the current risk scoring functions since scores are assigned in many cases based on subjective opinions on risk weights of behaviors, while the second problem is the *generality* of the scoring functions as they do not take into consideration the variation in the driving traits of individual drivers.

To tackle the first problem (i.e., *subjectivity*), we developed a robust and reliable risk scoring model based on SHRP2 Naturalistic Driving Study (NDS) data-set. Through the extraction of the behavioral patterns of over 3,000 drivers and their long-term risk rates quantified in terms of crash and near-crash rate, two customized Random Forest (RF) predictors, in the classification and regression contexts, were trained and validated to predict the level of risk taken by a driver. The models were chosen through an extensive selection process that comprises initial selection phase, data filtering and pre-processing, feature engineering, and hyper-parameters

optimization. Results clearly reflect the robustness of the developed models in several performance metrics. The developed RF regressor was bench-marked against another RF model that is trained using only the conventional predictors used in literature. The results show that the developed RF model clearly outperforms the other model in explaining the data variability and in the performance of the mean square and mean absolute errors.

A novel driver risk profiling framework was then presented. The information flow through three distinct computational layers (in-vehicle, edge/fog, and cloud layers) was justified. In the envisioned framework, the risk prediction problem is addressed through the use of driving behaviors and their environmental context as risk predictors. Thirteen driving behaviors and sixteen environmental features were utilized as inputs to the risk prediction models. Bias-variance trade-off analysis was performed to achieve the best prediction performance. Obtained results confirm that contextual information is an important factor in the prediction performance of risk rate. The developed risk prediction model was then utilized in a complete environment-aware driver profiling system that includes risk scoring and profiling components. The results indicate that the proposed system is robust and accurate in assigning representative risk scores for drivers. To highlight the functionality of the proposed system, an explanatory example was provided and investigated.

Motivated by the obtained results, a novel safety-based route planner was proposed. In the proposed system, the personalized risk profiles of drivers in different driving environments are incorporated in a safety-based routing system which suggests driving routes that minimize their personal risk. In this framework, the road

network is abstracted as a directed graph in which the road intersections are represented by graph nodes whereas graph edges represent road segments. Each road segment in the graph is characterized by a 3-tuple that contains the expected travel time of the segment, the expected risk of the segment from a holistic perspective, and the driver's personal risk score in the segment, given the static and dynamic environmental attributes of that segment. By taking the weighted sum from the 3-tuple segments a cost function was defined. Then, the safety-based routing problem was formulated as a Linear Integer Programming (LIP) problem with a constraint on the overall risk of potential routes. A real-world case study from the Ontario, Canada was investigated and solved using Gurobi optimizer.

To tackle the second problem (i.e., *generality*), we proposed a fault inference profiling system that is based on the actual involvement of drivers in risky events (i.e., crash or near-crash events) and what the drivers' fault contribution was in these events. In the proposed system, the behavior of drivers during risky events is classified under four distinct categories: *Faulty, Non-Faulty, Skilled*, and *Non-Skilled*. Two specific driving conflicts, involving conflicts with following and leading vehicles were investigated. Five Hidden Markov Models (HMM) where trained using vehicular and radar forward and rearward range data during these conflicts to infer the behavioral class of the subject driver. The HMM models were trained using the Baum Welch algorithm and the evaluation was performed using Forward algorithm on 248 events. Results show that these models can accurately infer the fault contribution of drivers.

7.2 Future Work

Future work in this area includes the following enhancements and suggestions:

Limitations: Exposure Information and False Labeling

In this thesis, the baseline driving events were utilized to provide exposure information about the effect of driving behaviors on risk rate. Despite the robustness of the results obtained from the developed risk prediction models using such events, more rigorous conclusions are expected using a larger set of exposure information. This can be achieved using the *Trip-based* data-set in the SHRP2 study. The *Trip-based* data-set contains the time-series driving data for more than 5,500,000 driving trips. Moreover, it contains a summary of driving behaviors for subject drivers in each trip. This huge amount of data can be leveraged towards drawing more rigorous conclusions on the effect of each driving behavior and the effect of different behavioral patterns of drivers on risk rate.

False labeling could impact the robustness of the results based on the false labeling severity. False labels could be identified manually during the pre-processing phase or automatically through incorporating fairness in the utilized ML algorithm (e.g., using Lagrange multipliers).

Modelling Environment Uncertainty

The proposed safety-based route planner does not consider the environmental changes during a long driving trip. Given the dynamic environmental factors such as the traffic density and weather conditions, modelling the environmental changes in road segments is of great value since the risk index of a given road segment as well as the personal risk index of a subject driver in that segment may differ as environment conditions change. Stochastic optimization techniques can be leveraged towards providing more realistic route suggestions at the beginning of a driving trip based on

expected states of road segments at the expected time the subject driver reaches them.

Computer Vision Based Fault Inference

In the proposed fault inference system, the vehicular and radar range data are utilized to train the HMM sequence models. An alternative method can be through applying sequence modelling techniques to the video frames of risky events. Also, in addition to HMM-based modelling approach, variants of recursive neural networks such as Long-Short-Term-Memory (LSTM) modelling approach can be utilized and compared to HMM-based approach.

Interaction Between Driving Styles

In the proposed safety-based routing system, route suggestions are based only on the risk profile of the subject driver in different driving environments. However, this system does not take into consideration the profiles of other drivers who are present in road segments during the driving trip of the subject driver. Modelling the interaction effect between different driving styles on risk can be used to provide better routing options that could more efficiently mitigate risk.

Hybrid Profiling System

A scoring model that combines the system presented either in chapter 3 or 4 and the fault inference system presented in chapter 6 needs to be formulated. The hybrid system should assign risk scores for drivers based on a weighted sum of their expected driving risk and their actual involvement in risky events.

Fault Inference Using a Larger Data-Set

The customized HMM models presented in chapter 6 were trained and validated using the 100-CAR NDS. More robust performance results are expected using the much larger SHRP2 time-series data-set, which contains over 9,000 risky events of different conflict types.

7.3 Concluding Remarks

In this work, we proposed a novel approach for profiling drivers through the use of large scale NDSs. In addition to the recommended future research directions, we summarize the recommendations and lessons learned throughout this research work.

- Depending only on the conventional behavioral FoMs reported in the literature is not sufficient for building statistically significant prediction models. Prediction models trained only on the conventional FoMs were not able to explain the data variability. That is reflected in the low R^2 score of these models.

- The developed prediction models seemed to benefit from the joint use of driving behaviors and their contextual information as predictors. However, there is a practical consideration on the applicability of retrieving the contextual information used in this work in real-time. There should be a trade-off between practicability and performance of developed risk prediction models. An important consideration is to only use the most relevant contextual information that greatly affects the risk prediction process.

- Using only the long-term behavioral patterns of drivers was sufficient to build robust prediction models that precisely predict the driving risk of drivers.

- In addition to the conventional applications of driver risk profiling such as the PHYD application, driver risk profiling can be utilized in a variety of other applications including the personalized safety-based route planning and the real-time warnings/recommendations for drivers.

- In the classification context, Random Forest classifiers outperformed other classifiers - including SVM - in different performance measures for inter-mediate sized, skewed data with un-balanced classes. Moreover, The training of SVM seemed a computationally inefficient process as it required finding a kernel function that matches the distribution of the feature space.

Bibliography

[1] "Road Safety Strategy 2025 - Home." [Online]. Available: https://roadsafetystrategy.ca/en/

[2] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver Behavior Analysis for Safe Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3017–3032, Dec. 2015.

[3] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 91–102, 2015.

[4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[5] J. A. Stankovic, "Research Directions for the Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[6] A. S. El-Wakeel, J. Li, A. Noureldin, H. S. Hassanein, and N. Zorba, "Towards a Practical Crowdsensing System for Road Surface Conditions Monitoring," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[7] M. Hung, "Gartner Insights on How to Lead in a Connected World," p. 29. [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

[8] B. Hussain, Q. U. Hasan, N. Javaid, M. Guizani, A. Almogren, and A. Alamri, "An Innovative Heuristic Algorithm for IoT-Enabled Smart Homes for Developing Countries," *IEEE Access*, vol. 6, pp. 15 550–15 575, 2018.

[9] P. Kodeswaran, R. Kokku, M. Mallick, and S. Sen, "Demultiplexing activities of daily living in IoT enabled smarthomes," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, Apr. 2016, pp. 1–9.

[10] "Wearable Devices and the Internet of Things | Mouser." [Online]. Available: https://ca.mouser.com/applications/article-iot-wearable-devices/

[11] L. F. Herrera-Quintero, J. C. Vega-Alfonso, K. B. A. Banse, and E. C. Zambrano, "Smart ITS Sensor for the Transportation Planning Based on IoT Approaches Using Serverless and Microservices Architecture," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 17–27, 2018.

[12] "WHO | Global status report on road safety 2015." [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/

[13] T. S. J. Darwish and K. A. Bakar, "Fog Based Intelligent Transportation Big Data Analytics in The Internet of Vehicles Environment: Motivations, Architecture, Challenges, and Critical Issues," *IEEE Access*, vol. 6, pp. 15 679–15 701, 2018.

[14] G. Castignani, R. Frank, and T. Engel, "Driver behavior profiling using smartphones," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct. 2013, pp. 552–557.

[15] P. Handel, I. Skog, J. Wahlstrom, F. Bonawiede, R. Welch, J. Ohlsson, and M. Ohlsson, "Insurance Telematics: Opportunities and Challenges with the Smartphone Solution," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 57–70, 2014.

[16] V. Neale, S. Klauer, T. Dingus, J. Sudweeks, and M. Goodman, "An Overview of the 100-Car Naturalistic Study and Findings," p. 10.

[17] K. L. CAMPBELL, "SAFETY The SHRP 2 Naturalistic Driving Study," p. 8, 2012.

[18] R. Eenink, Y. Barnard, M. Baumann, X. Augros, and F. Utesch, "UDRIVE: the European naturalistic driving study," p. 10.

[19] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proceedings of the National Academy of Sciences*, vol. 113, no. 10, pp. 2636–2641, Mar. 2016. [Online]. Available: http://www.pnas.org/content/113/10/2636

[20] "SHRP2 NDS Data Access." [Online]. Available: https://insight.shrp2nds.us/home

[21] T. Chakravarty, A. Ghose, C. Bhaumik, and A. Chowdhury, "MobiDriveScore #x2014; A system for mobile sensor based driving analysis: A risk assessment

model for improving one's driving," in *2013 Seventh International Conference on Sensing Technology (ICST)*, Dec. 2013, pp. 338–344.

[22] "Take the Aviva Drive challenge - Aviva." [Online]. Available: https://www.aviva.co.uk/car-insurance/drive/

[23] "Mobile Web - State Farm®." [Online]. Available: https://www.statefarm.ca/about-us/innovation-research/mobile

[24] "TD MyAdvantage - Safe Driving Discount | TD Insurance." [Online]. Available: https://www.tdinsurance.com/products-services/auto-car-insurance/my-advantage

[25] A. Kashevnik, I. Lashkov, and A. Gurtov, "Methodology and Mobile Application for Driver Behavior Analysis and Accident Prevention," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2019.

[26] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver Behavior Classification at Intersections and Validation on Large Naturalistic Data Set," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 724–736, Jun. 2012.

[27] S. Choi, J. Kim, D. Kwak, P. Angkititrakul, and J. H. L. Hansen, *Analysis and Classification of Driver Behavior using In-Vehicle CAN-Bus Information.*

[28] S. Daptardar, V. Lakshminarayanan, S. Reddy, S. Nair, S. Sahoo, and P. Sinha, "Hidden Markov Model based driving event detection and driver profiling from mobile inertial sensor data," in *2015 IEEE SENSORS*, Nov. 2015, pp. 1–4.

[29] C. G. Q. M, J. O. López, and A. C. C. Pinilla, "Driver behavior classification model based on an intelligent driving diagnosis system," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sep. 2012, pp. 894–899.

[30] M. R. Carlos, L. C. González, J. Wahlström, G. Ramírez, F. Martínez, and G. Runger, "How Smartphone Accelerometers Reveal Aggressive Driving Behavior?–The Key Is the Representation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[31] D. Alvarez-Coello, B. Klotz, D. Wilms, S. Fejji, J. M. Gómez, and R. Troncy, "Modeling dangerous driving events based on in-vehicle data using Random Forest and Recurrent Neural Network," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 165–170.

[32] N. Arbabzadeh and M. Jafari, "A Data-Driven Approach for Driving Safety Risk Prediction Using Driver Behavior and Roadway Information Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 446–460, Feb. 2018.

[33] M. Winlaw, S. H. Steiner, R. J. MacKay, and A. R. Hilal, "Using telematics data to find risky driver behaviour," *Accident Analysis & Prevention*, vol. 131, pp. 131–136, Oct. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0001457519304956

[34] E. Carvalho, B. V. Ferreira, J. Ferreira, C. d. Souza, H. V. Carvalho, Y. Suhara, A. S. Pentland, and G. Pessin, "Exploiting the use of recurrent neural networks

for driver behavior profiling," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 3016–3021.

[35] J. F. Júnior, E. Carvalho, B. V. Ferreira, C. d. Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning," *PLOS ONE*, vol. 12, no. 4, p. e0174959, Apr. 2017. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174959

[36] L. M. Bergasa, D. Almería, J. Almazán, J. J. Yebes, and R. Arroyo, "DriveSafe: An app for alerting inattentive drivers and scoring driving behaviors," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, Jun. 2014, pp. 240–245.

[37] E. Romera, L. M. Bergasa, and R. Arroyo, "Need data for driver behaviour analysis? Presenting the public UAH-DriveSet," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 387–392.

[38] C.-W. You, M. Montes-de Oca, T. J. Bao, N. D. Lane, H. Lu, G. Cardone, L. Torresani, and A. T. Campbell, "CarSafe: a driver safety app that detects dangerous driving behavior using dual-cameras on smartphones." ACM Press, 2012, p. 671. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2370216.2370360

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

[40] K. Tang, S. Zhu, Y. Xu, and F. Wang, "Modeling Drivers' Dynamic Decision-Making Behavior During the Phase Transition Period: An Analytical Approach Based on Hidden Markov Model Theory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 206–214, Jan. 2016.

[41] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Behavior classification algorithms at intersections and validation using naturalistic data," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2011, pp. 601–606.

[42] A. Sathyanarayana, P. Boyraz, and J. H. L. Hansen, "Driver behavior analysis and route recognition by Hidden Markov Models," in *2008 IEEE International Conference on Vehicular Electronics and Safety*, Sep. 2008, pp. 276–281.

[43] N. Oliver and A. P. Pentland, *Driver Behavior Recognition and Prediction in a SmartCar*, 2000.

[44] P. Boyraz, M. Acar, and D. Kerr, "Signal Modelling and Hidden Markov Models for Driving Manoeuvre Recognition and Driver Fault Diagnosis in an urban road scenario," in *2007 IEEE Intelligent Vehicles Symposium*, Jun. 2007, pp. 987–992.

[45] Y. Wang, X. Zhang, M. Li, P. Jiang, and F. Wang, "A GM-HMM based abnormal pedestrian behavior detection method," in *2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Sep. 2015, pp. 1–6.

[46] H. Aliakbarpour, K. Khoshhal, J. Quintas, K. Mekhnacha, J. Ros, M. Andersson, and J. Dias, "HMM-Based Abnormal Behaviour Detection Using Heterogeneous

Sensor Network," in *Technological Innovation for Sustainability*, ser. IFIP Advances in Information and Communication Technology, L. M. Camarinha-Matos, Ed. Springer Berlin Heidelberg, 2011, pp. 277–285.

[47] C. Chuang, C. Yang, and Y. Lin, "HMM-based driving behavior recognition for in-car control service," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, Jun. 2015, pp. 258–259.

[48] P.-C. Chung and C.-D. Liu, "A daily behavior enabled hidden Markov model for human behavior understanding," *Pattern Recognition*, vol. 41, no. 5, pp. 1572–1580, May 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320307004700

[49] T. G. Brown, M. C. Ouimet, M. Eldeb, J. Tremblay, E. Vingilis, L. Nadeau, J. Pruessner, and A. Bechara, "The effect of age on the personality and cognitive characteristics of three distinct risky driving offender groups," *Personality and Individual Differences*, vol. 113, pp. 48–56, Jul. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S019188691730168X

[50] J. M. Hankey, M. A. Perez, and J. A. McClafferty, "Description of the SHRP 2 Naturalistic Database and the Crash, Near-Crash, and Baseline Data Sets," Apr. 2016. [Online]. Available: https://vtechworks.lib.vt.edu/handle/10919/70850

[51] I. Jolliffe, "Principal Component Analysis," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1094–1096. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_455

[52] A. B. Graf and S. Borer, "Normalization in Support Vector Machines," in *Pattern Recognition*, G. Goos, J. Hartmanis, J. van Leeuwen, B. Radig, and S. Florczyk, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, vol. 2191, pp. 277–282. [Online]. Available: http://link.springer.com/10.1007/3-540-45404-7_37

[53] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[54] P. Hall, B. U. Park, and R. J. Samworth, "Choice of neighbor order in nearest-neighbor classification," *The Annals of Statistics*, vol. 36, no. 5, pp. 2135–2152, Oct. 2008, arXiv: 0810.5276. [Online]. Available: http://arxiv.org/abs/0810.5276

[55] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," p. 22.

[56] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012. [Online]. Available: http://www.jmlr.org/papers/v13/bergstra12a.html

[57] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004. [Online]. Available: https://link.springer.com/article/10.1023/B:STCO.0000035301.49549.88

[58] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Mar. 1986. [Online]. Available: http://link.springer.com/10.1007/BF00116251

[59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online]. Available: https://www.nature.com/articles/nature14539

[60] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Nov. 2016, google-Books-ID: omivDQAAQBAJ.

[61] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608014002135

[62] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, Dec. 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231206000385

[63] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for Regression and Multiclass Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[64] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, and M. S. Obaidat, "Edge Computing-Based Security Framework for Big Data Analytics in VANETs," *IEEE Network*, vol. 33, no. 2, pp. 72–81, Mar. 2019.

[65] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a resource (VaaR)," *IEEE Network*, vol. 29, no. 1, pp. 12–17, Jan. 2015.

[66] D. I. Tselentis, G. Yannis, and E. I. Vlahogianni, "Innovative Insurance Schemes: Pay as/how You Drive," *Transportation Research Procedia*, vol. 14, pp. 362–371, Jan. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352146516300898

[67] M. J. Davidson and J. A. O. III, "Systems and methods for utilizing telematics data to improve fleet management operations," US Patent US8 416 067B2, Apr., 2013. [Online]. Available: https://patents.google.com/patent/US8416067B2/en

[68] K. Takeda, C. Miyajima, T. Suzuki, P. Angkititrakul, K. Kurumida, Y. Kuroyanagi, H. Ishikawa, R. Terashima, T. Wakita, M. Oikawa, and Y. Komada, "Self-Coaching System Based on Recorded Driving Data: Learning From One's Experiences," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1821–1831, Dec. 2012.

[69] M. Witt, K. Kompaß, L. Wang, R. Kates, M. Mai, and G. Prokop, "Driver profiling – Data-based identification of driver behavior dimensions and affecting driver characteristics for multi-agent traffic simulation," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 64, pp. 361–376, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1369847818308131

[70] A. A. Rahman, W. Saleem, and V. V. Iyer, "Driving Behavior Profiling and Prediction in KSA using Smart Phone Sensors and MLAs," in *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information*

Technology (JEEIT), Apr. 2019, pp. 34–39.

[71] B. He, D. Zhang, S. Liu, H. Liu, D. Han, and L. M. Ni, "Profiling Driver Behavior for Personalized Insurance Pricing and Maximal Profit," in *2018 IEEE International Conference on Big Data (Big Data)*, Dec. 2018, pp. 1387–1396.

[72] A. Abdelrahman, N. Abu-Ali, and H. S. Hassanein, "On the Effect of Traffic and Road Conditions on the Drivers' Behavior: A Statistical Analysis," in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, Jun. 2018, pp. 892–897.

[73] H. A. H. Naji, N. Lyu, C. Wu, and H. Zhang, "Examining contributing factors on driving risk of naturalistic driving using K-means clustering and ordered logit regression," in *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, Aug. 2017, pp. 1189–1195.

[74] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[75] "Road traffic injuries." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

[76] A. Taha and N. AbuAli, "Route Planning Considerations for Autonomous Vehicles," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 78–84, Oct. 2018.

[77] S. Abdelhamid, H. S. Hassanein, and G. Takahara, "Vehicle as a Mobile Sensor," *Procedia Computer Science*, vol. 34, pp. 286–295, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914008801

[78] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 173–184, event-place: Istanbul, Turkey. [Online]. Available: http://doi.acm.org/10.1145/2348543.2348567

[79] H. Zhang, Q. Zhang, and X. Du, "Toward Vehicle-Assisted Cloud Computing for Smartphones," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5610–5618, Dec. 2015.

[80] M. Gerla, "Vehicular Cloud Computing," in *2012 The 11th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun. 2012, pp. 152–155.

[81] A. Boukerche and R. E. De Grande, "Vehicular cloud computing: Architectures, applications, and mobility," *Computer Networks*, vol. 135, pp. 171–189, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128618300057

[82] A. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "A Cloud-Based Environmental-Aware Driver Profiling Framework Using Ensemble Learning," in *ICC 2019 - 2019 IEEE International Conference on Communications*, May 2019, pp. 1–6.

[83] A. Haghani and S. Jung, "A dynamic vehicle routing problem with time-dependent travel times," *Computers & Operations Research*, vol. 32, no. 11, pp. 2959–2986, Nov. 2005.

[84] "WAZE Outsmarting Traffic Together," [online] Available: http://www.waze.com/, accessed: April, 2019.

[85] G. Difilippo, M. P. Fanti, G. Fiume, A. M. Mangini, and N. Monsel, "A Cloud Optimizer for Eco Route Planning of Heavy Duty Vehicles," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec. 2018, pp. 7142–7147.

[86] Z. Li, I. V. Kolmanovsky, E. M. Atkins, J. Lu, D. P. Filev, and Y. Bai, "Road Disturbance Estimation and Cloud-Aided Comfort-Based Route Planning," *IEEE Transactions on Cybernetics*, vol. 47, no. 11, pp. 3879–3891, Nov. 2017.

[87] A. S. El-Wakeel, A. Noureldin, H. S. Hassanein, and N. Zorba, "iDriveSense: Dynamic Route Planning Involving Roads Quality Information," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6.

[88] A. M. Taha, "Facilitating safe vehicle routing in smart cities," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–5.

[89] Z. Li, I. Kolmanovsky, E. Atkins, J. Lu, D. P. Filev, and J. Michelini, "Road Risk Modeling and Cloud-Aided Safety-Based Route Planning," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2473–2483, Nov. 2016.

[90] Q. Liu, S. Kumar, and V. Mago, "SafeRNet: Safe transportation routing in the era of Internet of vehicles and mobile crowd sensing," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2017, pp. 299–304.

[91] "Road Network File, Reference Guide, 2018." in *Statistics Canada Catalogue no. 92-500-G.*

[92] "Gurobi - The fastest solver." [Online]. Available: https://www.gurobi.com/

[93] T. A. Dingus, S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. D. Sudweeks, M. A. Perez, J. Hankey, D. J. Ramsey, S. Gupta, C. Bucher, Z. R. Doerzaph, J. Jermeland, and R. R. Knipling, "The 100-Car Naturalistic Driving Study, Phase II - Results of the 100-Car Field Experiment," Apr. 2006. [Online]. Available: https://trid.trb.org/view/783477

[94] "R.R.O. 1990, Reg. 668: FAULT DETERMINATION RULES." [Online]. Available: https://www.ontario.ca/laws/view

[95] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.