UTS UNIVERSITY OF TECHNOLOGY SYDNEY

A DATA-DRIVEN DECISION SUPPORT SYSTEM FOR MOBILE TELEMATICS

by Mohammad SiamiNamini

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of Dist. Prof. Jie Lu, and Dr. Mohsen Naderpour

University of Technology Sydney
Faculty of Faculty of Engineering and Information Technology

October 2020

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Mohammad SiamiNamini declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 4/11/20

DEDICATION

To my darling wife for her passion and patience and to my beloved parents for their encouragement which enabled my dreams to come true.

And

To my lovely little son Ariyan, who made me stronger, happier and more fulfilled than I could have ever imagined. I love you to the moon and back.

ACKNOWLEDGEMENT

I would like to express my appreciation to Distinguished Professor Jie Lu, who supervised this Ph.D. research, for her knowledgeable comments, valuable support and guidance during hard times, and constructive suggestions along the way. I want to thank Jie for her wonderful scientific and spiritual support during this journey and her willingness to allow my research to follow my interests. I am also grateful to Dr Mohsen Naderpour, my co-supervisor, for his knowledgeable suggestions and valuable technical advice throughout my PhD study.

Looking back over my PhD journey, I see my wife, Anahita, truly standing shoulder to shoulder with me. Thank you very much for being so supportive in all circumstances throughout the four years of this Ph.D., for understanding the stress I was subjected to, for sacrificing your time for me while you were also completing a Ph.D. and for caring for our little son, Ariyan, to give me the freedom to follow my scientific interests. I could not have accomplished this without your constant love and support.

I would like to thank my parents, the first ones who taught me, for their encouragement and support despite the geographical distance. Pursuing a PhD would not have been possible without their love and assistance. I express my gratitude to all my friends and colleagues in the Decision Systems & e-Service Intelligence (DeSI) Laboratory for their help and valuable comments during my study.

I would also like to express my appreciation to the Faculty of Engineering and Information Technology and the Centre for Artificial Intelligence (CAI) at the University of Technology Sydney (UTS) for their provision of conference registration and travel funds during this research. This research was also supported by

an International Research Scholarship (IRS) and the UTS President's Scholarship (UTSP) funded by UTS.

Last, but not least, a special thank you goes to Ms Sue Felix, Ms Jemima Moore, and Ms Michele Mooney for helping me to identify and correct grammar and syntax problems in my publications.

ABSTRACT

Mobile telematics is an emerging technology that collects data on human behaviour using smartphones. All smartphones have internal sensors with the capability to record and transmit data to an external server. This emerging technology is easy to use, the initial cost is very low, and generates a massive amount of data which are noisy, complex, and uncertain. This opens many opportunities for data-driven decision making such as driving behaviour risk analysis, usage-based insurance, remote sensing, and fleet management. Traditional decision-making techniques are not able to work with this type of unstructured and complex data. Thus, new techniques are needed based on advanced analytics to analyze mobile telematics streams.

This research develops a big data-driven decision support system (DSS) for mobile telematics. The research relies on the capabilities of advanced analytics techniques, machine learning, and fuzzy logic. The research presents an innovative analytical system for mobile telematics which consists of four major components: 1) a data preparation component that prepares a trajectory dataset to a new and ready-for-analysis format; 2) a driving style pattern recognition that extracts hidden human patterns in mobile telematics using unsupervised learning and unlabelled data; 3) a fuzzy risk assessment is proposed to assess risk of drivers by fuzzy logic using extracted patterns by unsupervised learning; and 4) a missing data imputation component which is a novel Choquet Fuzzy Integral Vertical Bagging (CFIVB) algorithm to classify large labelled mobile telematics stream datasets.

The proposed models were evaluated on two real-world mobile telematics datasets, namely an unlabelled dataset collected by a usage-based insurance company

containing 500,000 journeys of 2500 drivers, and an anonymized driving behaviour dataset consisting of streaming data of 408 trips of 310 unique drivers. Various validation measures were used to evaluate the performance of the proposed models. The area under a curve (AUC) and accuracy are used to evaluate the classification algorithms and the Davis-Boulding index, the Calinski-Harabasz index, execution time, and mean square error are utilized to evaluate clustering algorithms and find the optimal number of clusters. The sensitivity analysis results show the proposed model is consistent across different variations of the model.

The proposed DSS can be applied on all stream data risk assessments. Moreover, 29 unique driving styles were extracted from mobile telematics data and these patterns can be applied as labels for supervised learning modelling. In addition, performance measures depict the CFIVB algorithm performs well in this domain, and it can be applied for similar problems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1:

INTRODUCTION

1.1 BACKGROUND

Technological improvements in mobile application development, sensor technologies, the Internet of Things (IoT), and the processing of IoT data have led to a wide range of applications that enhance our lives, including smart homes, healthcare systems, vehicle monitoring, and a greater awareness of environmental problems. IoT applications enable hardware devices to connect with their surrounding environment and each other to report on or accomplish a task. Moreover, they generate huge amounts of data that are useful for behavioural and environmental analytics. Further, growth in the use of smartphones, as one type of interconnected device, is likely to further increase the number of useful IoT applications developed in future years (Wahlström, Skog & Händel 2017).

Telematics, which involves integrating sensors, computer systems, and communications to gather information about a vehicle's operations, is one such IoT application. However, this technology requires different kinds of velocity and acceleration sensors to be installed in the vehicle, which is expensive and difficult to develop. To overcome this problem, Malalur, Balakrishnan & Madden (2013) invented a new kind of telematics, known as mobile telematics, which uses the sensors in smartphones to record and track driving behaviour.

Because most people own a smartphone, mobile telematics offers a new, low-cost alternative for collecting data about driving behaviour (Wahlström et al. 2019).

All smartphones contain at least one component which is capable of measuring position by connecting to a fixed communication system, such as a cellular radio station, WiFi access point, or GPS receiver. Smartphones can also contain a three-axis accelerometer, a gyroscope, and/or a compass. These internal sensors give mobile telematics apps a wide scope to gather driving-style data. The apps are easy to use, and the initial hardware cost is either very low or free if the user already has a smartphone (Desyllas & Sako 2013). Further, the massive amounts of data they collect benefit a range of analytical uses like road safety (Zhao 2002), intelligent transportation systems (Zhao 2000), usage-based insurance (Bowne et al. 2013), and others. Perhaps more importantly, these apps can help people assess and improve their own driving behaviour by providing feedback on their driving styles with incentives to change bad habits (Malalur, Balakrishnan & Madden 2013). Thus, it is unsurprising that one of the biggest beneficiaries of mobile telematics is the insurance industry. With mobile telematics apps, insurers no longer need to rely on expensive in-vehicle sensor installations to take advantage of driver monitoring. As a result, many insurers are specifically targeting drivers who are willing to use mobile telematics with their marketing campaigns (Desyllas & Sako 2013).

All these benefits, however, are predicated on good definitions of driving. Thus, driving behaviour detection methods typically fall into two main groups (Wahlström, Skog & Händel 2017). The first is rules-based detection, which identifies risky habits by defining different thresholds for dangerous and normal behaviour (Song et al. 2019). The rules and thresholds are usually developed by transportation experts in autonomous driving, driving simulation, behavioural risk assessment, and similar fields (Guo et al. 2013). The second approach again relies on transportation experts, this time with a set of predefined templates that describe different driving styles ranging from normal to dangerous. A set of pattern matching algorithms and machine learning models are then used to classify a driver's behaviour according to the most similar patterns (Wahlström, Skog & Händel 2017). Yet, developing

good definitions of something so fluid and dynamic as driving behaviour is difficult, even for experts.

Further, although extensive research has been undertaken on driving style analytics, to the best of our knowledge, only a few studies have investigated decision support systems for mobile telematics. Moreover, much of the research on driving style analysis up to now has been conducted using data collected from questionnaires, site investigations, or laboratory simulations. However, driving behaviour in the real world is completely different from the simulated behaviour in generated data. We believe the dynamic properties of human behaviour mean that simulated data cannot reflect all driving habits.

1.2 RESEARCH PROBLEMS

The main goal of this research is to develop an up-to-the-minute big data-driven decision support system based on mobile telematics, which is a cheaper, easier alternative to in-vehicle data recorders, and one that leverages the current state-of-the-art in machine learning. Even though mobile telematics hold a great deal of promise, there are several challenges to overcome:

- The lack of research on a practical analytical framework to model a decision support system to analyze mobile telematics big datasets is the first obstacle. Modelling this kind of decision support system is very costly and time-consuming because of the behavioural research issues and data collection.
- The availability of labelled data is critical for any machine learning algorithm, and these models minimize their cost functions according to the labelled data, while the labelled data is not available in some domains for mobile telematics big datasets.
- Mobile-telematics-generated data is big with a very complex structure. These data are generated by various IoT devices in real-time, and the volume of these generated data is huge. Also, the collected data is transformed using wireless

network, which is noisy and reduced the quality of these data and makes them noisy, inconsistent, and incomplete (Hariri, Fredericks & Bowers 2019).

- Mobile telematics is unable to provide the driver's demographic features such as gender and age range and applying mobile telematics in business without this information is problematic.

In light of the aforementioned issues, the main objective of this research is to propose a big data-driven decision support system for mobile telematics, which uses the capabilities of fuzzy logic and advanced analytical techniques such as supervised and unsupervised learning models for driving style analysis and risk assessment in mobile telematics and the big data environment. The proposed decision support system has been applied in a domain, which is provided for the usage-based insurance sector to assess the performance of the proposed framework.

This section explains the main issues which significantly motivates this study and presents the research questions:

1) Most decision support systems have been defined on a particular business problem. All business managers agree that a well-defined business problem can be solved much more easily than a poorly defined problem. In addition to problem definition, alternative identification is another critical component of developing a decision support system. This creative step needs special consideration and brainstorming to generate a large number of ideas, alternatives and criteria (Power 2002). Therefore, to propose a new decision support system, firstly we should have a well-defined business problem and all alternatives, DSS components and criteria of the problem should be created innovatively. This process is very challenging and time-consuming.
2) Emerging technologies such as IoT, social media, and sensor technologies in particular mobile telematics generate a massive amount of data with a complex structure. The data generated by these devices are noisy, inconsistent, and incomplete (Hariri, Fredericks & Bowers 2019). Therefore, proposing a data

decision support system, which can consider the uncertain situation within this complex structure with a high level of precision is problematic.

3) Machine learning is the most prominently applied theory for big data analytics. The characteristics of mobile telematics big data is completely different from relational data, so traditional machine learning algorithms do not have a practical application in this environment (Zhu et al. 2018). Therefore, proposing a supervised machine learning and artificial intelligence model to extract hidden patterns from data for decision making in mobile telematics domain is difficult.

4) In addition, supervised learning algorithms have very good performance when there is a good source of labelled data. These algorithms however are not effective without labelled data. The complex structure of mobile telematics is very challenging when developing an unsupervised learning model. Various automatic and manual feature extraction techniques need to be proposed (Liu, Taniguchi, et al. 2017). Hence, proposing an unsupervised learning technique which considers the complex structure of mobile telematics as a kind of big unstructured data is a major issue.

5) Risk assessment is a process that examines the exposure of a planned activity and includes a broad range of tasks. A risk assessment process helps decision makers to understand the exposure associated with particular activities and prioritize them according to risk level. Different quantitative and qualitative methods have been proposed for risk assessment. Quantitative methods aim to provide a numeric score that estimates the risk level of incidents, while qualitative methods evaluate the risk of events based on some qualitative measures or expert opinions (Sengupta et al. 2016). Defining useful criteria for decision making in an uncertain situation such as mobile telematics is a difficult task, and various domain experts with the support of IT professionals should work on a project to define events and criteria and assess their probability and severity.

6) Mobile telematics provides a rich source of behavioural data, but it is unable to find the answer to declarative features as users do not provide this information by self-reporting through complementary mobile apps. Therefore, the mobile telematics domain has a big gap in relation to declarative data.

Based on the aforementioned challenges, the research questions of this study are as follows:

- **Research question 1**: How can we define an analytical decision support system framework for mobile telematics to address the challenges in this domain to help decision makers reach a decision?
- **Research question 2**: What are the characteristics of a practical analytical model in the mobile telematics environment? What components should be included?
- **Research Question 3**: How can we prepare a mobile telematics trajectory dataset to analyze driving behaviours? How can we detect driving behaviour which exhibits significant changes?
- **Research Question 4**: How can we propose an autonomous risk assessment support system for the mobile telematics domain in uncertain situations using massive data streams? How can we evaluate the proposed decision support system?
- **Research Question 5**: How can we define a pattern recognition methodology using unsupervised learning to automatically extract the decision-making criteria from driving patterns? What are the characteristics of an efficient clustering algorithm?
- **Research Question 6**: How can we solve the data quality problem in mobile telematics? How can we propose a supervised learning algorithm using labelled data to improve the quality of data?

1.3 RESEARCH OBJECTIVES

Based on these research problems, the following six research objectives are formulated:

Research objective 1: The first research objective corresponds to the first and second research questions to propose an analytical framework for the mobile telematics big data environment. The framework embeds the general risk assessment components required for analysing mobile telematics, including data preparation, driving style pattern recognition, and fuzzy risk assessment. In addition, the framework uses the capabilities of supervised learning to provide an estimation of null data in mobile telematics.

Research objective 2: The second research objective corresponds to the third research question. To address this objective, we propose a data preparation component to prepare the trajectory data collected by mobile telematics so it is ready for analysis. This component transforms trajectory data to a new format which shows the driving characteristics. Moreover, a change detection algorithm is applied in the proposed component to find the most significant driving events in driving streams.

Research objective 3: The third research objective corresponds to the fourth research question. To address this objective, a decision support system is proposed to learn the hidden driving patterns in big data for decision making. In addition, the proposed decision support systems are evaluated using a sensitivity analysis on a real-world dataset collected by a European insurance company.

Research objective 4: The fourth research objective will address the fifth research question and proposes a pattern recognition framework to extract unknown patterns from mobile telematics big data using unsupervised learning. Thus, an unsupervised learning algorithm is proposed to categorize big data streams into similar groups. The algorithm extracts criteria for the decision-making problem.

Research objective 5: Corresponding to the sixth research question, a new supervised learning algorithm is proposed to improve the missing data problem in mobile telematics using labelled data. To achieve this goal, a novel Choquet Fuzzy Integral Vertical Bagging algorithm is proposed to detect the gender of drivers from the driving data. In addition, a

feature extraction methodology is used to transform unstructured driving data into features that are understandable by machines.

1.4 RESEARCH CONTRIBUTIONS

According to the research objectives, the research contributions of this study are summarized as follows:

(1). The most important contribution of this research is to propose an analytical framework for mobile telematics to support the analytical requirements for risk assessment in this domain with a huge amount of unstructured data. The proposed framework uses the advantage of artificial intelligence, machine learning, unsupervised learning, and fuzzy logic.

(2). A data preparation component is proposed to prepare mobile telematics data for analytics. This component transforms trajectory data to a time-series of driving characteristics. The proposed solution offers a new way of detecting important driving events using the abrupt change detection algorithm. Also, a feature extraction technique is proposed to extract useful features from driving streams. To the best of our knowledge, no study has considered the proposed data preparation techniques for mobile telematics.

(3). An autonomous fuzzy decision support system for mobile telematics risk assessment using the advantages of artificial intelligence, machine learning and fuzzy logic is proposed. The proposed decision support system learns autonomously from big data and the decision support system is evaluated using sensitivity analysis to assess the risk of drivers according to the extracted driving patterns and the probability and severity of these extracted patterns.

(4). An empirical analysis of mobile telematics data is developed to propose a novel unsupervised learning framework specifically for mobile telematics data that extracts significant patterns in lieu of labels. A self-organizing map to reduce the complexity of data, and a deep auto-encoder architecture with nine layers that automatically extracts features from driving characteristics are used to prepare the data for partitive clustering. In addition, an empirical assessment of five partitive clustering algorithms is undertaken to find the best algorithm in the mobile telematics domain.

(5). A new supervised learning algorithm in the mobile telematics big data environment is proposed. We introduce a novel Choquet fuzzy integral vertical bagging classification algorithm with a new application with mobile telematics data. For the first time, we use the driving style dataset collected by mobile telematics devices to detect a demographic feature of a driver using the proposed algorithm.

1.5 RESEARCH METHODOLOGY

Research methodology is the "collection of problem solving methods governed by a set of principles and a common philosophy for solving targeted problems" (Gallupe 2007). Several research methodologies, such as case studies, field studies, design research, field experiments, laboratory experiments, surveys, and action research have been proposed and applied in the domain of information systems. The methodology of this research is planned according to the practice of design research (Kuechler Jr & Vaishnavi 2011; Niu, Lu & Zhang 2009), which has been proposed and applied in information systems.

1.5.1 GENERAL RESEARCH METHODOLOGY

Figure 1-1 depicts the five stages of the design research methodology (DRM). This research methodology was applied by (Niu, Lu & Zhang 2009)

(1). **Awareness of the problem**: In this first step, the limitations of the existing applications are analyzed and the significant research problems are acknowledged. The research

problems reflect a gap between the existing applications and the expected status. Research problems can be identified from different sources: industry experience, observations on practical applications and literature reviews. A clear definition of the research problem provides a focus for the research throughout the development process. The output of this phase is a research proposal for new research effort.

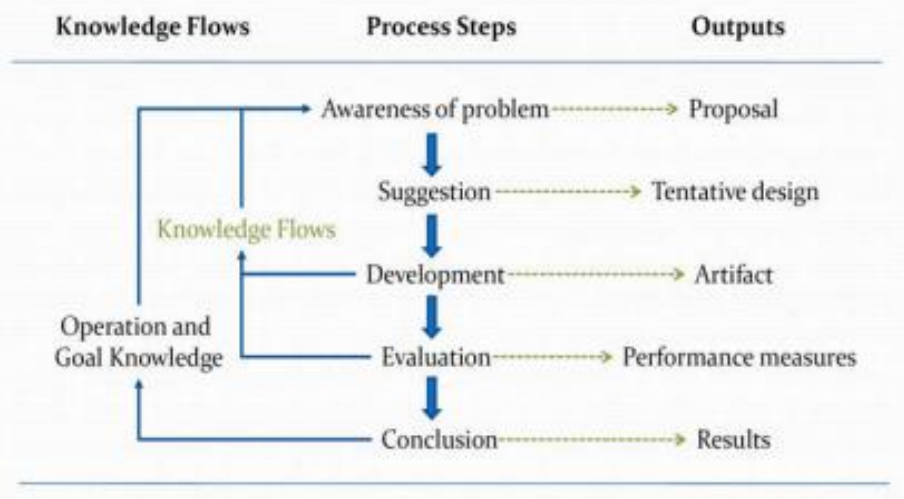

Figure 1-1 - General Research Methodology

(2). **Suggestion**: This phase follows the identification of the research problems where a tentative design is suggested. The tentative design describes what the prospective artefacts will be and how they can be developed. Suggestion is a creative process during which the new concepts, models and functions of artefacts are demonstrated. The resulting tentative design of this step is usually one part of the research proposal.

(3). **Development**: This phase considers the implementation of the suggested tentative design artefacts. The techniques for implementation are based on the artefact to be constructed. The implementation itself can be simple and does not need to involve novelty; novelty is primarily in the design not the construction of the artefact. The development process is often an iterative process in which an initial prototype is first built and then evolves as the researcher gains a deeper comprehension of the research problems. Thus, the output of the suggestion step is also feedback on the first step,

whereby the research proposal can be revised. This step includes the following sub-steps to create the prototype (Niu, Lu & Zhang 2009): a) planning, b) analysis, c) design, d) development, e) testing, f) implementation, and g) maintenance.

(4). **Evaluation**: This phase considers the evaluation of the implemented artefacts. The performance of the artefact can be evaluated according to criteria defined in the research proposal and the suggested design. The evaluation results, which might or might not meet expectations, are fed back to the first two steps. Accordingly, the proposal and design might be revised and the artefacts might be improved.

(5). **Conclusion**: This is the final phase of a design research effort. However, there are still deviations in the behaviour between the suggested proposal and the artefacts that are actually developed. A design research effort concludes as long as the developed artefacts are considered to be 'good enough' wherein the anomalous behaviour may well serve as the subject of further research.

1.5.2 THESIS RESEARCH PROCESS

This research was planned according to the general research methodology (GRM), which Figure 1-2 shows steps of this research.

Step 1: According to the GRM, the first step is defining the research problem by focusing on the limitations of the existing methods and the major industrial problems. The research problem can arise from observations, from personal interest, or from the current literature. The selected research problem was chosen based on the previous literature and industrial experience on the real-world project in data-driven decision support systems. Then, the current studies on this topic were reviewed to find the existing gaps in this area. After identifying the research gaps, we defined the research problems to address the gaps extracted from the current literature. We also devised the various research questions for this research project.

Step 2: The lack of an analytical method for mobile telematics leads us to propose a decision support system in this domain to help decision makers in relation to risk assessment

and to cover the gap in declarative data. In this research, we propose an analytical DSS framework with five components to analyze driving behaviour using fuzzy logic and supervised and unsupervised learning techniques.

Step 3: Data preparation is the first component in the proposed system that prepares data for analytics. This component removes unnecessary data using a change detection algorithm and feature extraction techniques. In the following steps, the data prepared with this component is used for supervised and unsupervised learning tasks.

Step 4: One of the main objectives of this research is to propose an autonomous decision support system which extracts criteria for decision making automatically using artificial intelligence and machine learning. Therefore, firstly, we an empirical analysis on driving characteristics using information collected by mobile telematics devices to find an efficient clustering algorithm in this domain with an optimal number of clusters. Moreover, driving behaviours are categorized into similar groups using this algorithm.

Step 5: After extracting the criteria for decision making from the driving patterns, we propose an autonomous data-driven decision support system with the ability to extract criteria from decision making automatically using fuzzy clustering. In this step, we propose a novel fuzzy DSS that innovatively assesses the risk of driving events using fuzzy logic according to the extracted patterns that learn from big data.

Step 6: As the missing data problem is a major issue in the mobile telematics domain, the missing data imputation component is proposed to improve the quality of data. A supervised learning algorithm is proposed to impute the unknown variables from driving behaviours to provide managerial insights for decision makers. In this case, a novel Choquet fuzzy integral vertical bagging algorithm is introduced to classify driving patterns and driving characteristics.

Step 7: After proposing new frameworks and algorithms, we evaluate the proposed methods in the mobile telematics datasets. Our proposed methodology may have unexpected results so we should review and revise our methodology to achieve suitable results. Different validation methods for each are proposed for each part of the model.

1.6 THESIS STRUCTURE

This thesis comprises nine chapters as shown in Figure 1-2. The research problems, background, questions, objectives and contributions, and the research methodology are introduced in Chapter 1. Chapter 2 reviews the literature on mobile telematics, driving style analytics and related works. Chapter 3 explains the big-data driven decision support system for mobile telematics. Chapter 4 proposes the data preparation component.

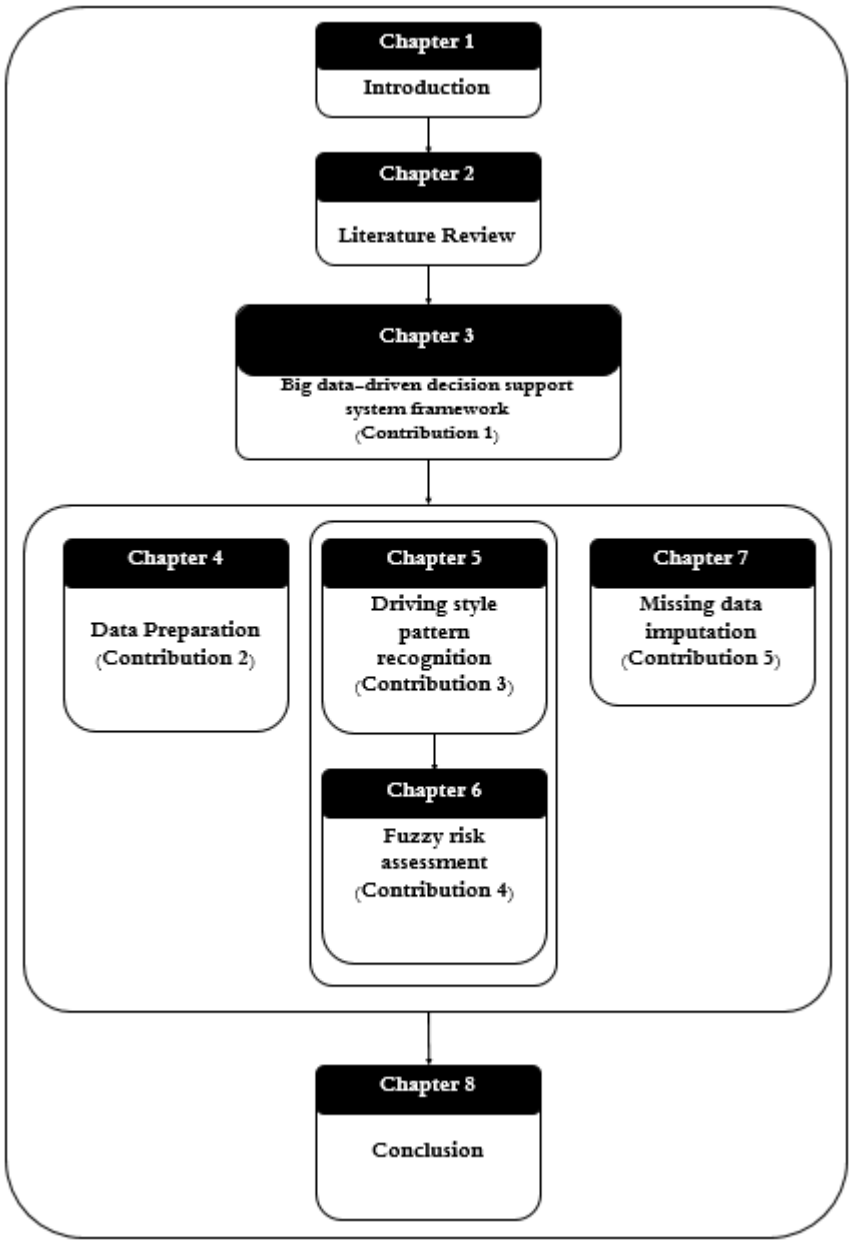

Figure 1-2- Thesis Structure

Chapter 5 proposes an unsupervised learning framework to extract driving patterns from mobile telematics data. Chapter 6 introduces a fuzzy risk assessment component for driving style risk assessment. Chapter 7 proposes the missing data imputation, which improves data quality, and finally chapter eight presents the conclusion and future research direction of this study.

1.7 PUBLICATIONS OF THIS RESEARCH

(1). M. Siami, M. Naderpour and J. Lu, "A Mobile Telematics Pattern Recognition Framework for Driving Behavior Extraction," in **IEEE Transactions on Intelligent Transportation Systems**, vol. 22, no. 3, pp. 1459-1472, March 2021, DOI: 10.1109/TITS.2020.2971214.

(2). M. Siami, M. Naderpour and J. Lu, "Risk Assessment Through Big Data-An Autonomous Fuzzy Decision Support System" submitted to ***IEEE Transactions on Fuzzy Systems*** (Major Revision)

(3). M. Siami, M. Naderpour and J. Lu, "A Choquet Fuzzy Integral Vertical Bagging Classifier for Mobile Telematics Data Analysis," 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), New Orleans, LA, USA, pp. 1-6, doi: 10.1109/FUZZ-IEEE.2019.8858812 (**ERA Tier A Conference**).

(4). M. Siami, M. Naderpour and J. Lu, " Generating a Risk Profile for Car Insurance Policyholders: A Deep Learning Conceptual Model," 2017, in Australasian Conference on Information Systems (ACIS), Hobart, Tasmania, Australia (**ERA Tier A Conference**).

(5). M. Siami, A. Namvar, M. Naderpour and J. Lu, "A fuzzy telematics data-driven approach for vehicle insurance policyholder risk assessment", in 13th international conference on Data Science and Knowledge Engineering for Sensing Decision Support (FLINS 2018), Belfast, North Ireland, UK (**ERA Tier B Conference**).

Chapter 2:

LITERATURE REVIEW

2.1 INTRODUCTION

To gain a better understanding of this thesis, this chapter reviews current literature on mobile telematics, driving style analytics and decision support systems related to this study. Moreover, I have provided some background regarding used algorithms and related works for this research.

2.2 MOBILE TELEMATICS

A telematics device is a kind of in-vehicle data recorder that includes a GPS sensor with the ability to transmit data to a remote server. It is a hardware device that can be incorporated into a vehicle and can record the characteristics of driving habits. These devices have been introduced to track the behaviour of drivers and generate a dynamic risk profile according to their driving characteristics. Mobile telematics is an easy to use and inexpensive alternative to the telematics which are available today through smart phones.

Duri et al. (2002) provide an overview of telematics applications in the automotive industry. Each car is equipped with sensors and communication devices and a computer with enough storage space and processing capabilities to run embedded applications. The

computer interface collects different data generated from a global positioning system (GPS) on car location and car engine performance for future processing.

In-vehicle data recorders were very expensive, so in recent years there has been a tendency to use mobile telematics instead of vehicle telematics. This monitoring technology uses embedded velocity and acceleration sensors together with a GPS in mobile hardware to transmit data on driving characteristics to an external server, which stores the data for future analytics. In-vehicle data recorders were difficult to implement and very expensive to install, therefore an inexpensive version of telematics was introduced by (Malalur, Balakrishnan & Madden 2013). Mobile telematics has all the capabilities of telematics in-vehicle data recorders but is cheaper to implement.

All smartphones contain at least one instrument capable of measuring position by connecting to a fixed communication system, such as a cellular radio station, Wi-Fi access point, or GPS receiver. Smartphones can also contain a three-axis accelerometer, a gyroscope, and/or a compass. These internal sensors give mobile telematics apps a wide scope to gather driving style data. The apps are easy to use, and the initial hardware cost is either very low or free if the user already has a smartphone (Desyllas & Sako 2013). Further, the massive amounts of data they collect benefit a range of analytical uses like road safety (Zhao 2002), intelligent transportation systems (Zhao 2000), usage-based insurance (Bowne et al. 2013), and others. Perhaps more importantly, these apps can help people assess and improve their own driving behaviour by providing feedback on their driving styles with incentives to change bad habits (Malalur, Balakrishnan & Madden 2013). Thus, it is unsurprising that one of the biggest beneficiaries of mobile telematics is the insurance industry. With mobile telematics apps, insurers no longer need to rely on expensive in-vehicle sensor installations to take advantage of driver monitoring. As a result, many insurers are specifically targeting drivers who are willing to use mobile telematics in their marketing campaigns (Desyllas & Sako 2013).

2.3 MOBILE TELEMATICS AND USAGE-BASED INSURANCE

An insurance agreement is an arrangement that transmits the risk of accident from car owner to the insurance company by paying a fee, but each person has a different level of risk based on their claim frequency and severity. According to (Ohlsson & Johansson 2010), an insurance premium is calculated as follows:

Insurance premium = claim frequency * Claim severity

where claim frequency is the probability of claims being made and severity is the size (amount of money) which is claimed by the customer. The proposed equation is useful for insurers to calculate the risk of their customers based on historical data and as a result, a driver might pay less for an insurance policy because he has not made any claims, however a major problem for insurers is that they are unable to make a decision based on driving style. In recent years, telematics devices have provided a new opportunity for insurance companies to calculate the risk of each policyholder based on their driving styles.

There is a relatively small body of literature on usage-based-insurance (UBI) which considers the value of telematics and mobile telematics devices. Table 2-1 overviews the most important ones from the current studies which collected data from telematics and used these for insurance and accident risk prediction.

The first sample of usage-based insurance using telematics was proposed by Vaia et al. (2012) . This product was the result of cooperation between Unipol, one of the largest insurers in Italy and Octo Telematics as a technology provider. They described the advantages of using this technology for all the involved parties. They proposed a two-stage methodology for calculating the premium for customers in the first year based on typical parameters such as mileage and total travelling time. They only introduced a telematics device as a technology for gathering data, but they could not propose a methodology to use these data for risk assessment based on driving styles. Azzopardi & Cortis (2013) provided a SWOT analysis for the application of telematics-based insurance to compare this new approach with the traditional one. Their results indicate that telematics could improve fleet management and

also manage insurance risk and the adoption of this technology in the insurance area would impress the industry. An overview of a technical solution for telematics systems which takes into account collecting, communicating, managing and analysing problems of usage-based insurance was proposed by (Husnjak et al. 2015). They also introduced the social, economic and environmental benefits of UBI for insurance service providers and users and provided some measures for premium calculation and the billing process of car policyholders based on their raw geospatial data. Jun, Guensler & Ogle (2011) developed a way to find the velocity patterns of crash involved and crash not-involved drivers. They evaluated the speed of different drivers from GPS data from light-duty vehicles. They found that most drivers who were involved in a crash had more high-speed experiences than drivers without a crash experience. Their results also indicate that drivers with crash had many instances of driving over the speed limit.

Table 2-1 - Usage-Based Insurance and Mobile telematics

Study	Study purpose	Sample	Measures/techniques	Relevant findings
(Vaia et al. 2012)	A Novel telematics-based usage-based insurance	--	Number of excessive speed events per 100 miles Number of hard braking events per 100 miles	New usage-based insurance with telematics Offer premium based on driving styles and data First year and other years Describes benefits for all stakeholders in insurance ecosystem New opportunities for IT service providers

Study	Study purpose	Sample	Measures/techniques	Relevant findings
(Azzopardi & Cortis 2013)	SWOT analysis	25 insurance stakeholders	--	Advantages and disadvantages of the adoption of telematics-based insurance
(Husnjak et al. 2015)	New billing method	Survey of 22 drivers	Location of the vehicle (a GPS point) Excessive forces acting on the vehicle	70% of 22 participants indicate a positive impact on their driving score. UBI could reduce the average claim frequency after 1 year by up to 30%
(AF Wåhlberg 2004)	Accident risk and driving behaviours	125 bus drivers	Correlation analytics between acceleration and accident risk	Sample is too small
(Toledo, Musicant & Lotan 2008)	Driver behaviour monitoring	191 drivers	Poisson regression on speed, acceleration and location	New methodology for detecting manoeuvres Risk calculation based on detected manoeuvres There is a correlation between risk score and drivers' crash records
(Baecke & Bocca 2017)	The value of vehicle telematics data in insurance risk selection processes	6984 vehicles (age < 30)	Logistic regression Artificial neural networks Random forest	Computational intelligence risk prediction model for telematics data
Handel et al. (2014)	Studying the advantages of smartphone data for data gathering in insurance	40-minute drive with different smart phones with different operating systems	Assessing the quality of the data gathered by smartphones in terms of accuracy, integrity, availability, and continuity of service Polynomial regression HDOP monitoring Position-speed time residual Sample time variation	Highlighted technical challenges of UBI with telematics devices Introduced a new way of measuring driving characteristics with smartphones

Study	Study purpose	Sample	Measures/techniques	Relevant findings
(Dong et al. 2016; Dong et al. 2017)	Driver identification and risk assessment with deep learning	200 trips of more than 2500 drivers	Convolutional Neural Networks (CNNs) Recurrent Neural Networks (RNNs) Auto-encoders	New deep learning architecture to identify driving style Estimating the number of drivers with deep autoencoder
Liu, Taniguchi, et al. (2017)	A novel visualization method to connect driving styles to colours in an RGB colour model	12958 frames of driving behaviour data in total at a frame rate of 10 fps.	Deep sparse auto encoder Accelerator opening rate Engine speed Cylinder pressure Longitudinal acceleration Steering angle Speed meter	Extracting unique driving patterns with deep sparse auto-encoder

Telematics devices are very expensive and difficult to implement so Malalur, Balakrishnan & Madden (2013) invented a new kind of telematics, known as mobile telematics and Handel et al. (2014) used mobile telematics data which are generated by smartphones for risk assessment in usage-based insurance. They indicated that smartphone data could be a suitable and less expensive substitute for telematics data. They also introduced some major attributes such as acceleration, braking, speeding, smoothness, swerving, cornering, and etc. These attributes can be collected from mobile telematics data and could be applicable for calculating the risk of drivers in real-world situations. These attributes are shown in Table 2-2. Then, they proposed scoring methodologies to calculate the risk of each driver based on their historical data which are extracted from the driver's smartphones.

Table 2-2 - Attributes for Risk Assessment with Smartphone Data

Driving attributes	Description
Acceleration	Number of rapid acceleration events and their harshness
Braking	Number of harsh braking events and their harshness
Speeding (absolute)	Amount of absolute speeding
Speeding (relative)	Amount of speeding relative to a location-dependent limit
Smoothness	Long-term speed variations around a nominal speed
Swerving	Number of abrupt steering manoeuvres and their harshness
Cornering	Number of events when turning at a too-high speed and their harshness
Eco-ness	Instantaneous or trip-based energy consumption or carbon footprint
Elapsed time	Time duration of the trip
Elapsed distance	Distance of the trip
Time of day	Actual time of day when making the trip
Location	Geographical location of the trip

2.4 FUZZY DECISION SUPPORT SYSTEM

The first time that the application of fuzzy logic and fuzzy set theories were applied to decision analysis and decision support systems was in the early 1970s (Zimmermann 1998). Since that time, various applications of fuzzy logic have been proposed to handle uncertain situations in decision-making processes. Lu et al. (2019) divided decision support systems into two categories, namely traditional decision support systems and data-driven decision support systems. Firstly, traditional decision support systems have been applied over the years for decision making. Multi-criteria decision-making (MCDM) is one of the first in model-driven DSSs. In an MCDM, various decision-making techniques have been applied to find the best alternative, such as simple linear weighing; the technique for order of preference by similarity to ideal solution (TOPSIS); analytic hierarchy process (AHP); etc. (Bao, Wu & Li 2018). The fuzzy version of these techniques has been used to model the uncertainty of the environment in complex situations (Dincer et al. 2016). A risk-based fuzzy DSS is proposed

by Seiti et al. (2019) using a fuzzy MCDM approach to assess the failure of components and equipment because of the lack of available information to apply quantitative models. They used fuzzy numbers to explain reliability, associated risks, and error for analyzing metrics. They evaluated the effectiveness of the proposed method across different scenarios in a steel plant case study, and the results gave flexibility and confidence to decision-makers to handle the risk of uncertain situations. In another study proposed by Zhu, Hu & Ren (2020), the uncertainty during the decision-making process was suitably handled (Zhu, Hu & Ren 2020). They presented a fuzzy rough number for design concept evaluation and used this concept in two methodologies, namely fuzzy AHP and fuzzy TOPSIS. The results showed that the fuzzy rough number had an outstanding performance for group decision making. These traditional forms of decision-making require a set of options and criteria to rank alternatives according to the goal of a decision-maker. Moreover, multiple criteria should be defined, experts and stakeholders should answer the related questions, and finally, numerical values are processed to select or classify one choice (Mulliner, Malys & Maliene 2016).

The second category is the data-driven decision support system. By integrating diverse operational databases with data warehouse technology in the late twentieth century, structured data has been widely used to support decisions (Shim et al. 2002). This integrated data contains invaluable information about the future to make better data-driven decisions. This data stores both internal and external information that is available through transactional systems or the Internet in an integrated data warehouse, which plays an important role in data-driven decision making (Huber et al. 2019). Fuzzy risk assessment is widely used to apply data-driven decision support systems for risk evaluation. Namvar et al. (2018) proposed a data-driven decision support system to assess the risk of lenders in financial service companies. They proposed a machine learning framework in a peer-to-peering lending environment. Their results show that supervised learning machine learning models could help decision-makers in relation to risk assessment in banking and automatic credit risk scoring. Another study (Naderpour, Lu & Zhang 2014), proposed an intelligent situation awareness support

system to manage abnormal situations, including hardware failure and human error. They assessed the risk of abnormal events using Bayesian networks and fuzzy logic in a safety-critical environment.

Recent advancements in information systems and big data on one hand, and advancements in artificial intelligence and machine learning algorithms on the other hand, provide new opportunities for decision-makers to use big-data-driven decision support systems in more innovative ways. Sensor data is noisy, and the analytical results produced from this data will be more sensitive to errors due to the increase in the volume, velocity, and variety of data. Currently, most of the studies on big data-driven DSSs that have been proposed are based on the capabilities of the supervised learning algorithms and labelled data (Chan et al. 2017), but labelled data is not easily accessible in real-world problems. Moreover, according to the study by (Lu et al. 2019), using unsupervised learning techniques in a data-driven decision support system is still a source of concern. In addition, according to the study of Shukla, Muhuri & Abraham (2020), although extensive research has been carried out on big data-driven decision support systems, few studies exist on the application of fuzzy logic to reduce the uncertain situation of big data. Therefore, to cover the aforementioned gaps, in this study, we propose an autonomous fuzzy decision support system using the advantage of the unsupervised learning algorithm and fuzzy logic for risk assessment through big data.

2.5 Driving Style Analytics

Wahlström, Skog & Händel (2017) divided the practical applications of mobile telematics into seven categories: navigation, transportation mode classification, cooperative intelligent transportation systems, mobile cloud computing, driver behaviour classification, and monitoring road conditions. Our focus is on driving style analytics and, within this, driver behaviour classification and pattern recognition.

According to the study of Wahlström, Skog & Händel (2017), driving behaviour classification methods typically follow one of two approaches. The first is to define driving behaviour according to one or more thresholds. For example, "safe acceleration" might be

defined as when the norm of acceleration or deceleration is less than $2\ {}^{m}/_{s^2}$; velocity changes beyond this threshold would be classified as extreme events (Fazeen et al. 2012). The second approach is to define a range of templates that represent different driving behaviours. For example, a harsh cornering event might be defined in a template by an acceleration value on the x and/or y-axis during a specified time window. Harsh cornering events by drivers are then identified by calculating the similarity of their behaviour to the template definition.

However, technological advancements, and particularly the integration of machine learning into pattern matching algorithms, now provide opportunities to classify driving styles more acutely than ever before (Saiprasert, Pholprasit & Thajchayapong 2017; Shou & Di 2018). For instance, Wang & Xi (2016) proposed a binary classification solution to distinguish aggressive driving patterns from moderate ones. Their method involves a support vector machine (SVM) and k-means clustering to decrease execution times and improve prediction accuracy. The k-means clustering algorithm first reduces the complexity of the input data, then SVM distinguishes between normal and abnormal driving styles. Cross-validation experiments show the approach to be faster and more accurate than SVM alone. In another study, Henriksson introduced a pattern recognition framework to identify driving contexts from vehicle-generated data. City driving styles were compared to open road driving by finding the hidden relations between driving attributes in these two contexts. In a comparison between SVM and a hidden Markov model, the results show SVM to be more reliable.

Using a driving behaviour monitoring system, Yu et al. (2017) categorized unusual driving behaviours into six groups: weaving, swerving, sideslipping, fast U-turns, turning with a wide radius, and sudden braking. Their method not only distinguishes between normal and abnormal driving patterns but also specifies the type of dangerous driving behaviour. In a comparison between SVM and a neural network as a training algorithm for the classification model, the neural network model was better able to detect dangerous driving patterns.

Driver identification is another research area in driving style analytics. To date, researchers have applied several artificial intelligence and machine learning algorithms to identify who is

behind the wheel. A data transformation method was proposed by Dong et al. (2016) to transform trajectory data into information that is usable in deep learning. They used a convolutional neural network (CNN) and a recurrent neural network (RNN) to distinguish drivers from passengers in a real-world dataset collected by a European insurance company. In a subsequent study, Dong et al. (2017) proposed another model based on an auto-encoder regularized network (ARNet) to estimate the total number of drivers using one vehicle. The algorithm contains multiple levels of neural networks, including a gated recurrent unit (GRU), an auto-encoder, and logits. Insurance companies can take particular advantage of these models because underwriters are very interested in how many people are actually driving a car, especially when policies and premiums are linked to the age and number of drivers. In another study, a driver identification methodology was proposed by Moreira-Matias & Farah (2017) using trip-based historical datasets collected by in-vehicle data recorders to identify the category of driver behind the wheel. They took the advantage of driver-labelled trip data to build a pattern of different drivers in different categories using various supervised learning algorithms.

The aforementioned methods are all supervised learning techniques that have shown outstanding performance in comparison to traditional methods of driving analytics. In fact, most current studies on driving style analytics with machine learning techniques are conducted in supervised learning scenarios. Only a few consider unsupervised methodologies for driving style analytics. One study by Liu, Taniguchi, et al. (2017) maps driving style patterns into three-dimensional data so as to visualize each pattern as a different colour . A deep auto-encoder framework reduces the data streams into three-dimensional data. Each dimension is then mapped to either red, green, or blue – one colour for each unique behaviour – and the auto-encoder extracts the features from the behaviours. However, using their framework in real-world scenarios is somewhat challenging because they used synthetic data to train the deep learning model. In the real world, many data are uncharacteristic and completely different from the data generated in a laboratory. Lee & Jang (2017) also proposed

an unsupervised learning framework to characterize driving style patterns, this time with data generated by in-vehicle data recorders. However, their study did not extend to exploring the performance of different clustering algorithms for driving style extraction. Moreover, the correlation between their results and driving styles described in the literature was not fully investigated. These issues, combined with the problem of in-vehicle data, warrant further study in a mobile telematics setting. Shouno (2018) incorporated a variational auto-encoder into a deep unsupervised learning framework for the purposes of reducing the input dimensions down to a two-dimensional space. Driving styles were then characterized according to a topological map. He tested his framework on a Honda driving simulator with 59 drivers, which again, is simulated data and completely different from those found in the real-world.

2.6 CHANGE DETECTION ALGORITHM

The change detection algorithm is an algorithm to find time windows of major change within time-series data – most commonly through statistical techniques. Change detection algorithms have a wide range of applications, e.g., signal segmentation (Basseville & Nikiforov 1993), climate change detection (Itoh & Kurths 2010), and driving behaviour analytics (Lee & Jang 2017).

Let us consider $\mathcal{Y}(t) \in R_d$ as time-series data with d dimensions at time t, and $y(t) = [Y(t)^T,\ Y(t+1)^T, \dots,\ Y(t+k-1)^T]^T \in R_{dk}$ is a consecutive time window of length k at time t. Following Liu et al. (2013) strategy, the dissimilarity between $y(t)$ and $y(t+n)$ is calculated from the equation below, and the result is used as a change score to reflect the amount of change between two time windows.

$$ChangeScore = D(p_t||p_{t+n}) + D(p_{t+n}||p_t) \quad (2\text{-}1)$$

where p_t and p_{t+n} are the probability distributions of $y(t)$ and $y(t+n)$. For simplicity, hereafter, we denote this dissimilarity as $D(p||p')$ instead of $D(p_t||p_{t+n})$.

To calculate the dissimilarity measure between two different time segments, Liu et al. (2013) proposed the relative unconstrained least-squares importance fitting (RuLSIF)

algorithm. RuLSIF calculates the change between two consecutive time windows with a density-based dissimilarity measure:

$$D(p||p') = -\frac{\alpha}{2n}\sum_{i=1}^{n}\hat{g}(Y_i)^2 - \frac{1-\alpha}{2n}\sum_{i=1}^{n}\hat{g}(Y_i')^2 + \frac{1}{n}\sum_{i=1}^{n}\hat{g}(Y_i) - \frac{1}{2} \quad (2\text{-}2)$$

where n is the window size and $Y_i \; and \; Y_i'$ are two consecutive time windows in d-dimensional time-series data. $\hat{g}$ is the density-ratio estimation of the data samples and α is a constant variable.

This algorithm plays an important role for data preparation component in this thesis. We used this algorithm for removing unrequired driving data to improve the quality of data for machine learning.

2.7 DEEP AUTO-ENCODER

The deep auto-encoder algorithm is a type of artificial neural networks which efficiently codes a dataset for dimension reduction. In recent years, the auto-encoder has been used in many research fields and it has an outstanding outcome. According to (Liu, Wang, et al. 2017), the structure of AE is similar to MLP with the following certain similarities and differences:

- AE consists of a one hidden layer feed-forward neural network such as MLP.
- MLP predicts the target value but AE reconstruct the input values
- AE has identical nodes in the input and output layers

The AE uses a weight matrix ω to convert the input vector x into a hidden h in the coding process. Then, in the next step AE decodes h to the $\tilde{x}$ by using the ω' matrix which should be the transpose of ω. AE decreases mean square errors (MSEs) which is the difference between $x \; and \; \tilde{x}$.

A deep auto-encoder model is a group of several auto-encoders that are arranged in a neural network architecture. A simple auto-encoder has two parts, an encoder and a decoder. An example of a deep auto-encoder is shown in Figure. 2-1.

In the encoding layer, the encoder function $h = f(Wx + b)$ is used for each layer to encode the input data. The encoding stage continues up to the middle layer, at which point a decoder function $h = f(W'x + b')$ begins to reconstruct the encoded input data. Sigmoid, tanh, soft sign, and ReLU functions are the most prominent activation functions for encoder and decoder functions (Zhang et al. 2018).

The set of parameters for a basic auto-encoder comprises $W_l, W'_{l'}, b_l, b'_{l'}$. These parameters are trained to minimize the loss function by

$$loss = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i)^2 \qquad (2\text{-}3)$$

The training procedure is unsupervised and the middle layer represents the encoded version of input data (Zhang et al. 2018). In this PhD research, we use a deep auto-encoder to automatically extract the features from the driving style data.

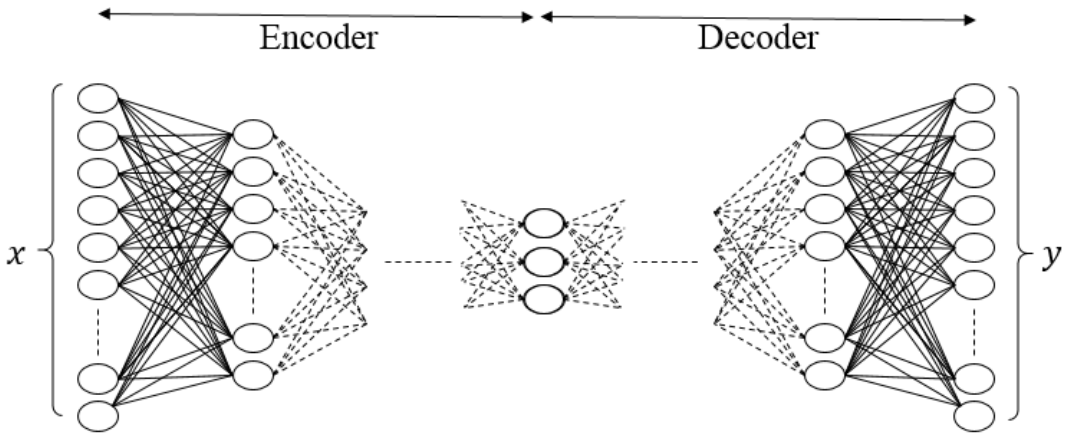

Figure 2-1- A Deep Auto-Encoder With Many Layers

2.8 FUZZY CLUSTERING

The fuzzy and k-means clustering algorithms are very similar and they achieve outstanding performance for pattern recognition. K-means clustering provides a discrete clustering result, which means each member is part of only one cluster, while fuzzy clustering offers more information in comparison to k-means by providing a range score between zero and one, which is the similarity between members and clusters (Heil et al. 2019). This characteristic is very useful for the pattern recognition of driving style, where most driving

behaviours are similar to each other, thus making it difficult for transportation experts to easily specify a discrete cluster for each different driving behaviour. Therefore, in this research, we use fuzzy clustering for the pattern recognition of driving style.

Let $X = \{X_1, X_2, \dots, X_n\}$ as a multidimensional input dataset. Fuzzy clustering categorizes n items into c clusters by developing an optimization process with the following objective function (Gionis, Mannila & Tsaparas 2007):

$$J_m^{(FCM)}(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \left\|x_j - v_i\right\|_2^2 \quad (2\text{–}4)$$

where u_{ij} is a membership function, $\forall\, j = 1, \dots, n. \sum_{i=1}^{c} u_{ij} = 1$. $V = \{v_1, v_2, \dots, v_c\}$ represents the center of clusters. The m is a fuzzy factor and should be $m > 1$ and usually is set as 2. FCM uses the following optimization steps to reach the optimal situation:

$$U^{(t+1)} = argmin\, J_m^{(FCM)}\{U, V^{(t)}\} \quad (2\text{–}5)$$

$$V^{(t+1)} = argmin\, J_m^{(FCM)}\{U^{(t+1)}, V\} \quad (2\text{–}6)$$

where the number of iteration steps is represented by t. $V^{(0)}$ and $U^{(0)}$ are initiated randomly and their values are updated through the optimization procedure. The membership function values and the vector of cluster centers are calculated using the following equations:

$$u_{ij}^{(t+1)} = \left(\sum_{k=1}^{c} \left(\frac{\left\|x_j - v_i^{(t)}\right\|_2^2}{\left\|x_j - v_k^{(t)}\right\|_2^2} \right)^{\frac{1}{m-1}} \right)^1 \quad (2\text{–}7)$$

$$v_i^{(t+1)} = \frac{\sum_{j=1}^{n} \left(u_{ij}^{(t+1)}\right)^m x_j}{\sum_{j=1}^{n} \left(u_{ij}^{(t+1)}\right)^m}$$

2.9 FUZZY SETS AND FUZZY LOGIC SYSTEMS

The fuzzy set theory was first introduced by Zadeh (1965). He proposed this logic to simulate uncertain situations in the human brain using a membership function between zero

and one. The significant difference between fuzzy logic and a crisp concept is the Boolean concept that a particular object could have a specific value or not, but in fuzzy logic, a particular value is given a range from zero to 1. This membership function helps experts to define linguistic variables for the input and output of their systems.

Definition 2–1. Fuzzy set Zadeh (1965): A is a fuzzy set and represents a universal set X by a membership function. $\forall \in X, \mu_A(x) \in [0,1], i.e. A: X \rightarrow [0,1].$

Definition 2–2. α-cut Zadeh (1965).: The α-cut or α-level set of the fuzzy set A is the crisp set Aα defined by:

$$A_\alpha = \{x \epsilon X \mid \mu_A(x) \geq \alpha\}$$

Definition 2–3. Fuzzy number Zadeh (1965).: A fuzzy set A in $\mathbb{R}$ satisfies the following conditions:

- A is normal,
- Aα is a closed interval for every $\alpha \in (0,1]$,
- The support of A is bounded.

Definition 2–4. Fuzzy logic system (FLS) (Siami et al. 2018). A simple fuzzy logic system consists of three phases:

1) Fuzzification,

2) A fuzzy interface engine,

3) Defuzzification.

In the first step, crisp inputs and variables are transformed to fuzzy sets. Then, a fuzzy interface engine defines the relationship between fuzzy input and output variable and finally, the fuzzy output variable is transformed into a crisp output in the defuzzification process.

2.10 PARTITIVE CLUSTERING

Partitive clustering is an unsupervised learning technique that clusters unlabeled input data into a number of partitions, i.e., members are grouped according to distance-based similarity. Partitive clustering algorithms assume that the input data can be categorized into prototypes; thus, they are also known as prototype-based clustering algorithms. The main goal is to compress the data into these prototypes. Each partitive clustering algorithm has different methods of defining the prototypes for the input data. For example, one of the most famous partitive clustering algorithms, k-means, uses the K-means++ algorithm to find the initial prototypes (Xiao & Yu 2012). Partitive clustering algorithms have been used in a wide range of applications, from big data clustering (Fahad et al. 2014) for customer segmentation (Lu et al. 2014; Namvar, Ghazanfari & Naderpour 2017), to weather prediction (Wang et al. 2018), to biomedical health (Khanmohammadi, Adibeig & Shanehbandy 2017), and many others. The main steps of a partitive clustering algorithm are outlined in Algorithm 2-1.

Algorithm 2-1 - Partitive Clustering Algorithm (Xiao & Yu 2012)

Input: Dataset and K number of prototypes, M max iteration ***Output: data points with a cluster label***
1. ***Initialize K data points from the input data as initial cluster prototypes.*** 2. ***Assign each data point to the closest prototype using a distance function.*** 3. ***Recalculate the center of each cluster with these new data points.*** 4. ***Repeat steps two and three if the clusters do not change significantly.***

2.11 SELF-ORGANIZING MAP

A self-organizing map (SOM) is a special type of unsupervised learning algorithm that generates a discretized map of an input space. SOMs have become a common technique in a wide range of applications, such as data visualization, dimension reduction, and vector quantization (Kohonen 1990). The main advantage of SOM is that they reduce computation

costs, which is particularly valuable if clustering is part of one's strategy. Given the complexity of calculating distances within multi-dimensional data, most clustering algorithms are computationally greedy, even with a small number of records. SOM decreases computation costs by abstracting a prototype of the input data. A clustering algorithm can then be used to classify the abstracted data instead of the full dataset (Vesanto & Alhoniemi 2000). Another advantage of SOM is its ability to tolerate noise. Each node in a SOM represents a group of input data, so it is less sensitive to data generated in noisy environments (Du et al. 2015). In contrast, one of the greatest weaknesses of SOMs is detecting outliers. By definition, outliers are rare data points and therefore, SOMs have difficulty generating a suitable prototype to represent those data (Mangiameli, Chen & West 1996).

The gist of these algorithms is to map the input data into a topographical map with N nodes on a regular two-dimensional rectangular or hexagonal grid, where each node has d number of features with a weight $\omega_i = [\omega_{i1}, \omega_{i2}, \dots, \omega_{id}]^T$. The algorithm is iterative. In each iteration step t, a data sample $x(t)$ is randomly selected from the training data, and the distances are calculated between $x(t)$ and all the nodes. The most similar node to $x(t)$ is selected with

$$c = argmin(dist(x(t), \omega_i), \quad \forall\, i\ in\ [1,2,\dots,N] \qquad (2\text{-}8)$$

where $dist(x(t), \omega_i)$ is equal to the distance of the sample $x(t)$ with the ith node.

After a "winning" neuron has been selected, it and its neighboring neurons are updated with a weight updating rule:

$$\omega_k(t+1) = \begin{cases} \omega_k(t) + \gamma(t) h_{kc}(t).\left(x(t) - \omega_j(t)\right), \forall k \in N_c \\ \omega_k(t), \qquad else \end{cases} \qquad (2\text{-}9)$$

where Nc is the winning neuron's neighbors, and $\gamma(t)$ is the learning rate, which is reduced in each iteration (t) with the following equation:

$$\gamma(t) = \gamma_0 . exp\left(-\alpha . \frac{t}{\tau}\right) \qquad (2\text{-}10)$$

where γ_0 is the initial learning rate, α is the exponential decaying constant, and τ is the maximum number of iterations. $h_{kc}(t)$ is a neighborhood kernel function that indicates the distance of the kth neuron to the winning neuron c, as calculated by

$$h_{kc} = \exp\left(-\frac{[(x_k - x_c)^2 + (y_k - y_c)^2]}{2(\sigma(t)^2)}\right) \quad (2\text{–}11)$$

where $\sigma(t)$ is equal to the width of the neighborhood function and decreases in each iteration t by

$$\sigma(t) = \gamma_0 . \exp\left(-\frac{t}{\tau} . \log(\sigma_0)\right) \quad (2\text{–}12)$$

where σ_0 is the initial width (Zhang, Chow & Wu 2016).

2.12 DANGEROUS DRIVING BEHAVIOUR

The behaviour of a driver is described by two fundamental parameters which are velocity and acceleration. These variables are closely interrelated and the permitted value for acceleration depends on the instantaneous velocity. Eboli, Mazzulla & Pungillo (2016) proposed a methodology to detect whether a particular car drivers' behaviour is safe or not. Each vehicle has an acceleration vector in lateral and longitudinal directions and the value of the acceleration norm is calculated by the following equation:

$$|\bar{a}| = \sqrt{a_x^2 + a_y^2} \quad (2\text{–}13)$$

where a_x is acceleration over x-axis and a_y is acceleration over y-axis.

The second law of Newton is proved that the Fs stimulus force is equal to the mass of the vehicle and the value of acceleration according to the following equation:

$$F_s = m . |\bar{a}| \quad (2\text{–}14)$$

where F_s is stimulus force defined by newton and m mass amount of object per KG.

The second law of Newton proves that the *Fs* stimulus force is equal to the mass of the vehicle and the value of acceleration according to the following equation:

$$F_R = m . g . \mu \quad (2\text{–}15)$$

where m mass amount of object per KG, g is g-force acceleration due to the gravity, and μ is the coefficient of side friction.

In this way, without considering superelevation, and when the vehicle is in the equilibrium mode, the *Fs* is equal to F_R and based on the value of F_R and *Fs*, we have three driving conditions:

1) The driving condition is safe when *Fs<FR*

2) The driving condition is safe when *Fs = Fr*

3) The driving condition is unsafe when *Fs>Fr*

To find a threshold value for acceleration to find the safe limit for velocity and acceleration, we should start working on the *Fr = Fs.* $F_s = F_r \rightarrow m.|\bar{a}| = m.g.\mu \rightarrow |\bar{a}| = g.\mu$, which can be written:

$$\sqrt{a_x^2 + a_y^2} = g.\mu \qquad (2\text{-}16)$$

By squaring both members:

$$a_x^2 + a_y^2 = (g.\mu)^2 \qquad (2\text{-}17)$$

According to the previous equation, acceleration level is related to the side friction between the road surface and tyre and the coefficient value depends on speed and meteoroidal condition. There are two kinds of side frictions: longitudinal side friction (μx), in the same direction of the motion, and lateral side friction (μx), perpendicular to the direction of the motion. According to Lamm, Psarianos & Mailaender (1999), the maximum value of friction over a longitudinal direction for a rural road is equal to:

$$\mu_{xmax} = 0.214.\left(V/_{100}\right)^2 - 0.640.\left(V/_{100}\right) + 0.615$$

$$\mu_y = 0.925.\mu_x \qquad (2\text{-}18)$$

$$\mu_{ymax} = 0.198.\left(V/_{100}\right)^2 - 0.592.\left(V/_{100}\right) + 0.569$$

The following ellipse equation is valid which is well-known as the 'ellipse of adherence:

$$\left(\frac{\mu_y}{\mu_{ymax}}\right)^2 + \left(\frac{\mu_x}{\mu_{xmax}}\right)^2 \leq 1 \qquad (2\text{-}19)$$

Figure 2-2 illustrates this equation in a visual form and additional information is as follows:

- μ_y and μ_x represent the components of side friction over the x and y axis
- μ_{xmax} is the maximum friction factor in the longitudinal direction
- μ_{ymax} is the maximum friction factor in the lateral direction

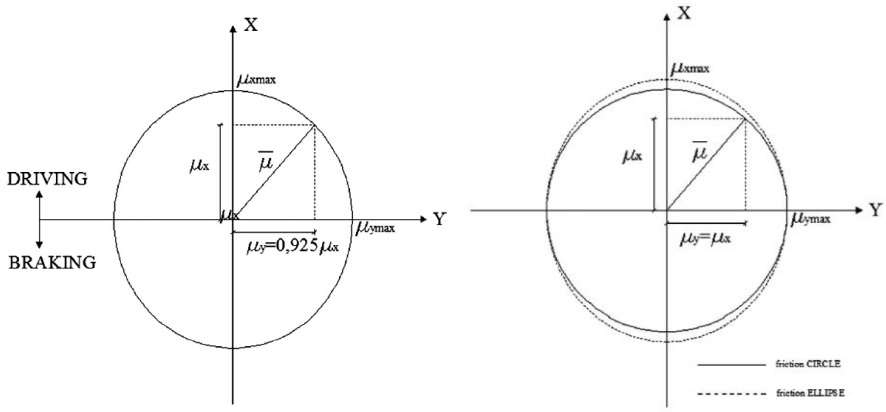

Figure 2-2 -Visualizing μ_{max} Calculation

Because of the importance of $\mu(\mu max)$ in this thesis, this value is replaced in the corresponding equation to calculate the limit value for acceleration:

$$a_x^2 + a_y^2 = (g.\mu_{max})^2 \qquad (2\text{-}20)$$

Because we want to calculate the limit value for acceleration, we consider the ellipse adherence to circle according to the assumption that $\mu x = \mu y$ (Figure 2-2). Hence, we have two modules which are the same for global side friction in two different directions. According to this assumption, all points of the border are the same distance from the centre.

By considering the proposed assumption, we have the following equations:

$$\sqrt{a_{lat}^2 + a_{long}^2} = g.\left(0.198.\left({}^{V}/_{100}\right)^2 - 0.592.\left({}^{V}/_{100}\right) + 0.569\right) \qquad (2\text{-}21)$$

Or

$$|\bar{a}| = g.\left[0.198.\left({}^{V}/_{100}\right)^2 - 0.592.\left({}^{V}/_{100}\right) + 0.569\right] \qquad (2\text{-}22)$$

For example, according to Figure. 2-3, the safe driving area for acceleration when the speed is close to zero is ±6 m/s^2 and when the norm of this value is larger than this value, the driver behaves dangerously.

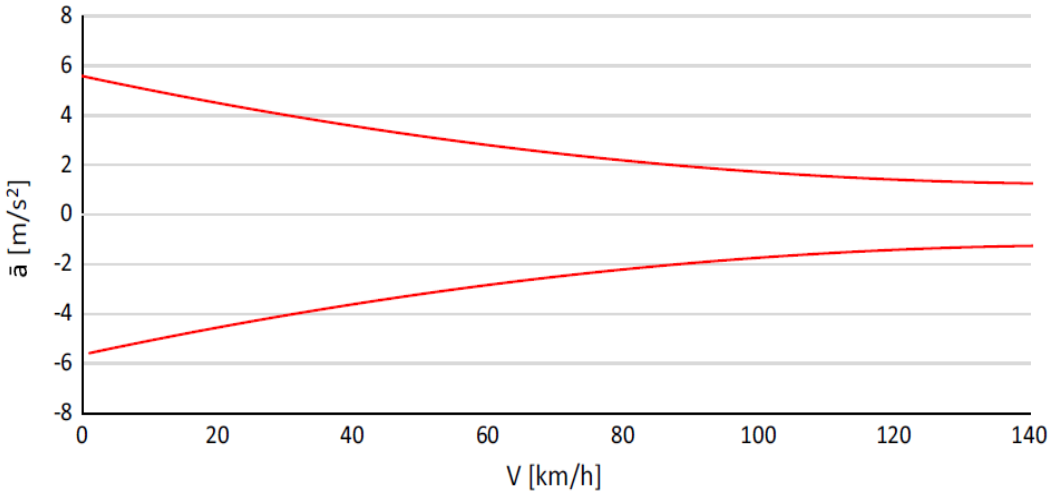

Figure 2-3- Safe Driving Area

2.13 SUMMARY

Mobile telematics is a cheap alternative to in-vehicle data recorders and has various applications in industry and academia. Driving behaviour is complex, nuanced, and dynamic, and understanding driving behaviour using synthetic data which relies on a good definition does not adequately measure the behaviour of drivers in the real-world. In addition, the lack of labeled data is a challenge, as highlighted in (Nguyen et al. 2019). To the best of our knowledge, much of the research until now has been conducted on data gathered from either questionnaires, site investigations, or laboratory simulations. We believe that the dynamic properties of human behaviour cannot be fully reflected in simulated data. Moreover, our literature review shows that driving style pattern recognition using mobile telematics data has not been studied in any great detail.

In addition, to the best of our knowledge, the application of unsupervised learning algorithms in data-driven decision support systems is still questionable, and few studies exist on the application of fuzzy logic to reduce the uncertain situation of big data in this domain. Therefore, in this study, we proposed a big data-driven decision support framework for mobile telematics environment to assess the risk of drivers and cover the aforementioned gaps in this domain.

Chapter 3:

BIG DATA-DRIVEN DECISION SUPPORT SYSTEM FRAMEWORK

3.1 INTRODUCTION

Decision support systems are computerized systems that help decision makers find the best option from a range of alternatives by learning and analyzing historical and current data (Lu et al. 2019). Multiple types of decision support systems (DSS) such as model-driven, data-driven, and knowledge-driven systems have been widely applied in different domains and they have become more prevalent in recent years (Power & Sharda 2007). Model-driven DSSs are complex decision systems, which help decision makers arrive at a decision from a range of alternatives and a set of options. Data-driven decision support systems use information extracted from databases and data warehouses to find the answers to business questions. The knowledge-driven decision support system is the third type of decision support systems. This type of DSS builds knowledge from data to support managers for decision making.

Data-driven decision support systems provide managerial insights for decision makers in various applications such as risk assessment. Risk refers to "the possibility of something bad happening", and risk assessment refers to "the process of examining the risks involved in a

planned activity" (Audi 1999). Risk assessment involves a broad range of activities that assess the probability and severity of an accident in the future. This procedure either qualitatively or quantitatively evaluates the risk level of different activities that may cause dangerous or hazardous situations. A risk assessment process helps decision makers to understand the exposure associated with particular activities and prioritizes them according to risk level. Different quantitative and qualitative methods have been proposed for risk assessment. Quantitative methods aim to provide a numeric score that estimates the risk level of incidents, while qualitative methods evaluate the risk of events based on some qualitative measures or expert opinions (Sengupta et al. 2016).

Modeling a DSS is very costly and time-consuming because of behavioural research issues and data collection (Power & Sharda 2007). Traditional DSSs are developed based on the information collected by decision makers using surveys, discussions and brainstorming (Zhou et al. 2020), while in recent years, the data collection procedure has changed significantly. Technological improvements in database engineering, information technology, and the Internet of Things (IoT) has increased the volume, variety, and velocity of data (Ghasemaghaei & Calic 2019), and recent advancements in artificial intelligence and advanced analytical techniques provide new opportunities for decision-makers to create unimaginable value from big data for decision making (Chen & Zhang 2014). In many real-world situations, the risk of events is assessed based on the likelihood and severity of each event. Over the years, many qualitative and quantitative risk assessment techniques have been developed which are used in different situations and there is no universal technique. The choice of techniques mainly depends on the objective, availability of data, life-cycle stage, and available resources. In era of big data, data-driven risk assessment is a must.

These rapid developments in data technology, particularly in mobile telematics, motivate us to explore the possibility of proposing a big data-driven decision support system framework for mobile telematics. The main purpose of this study is to use driving behaviours to provide a decision support system for risk assessment and missing data imputation.

This chapter details the proposed framework and elaborates its components. In this chapter, the proposed DSS and its components are proposed. This DSS uses the data collected by mobile telematics devices to understand driving behaviour using the capability of machine learning and advanced analytical techniques.

3.2 Decision Support System Framework

Wahlström, Skog & Händel (2017) proposed an architecture for mobile telematics to show the internal information flow in smartphone-based vehicle telematics. Figure 3-1 illustrates this architecture. The users drive a car or change the position of their smartphones while they are driving. The smartphone collects behavioural information using its internal computing sensors such as the accelerometer, gyroscope, etc. The collected data is transferred to a server to store all the data collected by the smartphones in mobile telematics data storage. The stored big data can be applied on various applications and useful business models can be developed on a mobile telematics infrastructure.

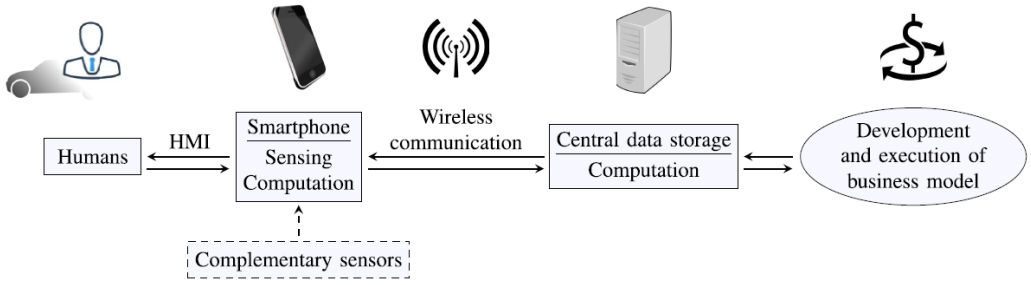

Figure 3-1- Smartphone-Based Vehicle Telematics (Wahlström, Skog & Händel 2017)

The DSS is based on this idea to develop an up-to-the-minute big data driven decision support system on mobile telematics, which is the cheaper, easier alternative of telematics, using machine learning, advanced analytics and fuzzy logic. Figure 3-2 illustrates the model and its related components. The main goal of this system is to use driving behaviour data to provide insights to help decision makers in relation to risk assessment and reducing null data points. The system uses advanced analytical techniques and machine learning algorithms, and

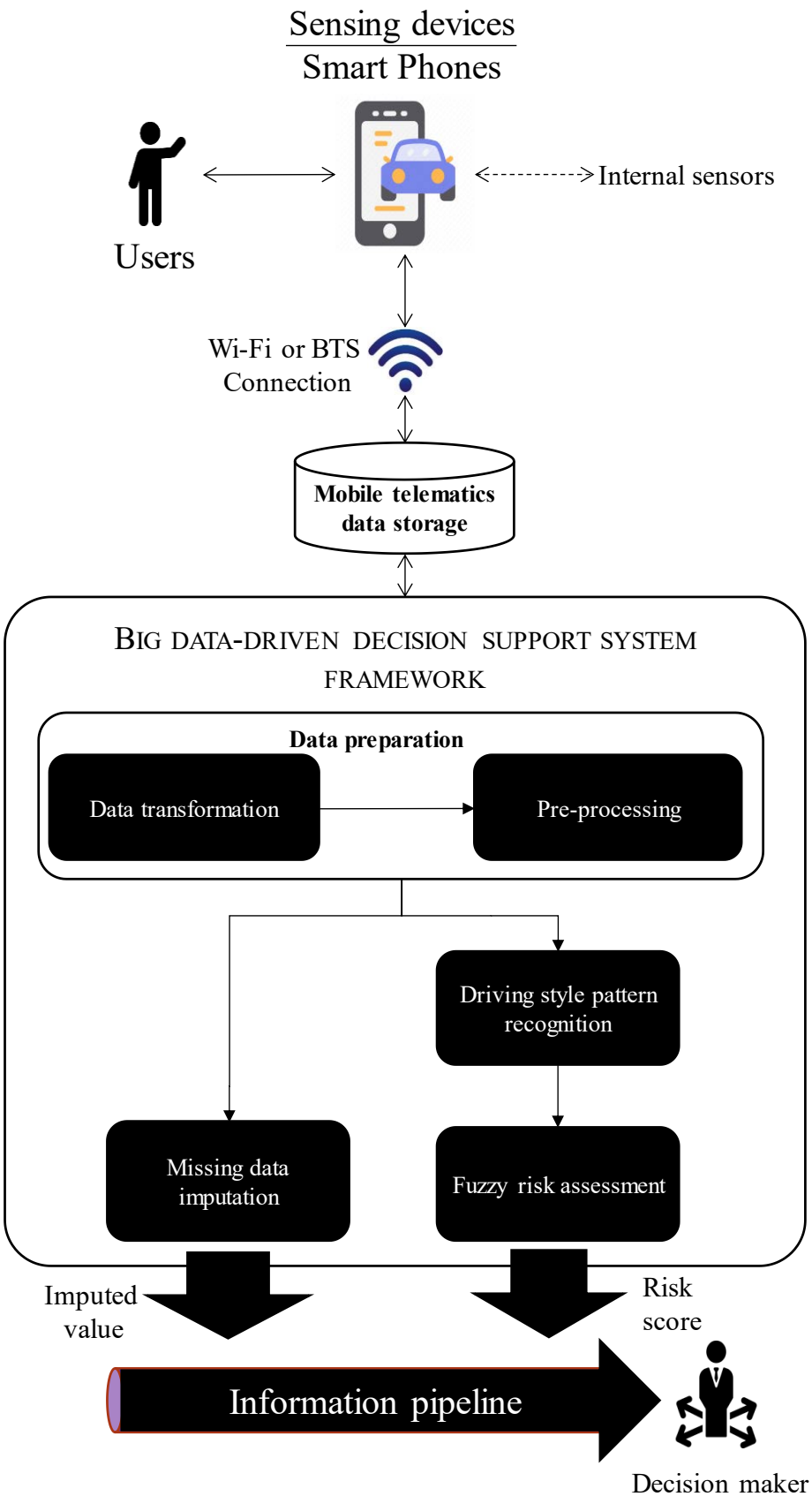

Figure 3-2- Mobile Telematics Big data Decision Support System

recognition, fuzzy risk assessment, and missing data imputation. Each component directly or indirectly supports decision makers through the information provided by the information pipeline.

3.2.1 DATA PREPARATION

Data preparation is an essential step in any data mining and knowledge discovery project. Hence, the primary goal of this component is to clean the data, reduce its complexity, and prepare them for analytics. Different industries face various challenges when applying big data to decision making, so understanding big data and its challenges for analytics is critical. Firstly, big data and its characteristics should be defined clearly to use it effectively. There are five dimensions of big data (Gandomi & Haider 2015) which are described as follows:

Volume: volume refers to the magnitude of data and the size of the data stored in the databases. The volume of stored data is reported as multiple terabytes and petabytes. The definition of big data based on volume has changed over time due to increased storage capabilities. Katal, Wazid & Goudar (2013) state that the word 'big' in the term big data defines volume. They also state that the volume of data will increase to zettabytes in the near future and the role of social networks in generating this huge amount of data is undeniable. Websites such as Facebook, Twitter, YouTube etc. connect different people to each other and also capture the daily interaction of people as digital information (Tan et al. 2013).

Variety: Variety refers to the structural heterogeneity in a dataset. Data can be structured, semi-structured or unstructured. Innovative methods need to be used to extract useful knowledge from these data. For example, clickstream data provides new opportunities for companies to extract knowledge from new datasets and apply this knowledge to cross-selling and up-selling marketing (Gandomi & Haider 2015).

Velocity: refers to the rate at which data are generated and the speed at which it should be analyzed and acted upon The increased number of digital devices such as smartphones and sensors has increased the rate of data generation (Chi et al. 2016). For example, Walmart processes more than one billion transactions per day. The data generated by technological

devices such as mobile apps, sensors, and wearable devices provide helpful information on customers, their personal preferences and buying habits. The role of agility in decision making is very important and various tools and applications can be used to help managers make proper decisions in a timely manner (Hofmann 2017).

Value: refers to the economic value of various data which varies significantly for organizations. Companies should focus on the value they will gain from data before investing in big data. Kaisler et al. (2013) expressed the usefulness of data for decision making and the importance of extracting knowledge from data for analysis.

Veracity: the quality of data is another concern in big data analytics. Organisations should have adequate knowledge of the quality and accuracy of data. The veracity or quality assurance of data has been investigated by practitioners and researchers in healthcare. Data veracity is critical in the area of healthcare because a decision which is based on incorrect data could endanger the health of patients, for example, doctors' poor handwriting is the most well-known form of inaccurate data.

Mobile telematics data collected by smartphones creates very large databases with high volume data, time series and unstructured trajectory data. These data are generated by devices in a short time and increase in volume in real time. These data can provide huge value for businesses but their complexity should be decreased using data pre-processing and data cleansing tasks.

Smartphones record the position of a vehicle as geolocation coordinates, e.g., latitude and longitude on a map. This component transforms these unstructured data to time-series data such as speed and acceleration.

Once the trajectory data is transformed to a new format, which shows the driving characteristics each second, a data pre-processing method is developed to decrease the complexity of data and change the data structure. Data pre-processing includes two change detection and feature extraction techniques, as explained in Chapter 4. The output of this

component is fed into the missing data imputation and driving style pattern recognition components.

3.2.2 DRIVING STYLE PATTERN RECOGNITION

Analysing driving behaviour using unsupervised learning and pattern recognition algorithms is a challenging task. An empirical analysis should be undertaken to compare the performance of different unsupervised algorithms. The driving style pattern recognition component discovers unique driving patterns from mobile telematics big data. This component is developed based on the capabilities of unsupervised learning algorithms. The proposed component has three main parts: a self-organizing map, a nine-layer deep auto-encoder, and partitive clustering algorithms. The SOM algorithm reduces the complexity of the data, the deep auto-encoder extracts the features, and the clustering algorithm groups driving events with similar patterns into behaviours. Further, given that clustering with mobile telematics data is an under-researched area, an empirical comparison of five well-known clustering algorithms has been undertaken to determine the strengths and weaknesses of each method and which is best suited to categorizing driving styles. The results of this component provide a basis for feeding to the next fuzzy risk assessment component.

3.2.3 FUZZY RISK ASSESSMENT

Generally, risk assessment techniques are proposed based on qualitative measures according to subject matter experts' opinion. Fuzzy set theory introduced by Zadeh (1965 helped decision makers use fuzzy logic to simulate uncertain situations in the human brain through a membership function. The proposed component uses a pattern recognition algorithm to extract driving patterns from smartphone-generated data, which are mostly unlabelled. The model learns from mobile telematics big data and extracts unique driving patterns from the huge amount of driving streams to extract unique driving categories. Each

driving category has a different risk level, thus it is necessary to propose a risk assessment methodology to estimate their risk score.

Therefore, this component has two main steps: first, it learns innovatively from big data using the capabilities of unsupervised learning algorithms and fuzzy clustering. Second, a fuzzy risk assessment model evaluates the risk of previous events and calculates the risk of new events according to their similarity to previously assessed events. The fuzzy membership functions are used in this study to simulate uncertain situations in real-world risk assessment problems and the big data environment. The final result of the DSS is a risk score which is calculated using the fuzzy inference system and unsupervised learning. The score can be used by the final decision maker for risk evaluation.

3.2.4 MISSING DATA IMPUTATION

Mobile telematics devices usually collect trajectory data that show driving behaviour. These devices are not able to collect demographic features of drivers such as age, gender, home location, etc. These features play an important role in risk assessment and many insurance companies rely upon them. For example, the insurance premium for young drivers is much higher than older drivers, in addition, gender could impact the cost of insurance for policyholders.

The main purpose of the missing data imputation component is to improve data quality in mobile telematics data. This is a supportive component to help decision makers increase their insight into declarative features. The component contains a novel supervised learning algorithm, i.e. a new a Choquet fuzzy integral vertical bagging classifier, which learns from driving characteristics to estimate the missing fields.

Demographic data gap in mobile telematics is very big and missing, and missing data imputation component helps us to reduce this gap using a provided supervised learning algorithm. We used the hidden knowledge of mobile telematics for based on the available knowledge to impute the value of missing data and features. For example, in this thesis we

detected gender of drivers based on driving behaviour. This use case is only one of the applications of the proposed algorithm.

3.2.5 INFORMATION PIPELINE

Mobile telematics has so far been used in a number of road safety applications (Zhao 2002), intelligent transportation systems (Zhao 2000), and usage-based insurance (Bowne et al. 2013), but applying this technology in real-world businesses is problematic. According to the usage-based insurance industry experts' opinion, driver risk is only one variable to consider in relation to premium calculation and providing a risk score from driving style is not enough to make a comprehensive decision based on mobile telematics, thus the risk score should be merged with other variables to provide useful managerial insights.

Figure 3-3 illustrates two major outcomes provided in this study, the risk score and the imputed value. The risk score is calculated by a fuzzy risk assessment model and the imputed value is provided by missing data imputation. For example, new customers use their smartphone while driving their car, the mobile telematics devices collect their information and submit this to the insurance company data warehouse, the insurance company uses this

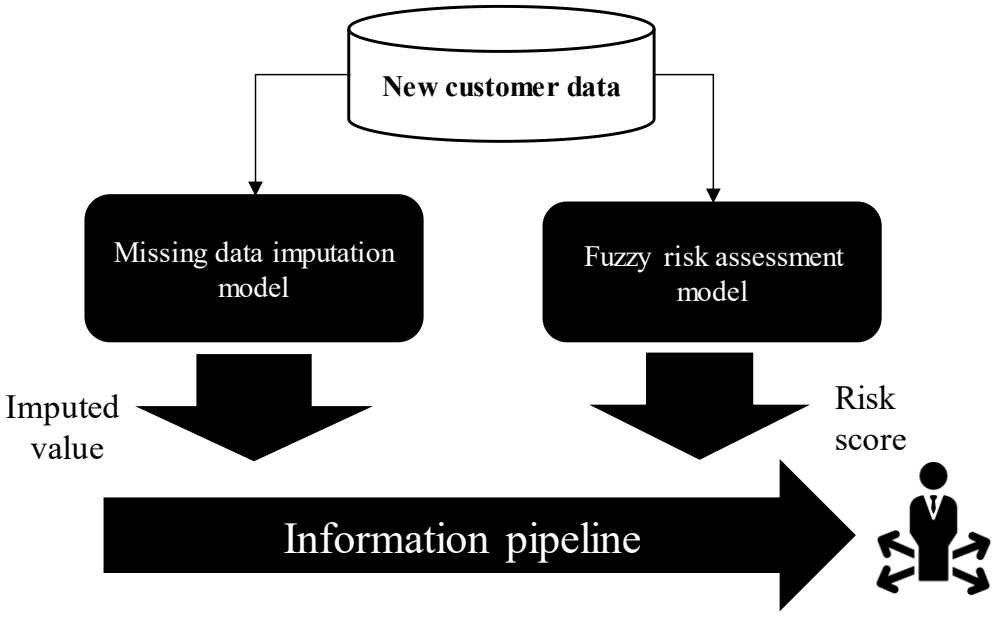

Figure 3-3- Information Pipeline

data to assess their risk, but the insurance premium calculation depends on other variables such as gender, age range, day/night parking location, living and working suburbs etc. Most of this declarative information is not provided by new customers, thus the missing data imputation model uses the collected data to provide an estimation about the missing fields.

In this study, the missing data imputation component is applied to detect the gender of the driver from the driving data using the Choquet fuzzy integral vertical bagging classifier. This component can be applied for other declarative variables such as the suburb in which they live, age range etc. Figure 3-4 provides an example of the information pipeline cross

Risk score	MD_1	MD_2	...	MD_i
RS_1	% IV_{11}	% IV_{21}	...	% IV_{i1}
RS_2	% IV_{12}	% IV_{22}	...	% IV_{i2}
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
RS_k	% IV_{1k}	% IV_{2k}	...	% IV_{ik}

Figure 3-4- Information Pipeline Cross Table.

The example shows the information pipeline capability to reduce the data gap in the mobile telematics domain. RS is the risk score calculated by the fuzzy risk assessment component, k is the total number of drivers, MD is the missing data domain, i is the total number of domains in which missing data imputation is used. The missing data imputation component can be applied on various missing data domains such as gender detection, which is the only domain that is proposed in this study. The imputed value (IV) is an estimation of missing data and can be merged by other variables and scores. The final cross-table can be applied for decision making.

table. The information pipeline merges the capabilities of fuzzy risk assessment and missing data imputation to provide business value for decision makers.

3.3 SUMMARY

This chapter proposed a framework for the decision support system which is used for risk assessment and missing data imputation using mobile telematics data. The framework consists of four components, namely data preparation, driving style pattern recognition, fuzzy risk assessment, and missing data imputation.

The data preparation component prepares data for analytics by proposing change data transformation and data pre-processing techniques. The driving style pattern recognition component, which evaluates the performance of various unsupervised learning algorithm for driving style pattern recognition. The fuzzy risk assessment component evaluates the risk level of driving behaviour using fuzzy logic in an uncertain situation. Missing data imputation improves the quality of data by proposing a novel supervised learning algorithm, and finally the information pipeline merges all the information together and provides a comprehensive report on driver risk and the imputed variables.

Chapter 4:

DATA PREPARATION

4.1 INTRODUCTION

Data preparation is the first step for any data-related project as data quality is critical to the final result. Therefore, various data manipulation techniques such as data transformation, feature extraction and abrupt change detection were undertaken to improve data quality.

This chapter introduces the proposed data preparation component as the first component. Figure 4-1 shows the proposed component and its related items.

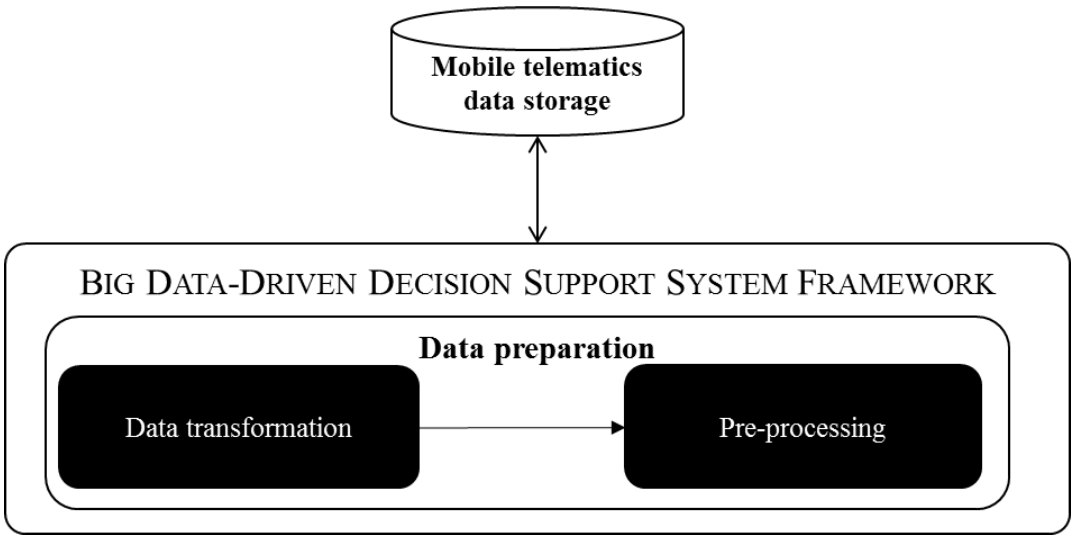

Figure 4-1- Data Preparation Component

4.2 MOBILE TELEMATICS DATA

Mobile telematics devices collect human behaviour using the mobile's GPS and its internal devices, thus mobile telematics data principles need to be explained. The following preliminary definitions explain the mobile telematics data principles.

Definition 4–1 (Dong et al. 2016). The position of a vehicle in a 2D coordinate space at time t is Pt. A driver starts each trip at time 0 from location $P_0 = [0, 0]$

Definition 4–2 (Zhou et al. 2016). Mobile telematics devices generate a series of GPS data for each trip (tr)

$$tr = P_0 \rightarrow P_1 \ldots \rightarrow P_i \ldots \rightarrow P_i$$

The starting point of a trip is P_0 and the end point is P_i. The trips associated with each driver occur at different times, day or night.

Figure 4-2 shows one short trip with length three. The car's position is recorded each second using mobile telematics devices and stored by a remote server.

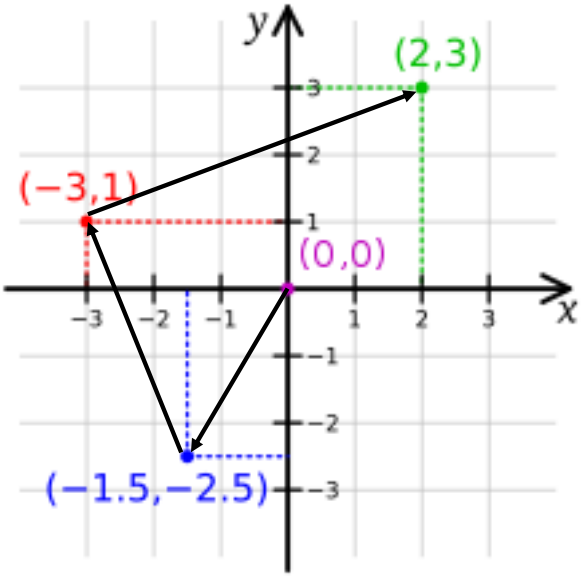

Figure 4-2- A Sample Driver's Trips

The example shows that the car started the trip at P_0 = (0,0), then moved to the second and third positions which are P_1 = (−1.5,−2.5) and P_2 = (−3,1) respectively, and finally the trip finished at P_3 = (2,3). The mobile telematics devices transfer these data to a remote server and driving characteristics such as velocity and acceleration are extracted from them.

ISO 8855 ((ISO) 2011-12) explains the technical definitions of road vehicle dynamics to provide a common language for designing and modelling moving objects. This definition is used to define objects' movements into one, two, or three dimensions. In this study, a two-dimensional coordinate system is used to model mobile telematics movements.

In line with international standard ISO 8855 , a two-dimensional coordinate system is considered to calculate driving characteristics. The coordinate system is depicted in Figure 4-3. The forward and backward directional movements of the car are plotted on the x-axis, and the left and right directional movements of the car are plotted on the y-axis. These assumptions are used to calculate the value of instantaneous velocity and acceleration. Therefore, changing the position of the vehicle in a forward or backward direction indicates x-axis movement and movement in a left or right direction indicates y-axis movement.

4.3 DATA TRANSFORMATION

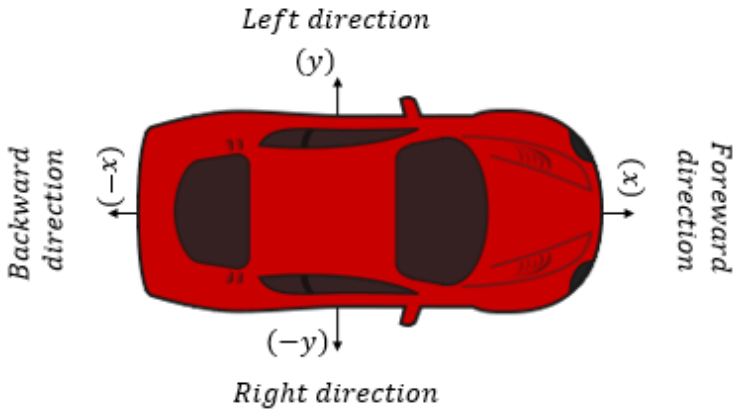

Figure 4-3- Two-Dimensional Vehicle Coordinate System (ISO 8855) ((ISO) 2011–12)

In general, the collected mobile telematics data is a group of trajectory points. These data are not understandable to supervised and unsupervised learning algorithms, so various data manipulation techniques need to be developed to transform the data to a new format which is usable for analytical tasks.

The main purpose of this part is to transform trajectory data into a new format. The new format is a time-series stream that shows the value of driving characteristics at time t.

Therefore, the following definitions explain how data transformation techniques help to transform telematics data into driving characteristic streams.

Definition 4–4. Let $|\overline{V_t}|$ be the instantaneous velocity of the vehicle at time t, calculated by:

$$|\overline{V_t}| = \sqrt{\left(v_{x_t}\right)^2 + \left(v_{y_t}\right)^2} \qquad (4\text{–}1)$$

where v_{x_t} and v_{y_t} show the instantaneous velocity of the vehicle at time t over the x and y axes. These are calculated by $v_{x_t} = \frac{\Delta x}{\Delta t}$ and $v_{y_t} = \frac{\Delta y}{\Delta t}$, respectively.

Definition 4–5. $|\overline{A_t}|$ is the norm of an acceleration/deceleration event at time t, calculated by:

$$|\overline{A_t}| = \sqrt{\left(a_{x_t}\right)^2 + \left(a_{y_t}\right)^2} \qquad (4\text{–}2)$$

where a_{x_t} is the value of the instantaneous acceleration at time t over the x-axis, which is equal to $\frac{\Delta v_x}{\Delta t}$. Similarly, the instantaneous acceleration over the y-axis is $a_{y_t} = \frac{\Delta v_y}{\Delta t}$.

Definition 4–6. ω_x is the value of the vehicle's angular speed around the x, y, or z-axis at time t, calculated by

$$\omega = \frac{d\theta}{dt} \qquad (4\text{–}3)$$

where $d\theta$ is the value of the angular displacement at time t. The name for this measure over the z-axis is the yaw rate, over the y-axis is the pitch rate, and over the x-axis is the roll rate.

4.4 DATA PRE–PROCESSING

In this study, two different data pre-processing techniques are used to improve data quality. Firstly, a change detection algorithm is used to detect the most important driving

events. Secondly, a feature extraction algorithm is used to extract useful features from unstructured driving streams.

4.4.1 CHANGE DETECTION

The data streams generated from mobile telematics do not arrive ready for analysis as smartphones record data without any knowledge of the mechanical features of the vehicle (Foresti, Farinosi & Vernier 2015). For example, if a driver stops for a long time, these data are recorded, even though they are useless for our purposes. So, to ensure these types of data do not decrease the performance of the model, they must be identified and removed.

The implemented version of the change detection algorithm developed by Liu et al. (2013) is used in this study. Driving characteristics are the input variables for this algorithm and the change score is the output.

The change detection algorithm is formulated by considering $V(t)$, $A_x(t)$, and $A_y(t)$ as the three dimensions which are velocity $V(t)$, x-axis acceleration $A_x(t)$, and y-axis acceleration $A_y(t)$ respectively. Here, $V(t)$, $A_x(t)$, and $A_y(t)$ are three time windows with a length of k, which are:

$$V(t) = [V(t)^T,\ V(t+1)^T, \dots,\ V(t+k-1)^T]^T \in velocity;$$

$$A_x(t) = [A_x(t)^T,\ A_x(t+1)^T, \dots,\ A_x(t+k-1)^T]^T \in x-axis\ acceleration;$$

$$A_y(t) = [A_y(t)^T,\ A_y(t+1)^T, \dots, A_y(t+k-1)^T]^T \in y-axis\ acceleration,$$

where T is the transpose. Let $Y(t) = [V(t),\ A_x(t), A_y(t)]$ be the three-dimensional input data at time t, and $y(t)$ be a group of n retrospective subsequences of input data at time t, which is:

$$y(t) = [Y(t),\quad Y(t+1), \dots,\quad Y(t+n-1)]$$

$y(t)$ and $y(t+n)$ are treated as two consecutive segments of the data stream. Figure 4-4 illustrates an example of n retrospective consecutive segments in one-dimensional time-series data (Kawahara & Sugiyama 2012). The strategy is to calculate a dissimilarity score for these two segments using Eq. 4-1 as the measure of change. We selected the relative unconstrained

least-squares importance fitting (RuLSIF) algorithm as the change detection and scoring algorithm. RuLSIF is extremely good at detecting driving style changes (Lee & Jang 2017), human activity sensing (Liu et al. 2013), and smart home signal processing (Aminikhanghahi, Wang & Cook 2018) in data streams. RuLSIF calculates a change score using a density-based dissimilarity measure from two consecutive time window using Eq 4-4:

$$ChangeScore = D(p_t||p_{t+n}) + D(p_{t+n}||p_t) \quad (4\text{-}4)$$

where $D(p_t||p_{t+n})$ is a dissimilarity measure between p_t and p_{t+n}.

Liu et al. (2013) proposed "RuLSIF" to calculate the dissimilarity measure between two different time segments. It calculates the change between two consecutive time windows with a density-based dissimilarity measure:

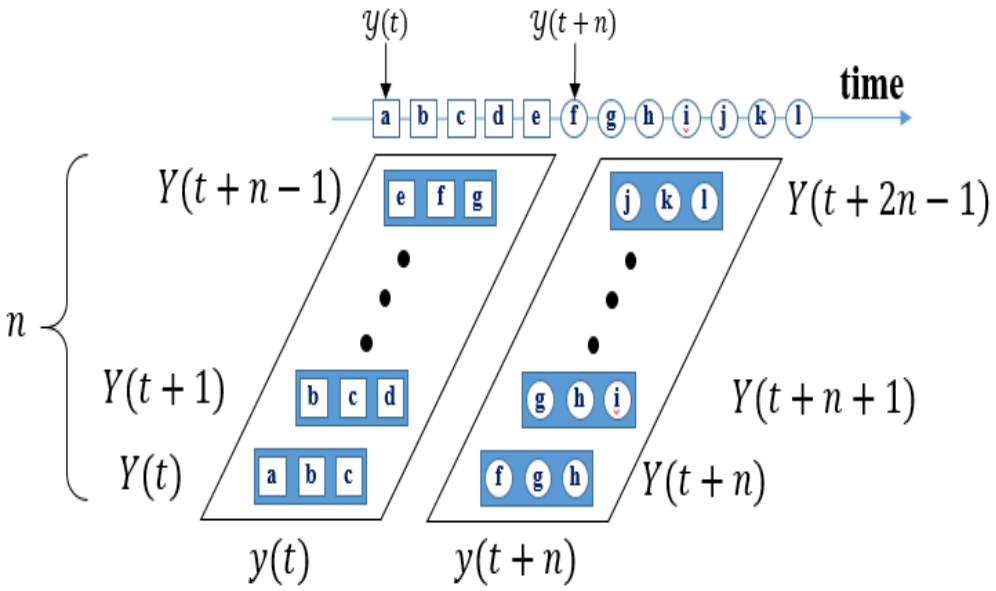

Figure 4-4 – Change Detection Sample

A sample of change detection with one-dimensional time-series data. $y(t)$ is the input data at time t, $Y(t)$ denotes k subsequences, and $y(t)$ is a group of retrospective subsequences. The value of n is equal to the window size, and k is the length of subsequence. In this example, k=3 is presented, but it is possible for this value to be larger.

$$D(p||p') = -\frac{\alpha}{2n}\sum_{i=1}^{n}\hat{g}(Y_i)^2 - \frac{1-\alpha}{2n}\sum_{i=1}^{n}\hat{g}(Y_i')^2 + \frac{1}{n}\sum_{i=1}^{n}\hat{g}(Y_i) - \frac{1}{2} \quad (4\text{-}5)$$

where n is the window size, and Y_i *and* Y_i' are two consecutive time windows in d-dimensional time-series data. $\hat{g}$ is the density-ratio estimation of the data samples, and α is a constant variable.

4.4.2 FEATURE EXTRACTION

Extracting useful features from big data streams is another technique that is used in this study. The data generated by smartphones are unstructured and are not comprehensible by machine learning algorithms. It is therefore necessary before starting analytics to transform these data into a new form by cleaning them, removing outliers, and extracting and selecting features.

Human activity recognition (HAR) is one of the applications of data collected from human behaviour. Researchers in this domain use the behavioural data collected by IoT devices and wearable sensors to recognize human activity. They used machine learning algorithms to detect human behaviour according to the signals provided by wearable sensors (Hassan et al. 2018; Lara & Labrador 2013). The data collected by wearable sensors is very similar to mobile telematics-generated data, thus the feature extraction technique provided by (Hassan et al. 2018; Lara & Labrador 2013) is used in this research.

The driving streams have been divided into time windows of lengths of 256 and sliding windows of 256, after which the statistical features from each time window are extracted. 14 statistical features including minimum, maximum, mean, median, first and third quantile, standard deviation, average absolute deviation, skewness, entropy, kurtosis, auto-correlation, zero crossing, and energy are used to extract meaningful features from big data stream, using the following equations:

1) Minimum: the smallest number for each time frame:

$$minimum = \min(window) \tag{4-6}$$

2) Maximum: the highest value for the selected time frame:

$$Maximum = \max(window) \quad (4\text{–}7)$$

3) Mean: the average value of all values in the selected time frame:

$$Mean = {}^{1}/_{n} \sum_{1}^{n} w_i \quad (4\text{–}8)$$

4) Median: the middle value for each time window, which is each equal to:

$$Median = \left\{\frac{n+1}{2}\right\}^{th} value\ of\ w \quad (4\text{–}9)$$

where n is the length of the sliding window.

5) First quantile: the top 25% value for each time window, which is each equal to:

$$Median = \left\{\frac{n+1}{4}\right\}^{th} value\ of \quad (4\text{–}10)$$

where n is the length of the sliding window.

6) Third quantile: the top 75% value for each time window, which is each equal to:

$$Median = \left\{3\left(\frac{n+1}{4}\right)\right\}^{th} value\ of \quad (4\text{–}11)$$

where n is the length of the sliding window.

7) Standard deviation: the following equation is used to calculate standard deviation:

$$Standard\ Deviation = \sqrt{{}^{1}/_{n} \sum_{1}^{n} (w_i - \overline{w})^2} \quad (4\text{–}12)$$

where w_i is the i[th] member in the time window, and $\overline{w}$ is the mean value of all members.

8) Mean absolute deviation: this measure for each time window is calculated by:

$$MAD = {}^{1}/_{n} \sum_{1}^{n} |w_i - \overline{w}| \quad (4\text{–}13)$$

9) The skewness which shows symmetry in data distribution.

$$Skewness = \frac{{}^{1}/_{n} \sum_{1}^{n} (w_i - \overline{w})^3}{sd^3} \quad (4\text{–}14)$$

10)**Entropy** level of each time frame is calculated by:

$$Entropy = 1/3 \sum_{1}^{n} c_i \, log(c_i)$$

$$c_i = \frac{w_i}{\sum_{1}^{n} w_j} \quad (4\text{–}15)$$

11)Kurtosis:

$$Skewness = \frac{{}^{1}/_{n} \sum_{1}^{n} (w_i - \overline{w})^4}{sd^4} \quad (4\text{–}16)$$

12)**Auto** correlation :

$$MAD = {}^{1}/_{n} \sum_{i=k+1}^{n} (w_i - \overline{w})(w_{i-k} - \overline{w}) \quad (4\text{–}17)$$

where k is lag and it is equal to one in this study.

13)**Zero** Crossing:

$$ZC(W) = \sum_{i=1}^{n-1} S(w_i, w_{i+1}) \quad (4\text{–}18)$$

$$S(x, y) = \begin{cases} 1, if\ (x.y) < 0 \\ 0, if\ (x.y) > 0 \end{cases}$$

14)Energy

$$Energy = {}^{1}/_{n} \sum_{1}^{n} w_i^2 \quad (4\text{–}19)$$

4.4.3 FEATURE SELECTION

After extracting useful features from driving streams and providing structured data from unstructured streams, the useful features are selected using feature selection and correlation analysis. Statistical correlation analysis is used in this study to find the critical features which have the highest chance of increasing the accuracy of the final prediction model using the selected features.

The correlation between two different variables such as X and Y can be calculated using linear correlation analysis. In this study, the correlation coefficients between the extracted features, mobile telematics stream and the output label, are calculated by (An et al. 2020):

$$r = \frac{COV(X,Y)}{\sigma^X \sigma^Y} \quad (4\text{–}20)$$

$$COV(X,Y) = {}^{1}/_{N} \sum_{i=1}^{N} (X - \bar{X})(Y - \bar{Y})$$

where $COV(X,Y)$ is the covariance between X and Y and σ^X and σ^Y are the standard deviation of variables X and Y respectively. $\bar{X}$ is the average value of X and $\bar{Y}$ is the average value of Y, and N is the total number of records.

According to the formula for linear correlation calculation, the correlation coefficient can have a positive or negative value. Positive r depicts a positive correlation between X and Y and a negative correlation is depicted by the negative correlation coefficients.

4.5 IMPLEMENTATION

This component is a supportive element for other components in this study. The implementation results of this component are thoroughly explained in the following chapters. The change detection algorithm is applied by driving style pattern recognition and fuzzy risk assessment components to prepare data for driving style risk assessment. The feature

extraction and feature selection techniques, which are explained above, are used in the missing data imputation component to reduce the complexity of mobile telematics big datasets. These two techniques are usually used for classification and supervised learning algorithms.

4.6 SUMMARY

Data preparation is the first component that prepares data for analytics. Mobile telematics data is big, unstructured, and noisy, thus various techniques need to be developed to improve the quality of data. Change detection, feature extraction, and feature selection are three techniques that are used in this chapter. The abrupt change detection algorithm is proposed to remove unnecessary time windows from mobile telematics data by selecting time frames with the highest change score. Feature extraction is used to create statistical meaningful features from unstructured data streams and feature selection is used to find features with the highest correlations among extracted features.

Chapter 5:

DRIVING STYLE PATTERN RECOGNITION

5.1 INTRODUCTION

Driving patterns are a useful source of knowledge for proposing a big-data-driven decision support system for risk assessment. The main purpose of this component is to develop an empirical analysis on mobile telematics data to find the best alternative for extracting driving patterns using unsupervised learning techniques and one the most state-of-the art machine learning techniques, such as deep auto-encoders. Deep learning is one of the current state of the art techniques which is proposed in this study. We evaluated the performance of deep auto-encoder in comparison to other unsupervised learning methods.

In order to propose this component, an unsupervised learning pattern recognition framework is proposed. The proposed framework has three phases. First, it decreases the complexity of the prepared data using a self-organizing map (SOM) and a deep auto-encoder. Second, an empirical study is undertaken on five of the most well-known and commonly-used partitive clustering algorithms in the field of pattern recognition to reveal the strengths and weaknesses of each, and to determine whether there is one best choice overall for categorizing driving styles from mobile telematics data, and also the optimal number of

clusters is found using a quantitative method. Finally, all the extracted clusters are investigated according to the transportation research to find a suitable name for each driving category.

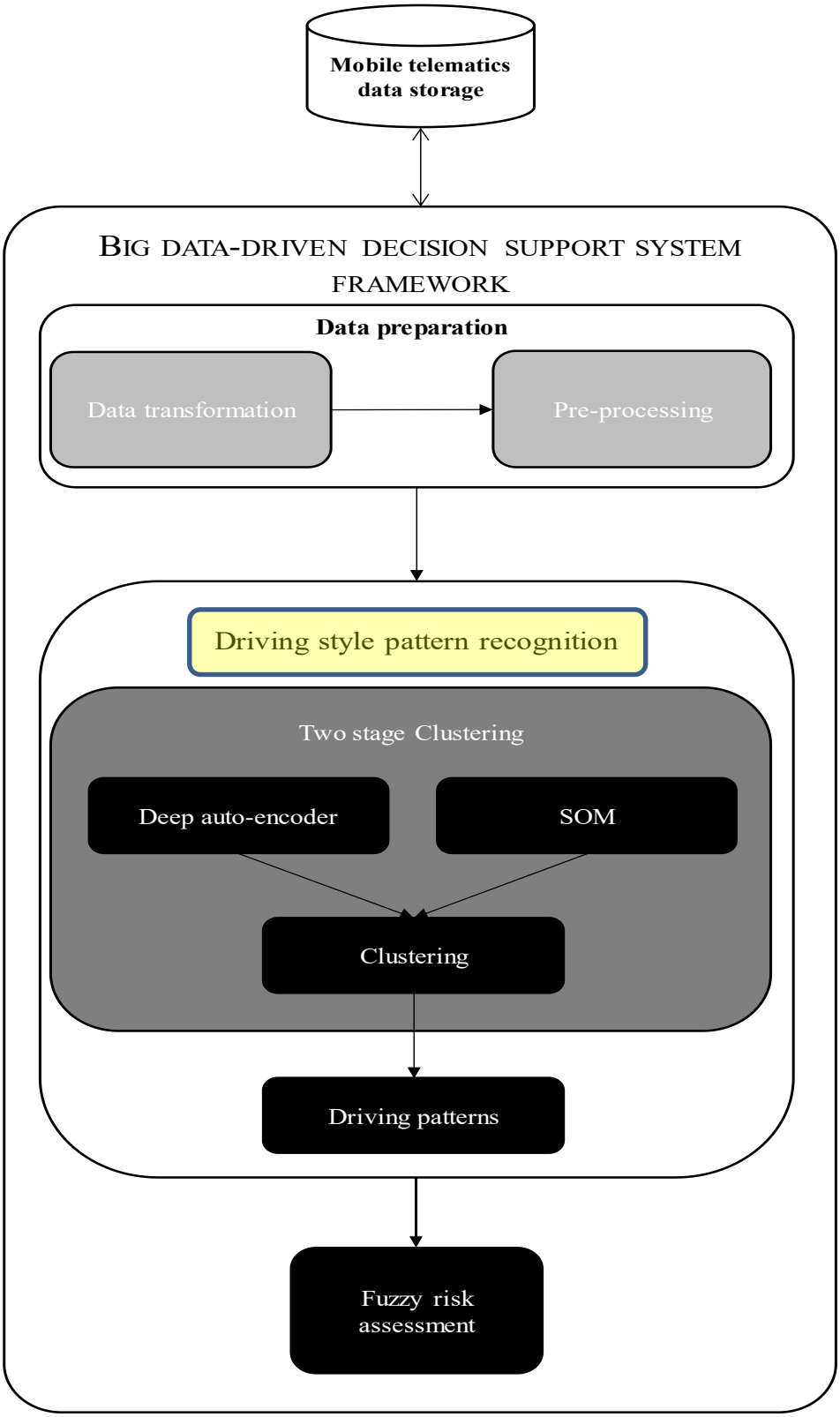

Figure 5-1- The Driving Style Pattern Detection Framework

Figure 5-1 illustrates the proposed driving style pattern recognition framework in this chapter. This component helps us to find the best option to cluster driving data. This component uses prepared data from the data preparation component, and then it evaluates the performance of various unsupervised learning algorithms for driving style pattern recognition.

The rest of this chapter is organized as follows. Sections 5.2 – 5.4 present the details of the proposed driving style pattern recognition component. Section 5.5 details the implementation and experiment results. Section 5.6 describes the various extracted driving patterns. Finally, Section 5.7 summarizes the chapter and describes the next step.

5.2 TWO-STAGE CLUSTERING

This component uses the prepared data from the data preparation component, which is explained in Chapter 4.

After preparing the data, a two-stage clustering algorithm categorizes the selected time windows into groups with similar characteristics. In this stage, a SOM and a deep auto-encoder are implemented to make a choice between them for clustering in the next step.

5.2.1 SOM

The SOM is a lattice output space with a rectangular topology. In SOM, the first step is to generate an initial SOM according to the number of input records. SOM iteratively maps the input records to the closest neuron in the hidden layers of the feature map. This neuron is known as the best matching unit (BMU). Then, the weight vector of each neuron is updated according to this change. The process is repeated until no remarkable change in the data is detected. The advantage of using the SOM algorithm for stream data clustering is that data generated from sensors is usually large in scale and noisy. A dimension reduction method decreases both the computational cost and the impact of noise (Vesanto & Alhoniemi 2000).

5.2.2 DEEP AUTO-ENCODER

The deep auto-encoder component consists of a number of neural networks with randomly generated weight and bias vectors, which are optimized during the training phase. Following Liu, Taniguchi, et al. (2017), we designed a deep network with five encoder layers. The number of nodes in each layer are: 45 → 22 → 11 → 5 → 3 → 5 → 11 → 22 → 45. Therefore, the network extracts the features using the encoder layers 45 → 22 → 11 → 5 → 3. The gradient descent optimizer is used to minimize reconstruction errors.

The driving characteristics are denoted as $X \in \mathbb{R}^{D \times T}$, where D is the dimension of the input data, and T is the length of the time window. For example, a driving event at time (t) is:

$$X_t = (V_1, \dots, V_T, Ax_1, \dots, Ax_T, Ay_1, \dots, Ay_T) \quad (5\text{-}1)$$

The activation function is a hyperbolic tangent so the value of the input variables should be in the range $(-1,1)$. Obviously, the raw velocity and acceleration values will not fall within this range, so the data needs to be normalized before running the deep auto-encoder. We performed minimum and maximum normalization to transform the value of each dimension into $(-1,1)$.

The input data for the first layer of the deep auto-encoder is denoted as X_t, and the encoder function is

$$h_t^l = \tanh\left(\mathrm{W}_{en}^l X_t + b_{en}^l\right) \quad (5\text{-}2)$$

where W_{en}^l is a weight matrix for the encoder at the lth layer and b_{en}^l is the bias vector for the lth encoder layer.

The decoder function is a tanh function:

$$r_t^l = \tanh\left(\mathrm{W}_{de}^l h_t + b_{de}^l\right) \quad (5\text{-}3)$$

where W_{de}^l is the weight matrix and b_{de}^l is the bias vector.

An assumption during the encoder-decoder process is that the output value of r_t^l is equal to x_t^l. Thus, the objective function needs to calculate the reconstruction error between r_t^l and x_t^l:

$$O(V^l) = \frac{1}{N_V}\sum_{t=1}^{N_V} \| r_t^l - x_t^l \|_2^2 \quad (5\text{–}4)$$

where $\frac{1}{N_V}\sum_{t=1}^{N_V} \| r_t^l - x_t^l \|_2^2$ is the average value of the squared error between the reconstructed data and the input data.

A gradient descent optimizer with a learning rate of λ helps to minimize the reconstruction errors. Finding the best learning rate for a deep auto-encoder is typically very difficult and time-consuming. Hence, we opt for a linear grid search, where the first learning rate considered is λ^* and the search distance is θ, which is very small. This results in a learning rate of $\lambda^+ = \lambda^* + \theta^+$. θ is updated with

$$\theta^+ = \begin{cases} \theta & O(V^l) \geq O^+(V^l) \\ -0.5 \times \theta & O(V^l) < O^+(V^l) \end{cases} \quad (5\text{–}5)$$

The search stops when the change in the construction error between $O(V^l)$ and $O^+(V^l)$ falls below a set threshold.

The features extracted through this deep auto-encoder process are then carried forward for use in the partitive clustering step.

5.2.3 CLUSTERING

The SOM and deep auto-encoder algorithms have reduced the data to an abstract subspace. However, there will still be too many points to analyze directly, so they need to be clustered into similar groups. As mentioned in the introduction, no research has been undertaken to determine the best option for clustering unlabelled telematics data. Therefore, we conducted an empirical study on this issue with a range of different partitive clustering

algorithms to find the most suitable choice for clustering mobile telematics data and finding the optimal number of clusters in this domain.

5.3 IMPLEMENTATION

The proposed component is implemented in Python 2.7 on an Intel® Xeon® 3.01 GHz CPU, 64 GB of RAM, and running a Linux operating system. The software platform is Anaconda 2.7. The specific version of the SOM library by Saraee, Vahid Moosavi & Rezapour (2011) and the RuLSIF library proposed by Liu et al. (2013) are used for implementation. The Tensorflow is used for deep auto-encoder implementation.

A large-scale dataset collected by a European insurance company containing trip data for over 500,000 journeys from more than 2500 drivers is used in this study. The computational cost to process this entire dataset would be extremely high. But, according to Dong et al. (Dong et al. 2016), each person has their own driving patterns so no new useful information would be gained by analyzing more than a few trips per driver. We selected the 20 longest trips per driver to include in the analysis. Thus, the final dataset contained 50,000 journeys (20 trips × 2500 drivers). Table 5-1 provides brief details about the data used.

Table 5-1 - Selected Dataset

Trips	Drivers	Journeys per driver	Traveling time (minute)		
			Min	Average	Max
50,000	2,500	20	23:21	26:13	30:00

To extract the driving characteristics from the trajectory data, we split the data into three streams. The first stream is velocity, which is the speed of a vehicle during a trip at any given time. The second and third streams are acceleration over the x and y axes. These streams are used to assess hard braking, sharp starts (Lee & Jang 2017), and cornering behaviour (Fazeen et al. 2012; Handel et al. 2014).

To remove useless data, we divided the data into time windows of approximately 15 seconds with a slide in steps of 1 second because, according to Zhang, Zhao & Rong (2014), it takes at least 15 seconds to complete a single driving event. The RuLSIF-based change detection scores were then calculated for each time window. Figure 5-3 shows the input and output of the RuLSIF change detection algorithm with velocity, x-acceleration, and y-acceleration as the three input variables. Figure 5-3 shows the change score for the corresponding time frame. Approximately 7.9 million time windows were assessed. Following Lee & Jang (2017), we selected the 5% with the highest RuLSIF scores to represent the most significant changes, and further selected all windows with a change score greater than a threshold of 68.598. This left 394,833 windows, each representing one driving event with 15 seconds of data.

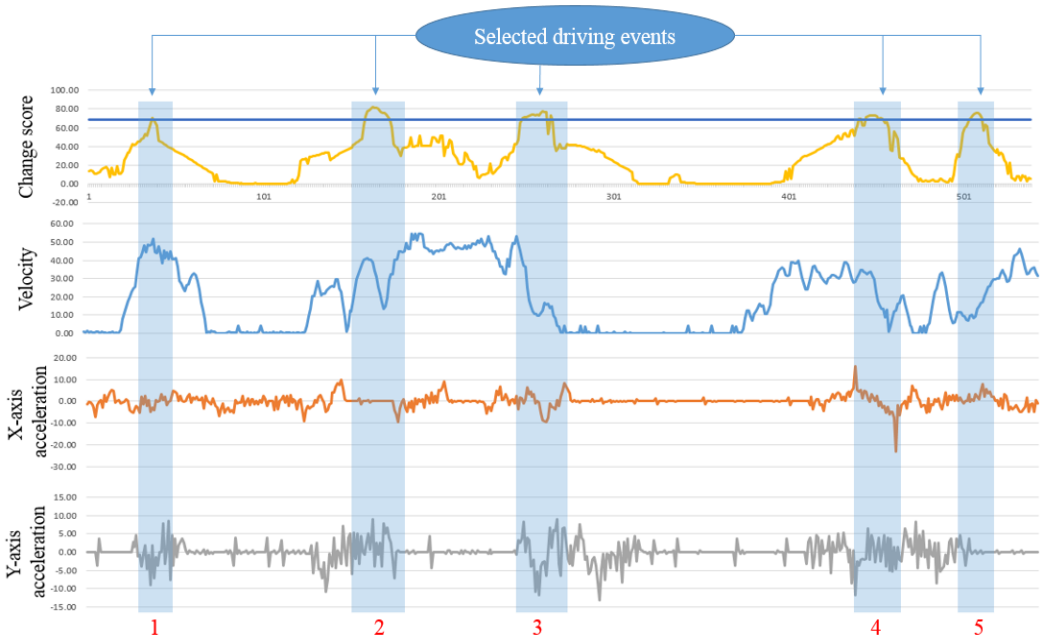

Figure 5-2 – Change Detection Scores

The five shaded blue columns indicate a detected driving behavior that exceeds the change threshold (in this case 68.60). All of these events have significant changes in driving behavior over velocity, x-acceleration, and/or y-acceleration. For example, the driver in Event 1 is driving at high and radically varying speeds, with many variations in y-axis acceleration. Event 4 displays high variations in all the variables.

5.3.1 TWO-STAGE CLUSTERING

After preparing the data and selecting the time windows with highest change score, using the data preparation component, the events were ready to analyze their driving characteristics. As aforementioned, we used SOM to reduce the complexity of data and a deep auto-encoder to extract the features. Figure 5-3 illustrates two stages in the two-stage clustering algorithm. Firstly, SOM defines a map with an appropriate number of nodes. The number of nodes is crucial because when n is small, the prototype is very generic and when n is very large, the prototype is very detailed. Therefore, to define an optimal number of nodes, we followed Céréghino & Park (2009) and identified a number of nodes equal to $5 \times \sqrt{n}$ where n is the total number of selected events. With 394,833 events, the optimal number of nodes was 2814. The next challenge was defining an appropriate map size for the input data. We selected a map size of 21×134 based on the eigenvalues and eigenvectors (Vesanto & Alhoniemi 2000).

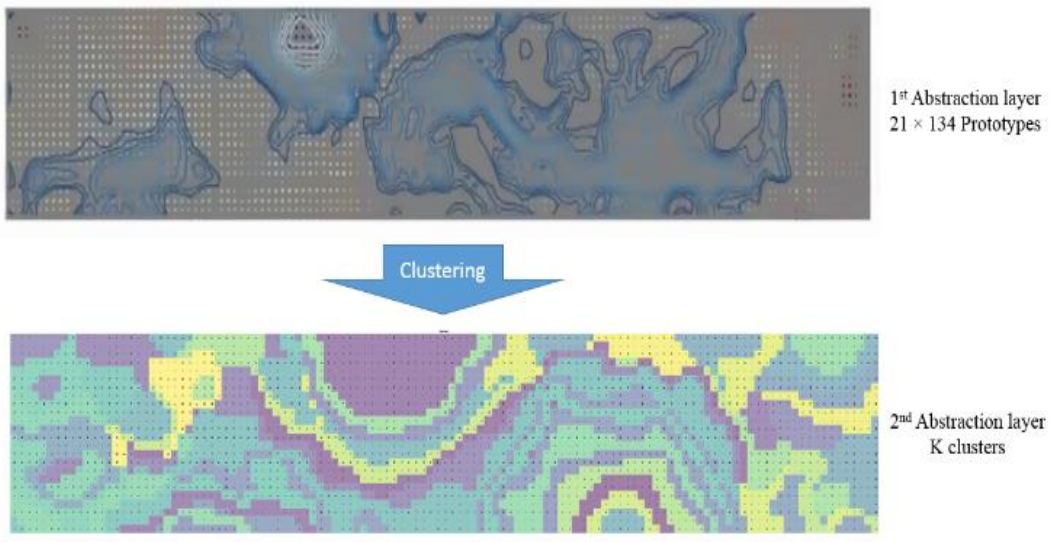

Figure 5-3 – Two-Stage Clustering

The second step in a two-stage clustering algorithm is partitive clustering. In this step, we used various partitive clustering algorithms to find the best clustering algorithm with the highest performance (Algorithm 5-1). A key concern in developing a partitive clustering algorithm is finding the optimal number of clusters. However, because the number of clusters is generally unknown in real-world problems, we developed Algorithm 5-2 to address this issue. In brief, the algorithm applies the sum of square error (SSE) and a bootstrapping technique to find a robust result.

5.3.2 PERFORMANCE EVALUATION

To determine the optimal clustering algorithm for the framework, and for mobile telematics data in general, we compared five different partitive clustering algorithms against three metrics with a five-fold cross-validation method and the performance validation algorithm. The details follow.

The five algorithms we chose for comparison were k-means (Arthur & Vassilvitskii 2006), MINIbatch k-means (Sculley 2010), agglomerative clustering (Kurita 1991) , spectral clustering (Von Luxburg 2007), and BIRCH clustering (Zhang, Ramakrishnan & Livny 1996). After preparing the data for clustering using the SOM and DAE, we used the test samples to compare the performance of the five models in terms of execution time, the Calinski-Harabasz and the Davis-Boulding indexes.

1) **Execution time** is the total time to determine the results – smaller values are better.

2) **Calinski-Harabasz (CH)** is a score calculated by assigning N data objects $X = \{x_1, x_2, \dots, x_n\}$ to K different clusters $C = \{c_1, c_2, \dots, c_n\}$ using the following equation (Caliński & Harabasz 1974):

$$\mathrm{CH(k)} = \frac{Tr(S_B)}{Tr(S_W)} \times \frac{N-k}{k-1}$$

$$Tr(S_B) = \sum_{i=1}^{K} N_i \|m_i - m\|^2 \quad (5\text{-}6)$$

$$Tr(S_w) = \sum_{i=1}^{K} \sum_{j=1}^{N_i} \|x_j - m_i\|^2$$

where k is the number of clusters, i is the number of items in a cluster ni, and mi is the centroid of cluster i. Tr(SB) shows the sum of between-cluster distances, and Tr(Sw) is the sum of within-cluster distances.

3) **Davies–Boulding (DB)** is another performance index that evaluates the clusters based on the sum of within-cluster scatters and between-cluster separations (Davies & Bouldin 1979):

$$\mathbf{DBI} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1, i\neq j}}^{\boldsymbol{n}} \boldsymbol{max}\left(\frac{\boldsymbol{\sigma_i + \sigma_j}}{\boldsymbol{d(c_i, c_j)}}\right) \quad (5\text{-}7)$$

where n is the number of clusters, σ_i is the average distance of all members of the i-th cluster to the center of the j-th cluster, and σ_j is the average distance of all members of cluster j to the center of the i-th cluster. d(ci,cj) is the distance between the center of the two clusters i and j (Halim et al. 2017; Maulik & Bandyopadhyay 2002).

Algorithm 5-1 - Performance Validation Algorithm

Input: selected events from change detection
Output: performance results

1. *X = data from change detection*
2. *X_SOM =training SOM model*
3. *X_DEA = training Deep Auto-encoder*
4. *Clustering_methods = [K-means, MinibatchKM, Spectral, Agglomerative, birch]*
5. *For fold in 5-fold cross validation for (X_SOM, X_DEA)*
   - *5.1. For clustering in Clustering methods*
     - *5.1.1. Finding optimal number of clusters for clustering by using training data with algorithm 5-2*
     - *5.1.2. CH[clustering , fold] = the value of Calinski Harabasz for current fold by using test data*

5.1.3.DB[clustering , fold] = average value of Davis Boulding for current fold by using test data
6. *CH[clustering] = average of CH value in all folds for all clustering_methods*
7. *BD[clustering] = average of BD value in all folds for all clustering_methods*

Algorithm 5-2 - Finding optimal number of clusters algorithm

Input: low complexity data, max number of clusters C, max iterations n
Output: optimal number of clusters

1. *X = data from SOM nodes*
   1.1. For k = 2 to C
   1.1.1.For i = 1 to n
   1.1.2.Di = Random under sampling for 80%
   1.1.3.Clustering Di into k clusters
   1.1.4.SSEi =Sum of Square errors
2. $SSE_k = {}^{1}/_{N} \sum SSE_i$
3. *k = the first k which has the amount of improvement rather than previous k is less than 1%*

5.3.3 Experimental Results

The results of the comparison follow, starting with the Davis Boulding index in Table 5-2. As shown, SOM performed better than the deep auto-encoder with all five clustering algorithms. For SOM, the performance from best to worst was k-means, Birch,

Table 5-2 – Davis-Boulding Index Results

Feature extraction	Clustering	Average	Minimum	Maximum	std
SOM	K-means	**0.120**	0.112	**0.130**	0.008
	MINIbatchKM	0.140	0.116	0.177	0.023
	Spectral	0.172	0.125	0.305	0.075
	Agglomerative	0.140	**0.107**	0.162	0.025
	Birch	0.132	0.114	0.155	0.017
Deep Auto-encoder	K-means	0.154	0.148	0.160	**0.005**
	MINIbatchKM	0.165	0.142	0.189	0.018
	Spectral	0.204	0.158	0.317	0.064
	Agglomerative	0.169	0.138	0.199	0.028
	Birch	0.159	0.140	0.184	0.019

MINIbatchKM, agglomerative, and spectral clustering. Similarly, k-means clustering achieved an outstanding performance in DAE in comparison to the others. Notably, k-means had the lowest standard deviation and was also the most stable in different folds with both SOM and the deep auto-encoder. Therefore, from a Davis-Boulding point of view, SOM + k-means is the optimal choice.

The results against the Calinski-Harabasz index are shown in Table 5-3. Again, k-means clustering had the highest average CH score with a reasonable standard deviation with both SOM and the deep auto-encoder. In this case, DAE+k-means clustering placed first, followed by SOM+k-means. The SOM+Spectral clustering is the third method with a high CH score, but the standard deviation for this model is very high. From a deeper analysis of each fold, we found this algorithm was always unstable.

Table 5-3 – Calinski-Harabasz Index Results

Feature extraction	Clustering	Average	Minimum	Maximum	std
SOM	K-means	17,375.60	16,754.93	17,891.93	475.62
	MINIbatchKM	15,136.85	14,784.21	15,765.54	381.08
	Spectral	17,040.31	15,191.46	**19,364.84**	1962.88
	Agglomerative	14,871.24	14,714.06	15,010.68	138.09
	Birch	14,753.34	14,528.51	15,171.88	**255.69**
Deep Autoencoder	K-means	**18,242.75**	**17,248.70**	18,967.43	640.11
	MINIbatchKM	16,266.25	15,744.51	16,813.88	480.94
	Spectral	16,292.00	14,336.08	18,900.63	1671.52
	Agglomerative	15,582.44	15,178.67	16,010.59	408.40
	Birch	15,522.14	15,035.29	16,199.02	457.58

Table 5-4 shows the execution times. Efficiency is an important factor since data on driving characteristics tends to be very large-scale and unsupervised learning algorithms are prone to long runtimes.

As the results show, SOM had much faster running times with all clustering methods than the deep auto-encoder. The most important reason for this difference is that, with a deep auto-encoder, one record corresponds to one point while, with SOM, one point equals an abstracted group of records.

Table 5-4 - Execution Time Results (Minutes)

	SOM	Deep Auto-encoder
K-means	40.87	180.93
MINIbatchKM	**33.47**	**66.99**
Spectral	82.35	250.70
Agglomerative	41.03	199.38
Birch	41.30	190.73

Across all three metrics, the optimal clustering choice for driving style pattern recognition is clear – SOM + k-means, firstly because it had a very low DB index in comparison to other methods, which means that the extracted clusters with SOM+k-means are unique and they are less similar to other clusters in comparison to other techniques (Liu et al. 2010). In addition, the CH index in a deep auto encoder is slightly better than SOM+k-means, and this difference is not large enough to encourage us to select this algorithm as the selected method as its computation cost is very high and BD index is very low.

5.4 Extracted Driving Patterns

From the three tests in the previous section, we determined that k-means in tandem with SOM was the best overall algorithm to recognize driving style patterns. The next step is finding the optimal number of clusters. We used Algorithm 5-2 to determine the optimal number of clusters using the SOM+k-means clustering algorithm. We found that the optimal number of clusters was 29. Figure 5-4 illustrates the sum of square errors for different numbers of clusters in each iteration, showing 29 as the optimum number of clusters, because 29 clusters does not exceed the defined threshold of 1%, nor does it meet the stopping condition of less than 1% improvement.

Hence, we extracted 29 unique driving behaviours from our data set. Each cluster is a group of time-series data and raw numbers. In relation to the results, however, raw numbers do not mean much to transportation experts, so we need to understand each driving pattern and find a meaningful name for each cluster. We followed a matching method to find a suitable label for each driving behaviour. In this algorithm, first, we reviewed various driving behaviours introduced by top-ranked, highly cited publications and selected three papers to review: (Chen et al. 2015; Fazeen et al. 2012; Yu et al. 2017). Second, we developed a

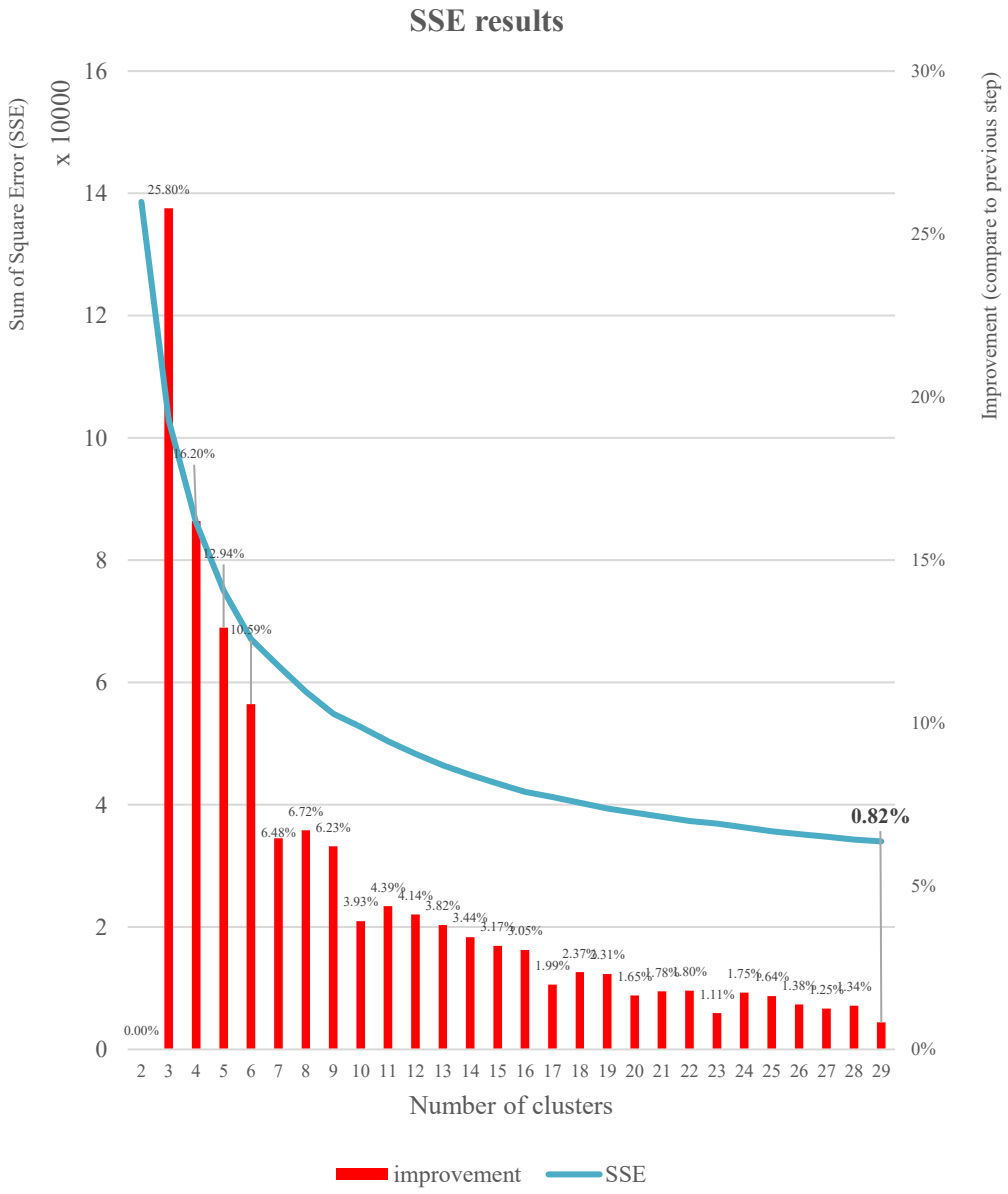

Figure 5-4 - Sum Of Square Error Per Number Of Clusters For SOM + K-Means Clustering

descriptive analytics to understand each category using average, minimum, maximum, and standard deviation across velocity, acceleration, x-axis and y-axis acceleration. Then, we compared the similarity between the extracted driving patterns and current driving behaviours in the literature. We then selected the most similar pattern in the literature as a representative of each category. Finally, we named each cluster to reflect the name of the most similar driving behaviour found in the literature.

After implementing the matching method, we understand the characteristics of all 29 clusters. For example:

- Cluster 17 represents those who drive at very low speed with low acceleration and accounts for 16.5% of the events. This behaviour shows that these drivers have a tendency to stop the vehicle.
- Cluster 29, the y-acceleration is close to zero and the x-acceleration is higher than zero for a short period of time, which is similar to the cornering behaviour described by Fazeen et al. (2012).
- Cluster 13, is normal driving behaviour, i.e., standard speeds with very low acceleration, few changes, and a small standard deviation. Yu et al. (2017) describe this type of driving as "normal driving behaviour".
- Cluster 8 exhibits swerving behaviour as described by Chen et al. (2015). The value of both the x-axis and y-axis acceleration has a high peak value with a high standard deviation.
- Drivers in Cluster 2 exhibit weaving behaviour at high speeds. They have a very high variation between x- and y-axis acceleration. The standard deviation of acceleration is very high, and the mean value of acceleration is high (Yu et al. 2017).
- Cluster 6 reflects sudden braking and accounts for 4.3% of the driving events. x-acceleration remains unchanged while y- acceleration significantly decreases, and the standard deviation of the y-acceleration is high (Chen et al. 2015).

- Cluster 26 is characterized by high variations in both x- and y-acceleration. The velocity range is medium, and the standard deviation of acceleration is high with low acceleration.
- In the above, we explained the top clusters that account for the 50% of driving events to describe how the clustering results are matched with a corresponding name in the transportation research. Additional information on the other clusters, along with the statistical results for each group, are provided in Tables 5-5 to 5-8.

Algorithm 5-3 - The Matching Method

Input: Extracted driving behaviour, current driving behaviour in the literature

Output: Corresponding driving patterns in the literature, and the name of all clusters

1. *DB = 29 extracted driving behaviours by SOM + k-means*
2. *DB_lit = various driving behaviours in the current literature*
3. *For each driving_behaviour in DB:*
   - 3.1. *Developing descriptive analytics with minimum, maximum, average, and standard deviation across all input variables*
   - 3.2. *Corresponding_patterns[driving_behaviour] = the most similar pattern in the literature for driving_behaviour*
   - 3.3. *Names[driving_behaviour] = finding a suitable name according to the corresponding driving behaviour*
     - 3.3.1. *DB[clustering , fold] = average value of Davis Boulding for current fold by using test data*
4. *Return Corresponding_patterns, names*

Table 5-5 - Frequent Driving Behaviors - Part I

Cluster number	Frequency Percentage (%)	Average speed (km/h)	Average acceleration (m/s^2)	Acceleration std	Name	Patterns of behavior
17	16.49%	1.961	-0.108	0.088	Warm stopping	Drivers in this group drive at low speeds with low deceleration and are prone to stopping.
29	9.08%	45.808	0.353	0.202	Cornering with medium speed	During cornering behavior, y- acceleration is close to zero. x-acceleration is high with significant standard deviation (Fazeen et al. 2012).
13	6.68%	50.550	0.333	0.022	Driving at normal speed	Drivers proceed at normal speed with very low acceleration and standard deviation.
8	5.18%	39.130	0.011	1.288	Swerving at medium speed	While swerving, x- and y-acceleration both have a high peak and high standard deviation (Chen et al. 2015).
2	4.96%	80.925	0.002	2.225	Weaving at high speed	There is high variation between x- and y-acceleration. y-acceleration is very smooth over an extended period. The standard deviation of acceleration is high, but with a low mean (Yu et al. 2017).
11	4.59%	87.621	0.1114	0.163	Cornering at high speed	During cornering, speeds are high and x-acceleration increases rapidly over a short period of time, while y-acceleration is almost zero (Yu et al. 2017).
27	4.54%	70.784	-0.625	0.1769	Sideslipping with high speed	y-axis acceleration fell sharply, the minimum and average value of y-axis acceleration is negative and x-axis acceleration is not near to zero (Chen et al. 2015).
23	4.49%	76.21	0.0315	2.2296	weaving with high speed	There is high variation between x- and y-acceleration. y-acceleration is very smooth over an extended period. The standard deviation of acceleration is high, but with a low mean (Yu et al. 2017).
6	4.37%	61.207	-2.258	1.257	Sudden braking from a high speed	During sudden braking, x-acceleration remains unchanged, and y- acceleration decreases significantly. Standard deviation in y-acceleration is very high (Chen et al. 2015).

Table 5-6 - Frequent Driving Behaviors - Part II

Cluster number	Frequency Percentage (%)	Average speed (km/h)	Average acceleration (m/s^2)	Acceleration std	Name	Patterns of behavior
26	3.86%	56.389	0.190	2.039	Weaving at medium speed	There is a high variation between x- and y-acceleration. y-acceleration is very smooth over extended periods. The standard deviation of acceleration is high. The mean value of acceleration is very low (Yu et al. 2017).
12	3.39%	65.784	-0.381	0.151	Cornering at high speed	During cornering, speeds are high and x-acceleration increases rapidly over a short period of time, while y-acceleration is almost zero (Yu et al. 2017).
18	3.10%	40.558	6.838	0.963	High acceleration behavior	Acceleration is high over a very short time reflective of sudden maneuvering habits (Fazeen et al. 2012).
16	2.87%	95.259	0.242	0.123	Cornering at very high speed	During cornering, speeds are very high and x-acceleration increases rapidly over a short period of time, while y- acceleration is almost zero (Yu et al. 2017).
3	2.43%	8.499	-1.157	1.179	Cornering at low speed	During cornering, speed is low and x-acceleration increases significantly for a short period of time. y-acceleration is almost zero (Yu et al. 2017).
28	2.34%	26.679	0.278	1.307	Changing lanes at low speed	Y-acceleration decreases before changing lanes and increases afterwards (Fazeen et al. 2012).
22	2.20%	32.458	-1.578	0.797	Sudden braking from a medium	During sudden braking, x-acceleration remains unchanged, and y- acceleration decreases significantly. Standard deviation in y-acceleration is very high (Yu et al. 2017).
14	1.95%	12.707	-6.173	1.730	Sideslipping with low speed	y-acceleration decreases sharply with a high range in x-acceleration and negative value. All these behaviors occur within a very short period (Chen et al. 2015).
9	1.88%	12.576	-7.108	1.836	Sudden braking from a low velociy	During sudden braking, the x-acceleration remains unchanged, and the y-acceleration decreases significantly. The standard deviation of the y-acceleration is very high (Yu et al. 2017).

Table 5-7 - Frequent Driving Behaviors -Part III

Cluster number	Frequency Percentage (%)	Average speed (km/h)	Average acceleration (m/s^2)	Acceleration std	Name	Patterns of behavior
10	1.82%	19.547	0.575	1.030	Left lane change	A left lane change is formed by a small decrease in the value of x-axis acceleration, followed by an increase in x-axis acceleration (Fazeen et al. 2012).
25	1.81%	40.335	6.273	2.628	Sudden starting	Very high acceleration starting from a stop up to a very high average speed.
15	1.80%	56.533	-7.213	1.447	Sideslipping with medium speed	y-acceleration decreases sharply with a high range in x-acceleration and negative value. All these behaviors occur within a very short period (Chen et al. 2015).
7	1.71%	34.481	1.096	1.616	Right lane change	A right lane change is formed by a small increase in the value of x-axis acceleration, followed by a decrease in x-axis acceleration (Fazeen et al. 2012).
1	1.63%	37.197	-2.111	0.842	Safe cornering	Drivers drive at medium speed and decrease their speed before turning.
5	1.50%	107.15	5.160	0.154	Turning with a wide radius	High and rapid x-acceleration with a y-acceleration of close to zero. The x-acceleration mean is far from zero (Yu et al. 2017)
19	1.37%	32.098	-9.408	2.055	Unknown	Driving patterns in this group are unknown and are not understandable for experts.
24	1.29%	44.609	-10.51	2.087	Fast U-turn	A rapid rise in x-acceleration to a very high value followed by a rapid drop to a very low value for a short period of time. The standard deviation acceleration is high (Chen et al. 2015).
4	0.98%	52.887	-6.621	1.975	Sudden braking	Speed decreases over a very short time. This type of sudden braking is much more dangerous than in Cluster 6 because the deceleration is much higher.

Table 5-8 - Frequent Driving Behaviors - Part IV

Cluster number	Frequency Percentage (%)	Average speed (km/h)	Average acceleration (m/s^2)	Acceleration std	Name	Patterns of behavior
20	0.96%	29.389	-1.064	1.131	Turn right cornering	Drivers decrease their speed and acceleration changes from the x-axis to the y-axis. This group could be merged with cluster number 26.
21	0.73%	120.49	-0.106	0.152	Very high speed with low	Driving at very high speed without any significant changes.

These tables show the 29 clusters extracted with SOM+ K-means, which represent the driving patterns within our dataset. The average speeds and acceleration values are the means of the velocity and instantaneous acceleration figures for all driving patterns in each group. The names for each group were derived from the patterns and values at each cluster center with reference to previous studies in transportation.

5.5 SUMMARY

Understanding driving patterns with unsupervised learning techniques is an underexplored area of research and finding the best clustering algorithm with the highest performance and the optimal number of clusters is still problematic in this domain. In this chapter, we proposed an empirical analysis on driving characteristics using information collected by mobile telematics devices to find an efficient clustering algorithm in this domain with an optimal number of clusters. Moreover, driving behaviours are categorized into similar groups using this algorithm. The experiment results show that SOM + K-means clustering algorithm is the best choice for this domain.

In the next chapter, we propose a decision support system which automatically extracts criteria for risk assessment from mobile telematics big data. To ensure the system is comprehensive and effective, we design a risk assessment framework that can evaluate the probability and severity of each pattern and calculate a risk score for each unique behaviour.

In addition, we utilize the most important findings in this to provide a reliable clustering algorithm for driving style pattern recognition.

Chapter 6:

FUZZY RISK ASSESSMENT

6.1 INTRODUCTION

The risk of the occurrence of unwanted situations or events is traditionally calculated based on probability and severity, and risk assessment is a process to estimate these two elements. This chapter relies on these principles and proposes a fuzzy risk assessment component as illustrated in Figure 6-1. The component has two main parts. First, the risk factor identification part extracts the decision-making criteria from mobile telematics data and provides a fuzzy risk modelling for driving events. Second, the risk of driving events is calculated by the fuzzy expert system provided in the risk factor identification part.

The prepared data is used by the unsupervised learning algorithm to extract hidden driving patterns from driving styles. These driving events play the role of criteria for decision making and a fuzzy risk model is provided according to the extracted driving events from this expert system. Finally, a risk score is calculated for driving events according to the knowledge extracted from mobile telematics. This chapter explains the related parts of this component in the following subsections.

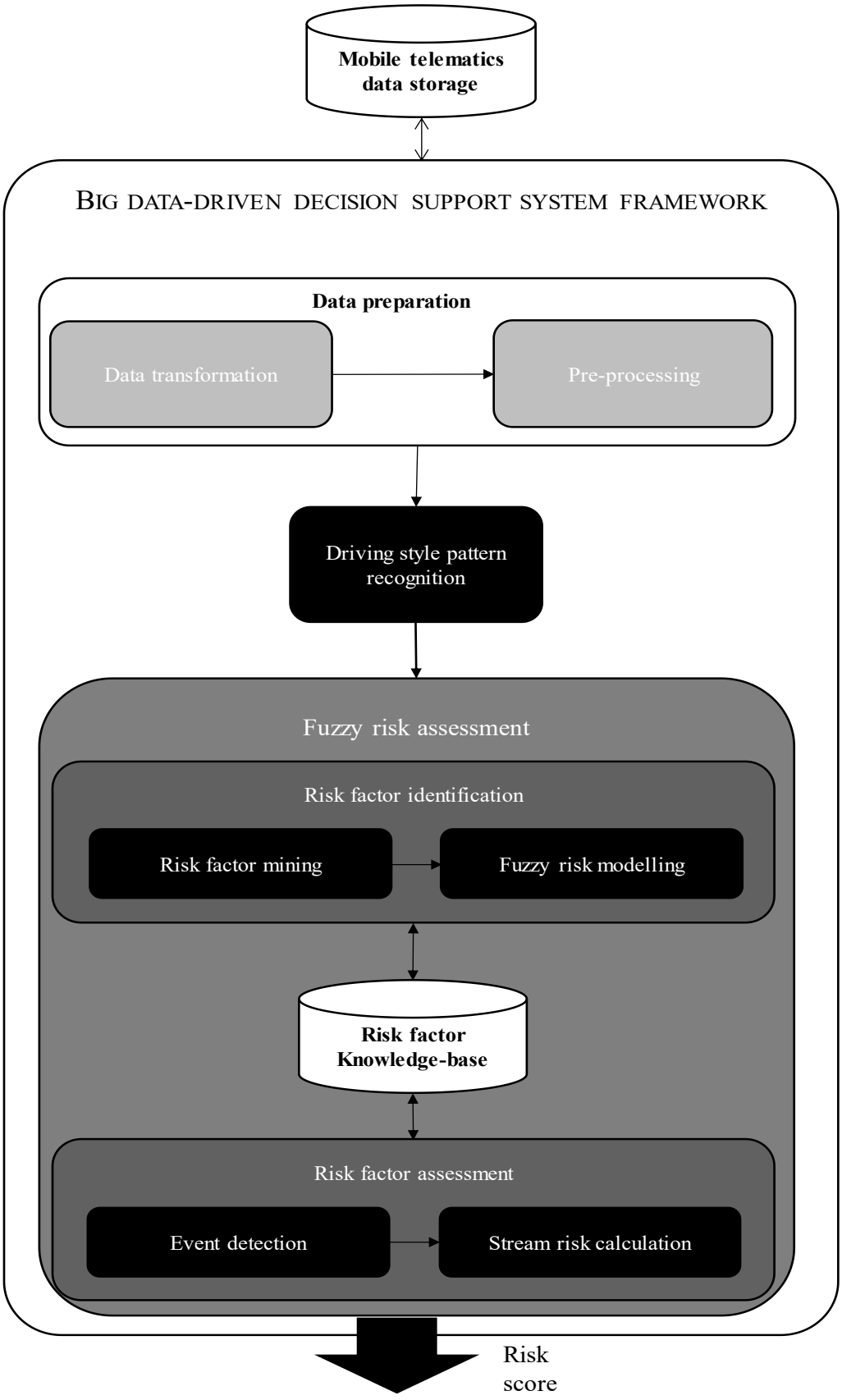

Figure 6-1 – Fuzzy Risk Assessment Component

6.2 RISK FACTOR IDENTIFICATION

This part uses the data prepared by the data preparation component which is explained in Chapter 4, mines the risk factors using unsupervised learning according to the empirical analysis in Chapter 5, and estimates the risk level of each category by providing a fuzzy risk assessment model.

6.2.1 RISK FACTOR MINING

Defining criteria for decision-making is a critical step for any risk assessment and decision support system. A complete list of all risk and success criteria should be defined according to the previous literature and expert judgment. This process is costly and time-consuming (Ahmadabadi & Heravi 2019). In order to remove this process and decrease the processing time, the risk factor mining model proposes an unsupervised learning model to extract decision-making criteria from mobile telematics data. In this part, according to the empirical study in Chapter 5, a two-stage clustering algorithm using SOM and fuzzy k-means clustering is proposed to extract unique patterns from data. These unique patterns play the role of criteria in the proposed decision support system.

- **Self-organizing map**

The SOM algorithm is explained in Chapter 5. This algorithm is used to reduce the complexity of data and prepare the data for clustering.

- **Fuzzy clustering**

Mobile telematics stream data is very similar to each other and one stream of data is similar to more than one group of data. Thus, the fuzzy k-means clustering is used to calculate one score to find the similarity between stream data and selected clusters (Shen et al. 2019).

Fuzzy clustering is one of the widely used clustering algorithms. SOM has reduced the stream data into an abstract subspace. However, there will still be too many points to analyze

directly, so they need to be clustered into similar groups. Therefore, we utilized a fuzzy clustering algorithm to extract the unique patterns from the streams. In fuzzy clustering, the following objective function is used to cluster abstract data X from SOM.

$$O = \sum_{i=1}^{c} \sum_{k}^{N} u_{ik}^{m} \|x_k - v_i\|^2 \quad (6\text{-}1)$$

where the square weighted distance is calculated by:

$$\|x_k - v_i\|^2 = \sum_{j=1}^{n} \frac{(x_{kj} - v_{ij})^2}{\sigma_j^2} \quad (6\text{-}2)$$

where m is the fuzzification coefficient which is greater than 1, σ_j is the standard deviation of the jth feature. The input data X has n records and we want to cluster them to c number of clusters, and U is the partition matrix with the shape of c×n. V is the center of clusters. Algorithm 6-1 explains the fuzzy clustering algorithm.

The fuzzy clustering provides a partition matrix U, which depicts the coefficient of each record to the clusters. The clustering results will be used in the next step for risk modelling.

Algorithm 6-1 - Fuzzy Clustering Algorithm (Shen Et Al. 2019)

Input: Data set X and c number of clusters, ε very small threshold

Output: data points with cluster label

1) ***Set the number of clusters as an input parameter.***
2) ***Initialize $U_{c\times n}$ as the partition matrix***
3) ***Do***
   - ✓ $\boldsymbol{v_{ij} = \frac{\sum_{k=1}^{N} u_{ik}^{m} x_{kj}}{\sum_{k=1}^{N} u_{ik}^{m}}}$
   - ✓ $\boldsymbol{u_{ij} = 1 / \sum_{s=1}^{c} \left(\frac{\|x_k - v_i\|}{\|x_k - v_s\|} \right)^{2/m-1}}$

While $\boldsymbol{\|U_{Iter} - U_{iter-1}\| < \varepsilon}$

6.3 FUZZY RISK MODELLING

After extracting the unique patterns from the driving streams using SOM + fuzzy clustering, these patterns are used in the proposed fuzzy risk assessment model as criteria for risk assessment. The fuzzy risk modelling uses these criteria and provides a risk score. Finally, the extracted criteria with their risk score are stored in the risk factor knowledge-base.

The risk modelling is done through an FLS which is able to handle uncertain and vague variables and simulate human reasoning. The FLS takes into account the factor probability and severity estimations and produces a risk level. The probability of all events is calculated using predictive analysis. We estimate the likelihood of all events using statistical analysis based on the frequency of previously occurred events. On the other hand, assessing the severity of an event mainly depends on a business problem, and a number of domain experts should conduct an investigation process to assess the severity of an event. In this study, we calculated the severity of the events using previous research conducted in the transportation field (Eboli, Mazzulla & Pungillo 2017; Siami et al. 2018).

To estimate the risk level of the factors, the FLS needs to use membership functions. A number of membership functions can be used to determine fuzzy linguistic variables, such as triangular, trapezoidal, and Gaussian. Selecting a suitable membership function fundamentally depends on the characteristics of the variable, the available information and expert knowledge. We assume that the parametric (trapezoidal/triangular) functions are good enough to capture the vagueness of the variables. Therefore, a combination of trapezoidal and triangular functions are used (Naderpour, Lu & Zhang 2013). The relations between the input and output variables are defined using a risk matrix. Mamdani's fuzzy interface method is described to implicate each rule and aggregate input variables to risk output. Finally, the fuzzy output variable of risk is converted to a crisp variable during the defuzzification process.

6.4 RISK ASSESSMENT

After finding the risk score of all the extracted criteria using the fuzzy clustering algorithm, we provide a methodology to assess the risk of new events according to the knowledge offered from big data.

6.4.1 EVENT DETECTION

Event detection is a part of the fuzzy risk assessment component that detects the most similar patterns in the knowledge-base. In this part, fuzzy clustering provides a U partition matrix for all new events according to their similarity to the previously known events. Let U be the partition matrix from fuzzy clustering:

$$U\ partition\ matrix = \begin{matrix} \\ D_1 \\ \vdots \\ D_i \end{matrix} \begin{matrix} \begin{matrix} C_1 & C_2 & \cdots & C_j \end{matrix} \\ \begin{bmatrix} U_{11} & \cdots & U_{1j} \\ \vdots & \ddots & \vdots \\ u_{i1} & \cdots & u_{ij} \end{bmatrix} \end{matrix} \tag{6-3}$$

where D_i is a multidimensional stream data. C_j is the list extracted clusters. u_{ij} is the similarity score of ith input stream to the jth clusters. The fuzzy clustering algorithm is responsible for providing these fuzzy scores.

6.4.2 STREAM RISK CALCULATION

Stream risk calculation is the final part in the fuzzy risk assessment component. Each stream contains many fixed-length windows, so in this part, we calculate the total risk per event according to the following equation:

$$RE_i = \mathcal{F}_k\left(\sum_{j=1}^{C} R_j u_{ij}\right), (i = 1,2,3,\dots,N, k = 1,2,\dots,c) \tag{6-4}$$

where $\mathcal{F}$ is the function which is used to calculate the top k similar events. N is the total number of events for which we want to calculate the risk. c is the total number of criteria, which is extracted using the clustering algorithm. R is the risk score of the event and u_{ij} is the similarity score which is calculated by fuzzy clustering.

The total risk score per driving stream entirely depends on the events' risk (RE_i) for all sub-streams, thus the total risk per streams will be obtainable using the following equation:

$$Risk\ Of\ Stream = \mathcal{AF}(RE_i), (i = 1,2,3, \dots, N) \quad (6\text{-}5)$$

where $\mathcal{AF}$ is the aggregation function which could be optimistic, pessimistic or neutral. This aggregation function depends on the business problem and business strategy. For example, the total risk score in the optimistic strategy can be equal to the minimum score of all events. In a pessimistic strategy, the maximum value can be considered as the total risk. Finally, for a neutral strategy, the average value of all events is regarded as the total risk value of the stream.

6.5 IMPLEMENTATION

To evaluate the performance of our proposed system, we implemented it in a real case study to assess the risk of drivers according to their driving behaviours.

We implement the fuzzy risk assessment component in Python 2.7 on an Intel® Xeon® 3.01 GHz CPU, 64 GB of RAM, running a Linux operating system. The software platform was Anaconda 2.7. We used the implemented versions of SOM (Saraee, Vahid Moosavi & Rezapour 2011), scikit-fuzzy, and change detection libraries (Liu et al. 2013).

As explained in Chapter 4, the stream data has been divided into 15-second fixed-length time windows with a one-second sliding step because each single driving event can be completed in 15-second. Then, the RuLSIF-based change detection scores were applied to remove unnecessary data from the streams. According to the score extracted from the change detection algorithm, the most important time frames are selected. Therefore, 394,833 windows are selected according to the outcome of the data preparation component.

6.5.1 RISK FACTOR MINING

After completing the data preparation and removing the unnecessary time frames by selecting the time windows with the highest change score, the analytical process is started to

extract risk criteria from driving behaviours. A two-stage unsupervised learning framework is proposed to extract driving patterns from big data. Chapter 4 explains the optimal number of nodes for SOM according to the input samples and identifies a number of nodes equal to $5 \times \sqrt{n}$ where n is the total number of selected events. With 394,833 events, the optimal number of nodes is 2814. In addition, we selected a map size of 21×134 based on the eigenvalues and eigenvectors (Vesanto & Alhoniemi 2000).

After reducing the complexity of data using SOM, we clustered the data points using fuzzy clustering. Finding the optimum number of clusters is crucial for any partitive clustering algorithm; thus the empirical analysis which is undertaken in Chapter 5 shows that 29 clusters is the optimal number of clusters for this study. Table 6-1 summarizes the information on the extracted clusters from the data.

Table 6-1 - Fuzzy Clustering Result

Cluster number	Frequency Percentage (%)	Cluster number	Frequency Percentage (%)	Cluster number	Frequency Percentage (%)
17	16.493%	12	3.388%	15	1.803%
29	9.082%	18	3.101%	7	1.706%
13	6.682%	16	2.872%	1	1.629%
8	5.179%	3	2.428%	5	1.496%
2	4.955%	28	2.345%	19	1.371%
11	4.594%	22	2.200%	24	1.287%
27	4.545%	14	1.947%	4	0.978%
23	4.495%	9	1.875%	20	0.960%
6	4.369%	10	1.822%	21	0.728%
26	3.863%	25	1.808%		

The clustering results of the selected dataset are summarized in the Table 6-1. These clusters are extracted patterns from the driving style database and they play the role of decision-making criteria in our proposed decision support system.

6.5.2 FUZZY RISK MODELLING

After extracting the unique driving patterns from the driving streams, the risk of these patterns needs to be assessed using fuzzy logic. In order to do this, the probability and severity of each driving pattern is calculated, then the risk score of each driving pattern is calculated using the risk matrix.

1) The probability estimation

According to the criteria mining step, 29 possible driving patterns are extracted from the driving dataset. These patterns are unique with different likelihoods of occurrence. The probability of each driving pattern is calculated using the following equation:

$$P(E_i) = \frac{n(E_i)}{\sum_{i=1}^{c} n(E_i)} \quad (6\text{-}6)$$

where $n(E_i)$ is equal to the number of times that the i^{th} driving pattern occurs, which is divided by the total number of driving events.

The probability score of all driving events is calculated and the results show that these scores are in a range between 0.7% and 16.5%. Thus, the min-max transformation is used to normalize data in a range between zero and one.

2) The severity estimation

Assessing the severity of an event depends on the case study and is usually conducted by an investigating process with a number of experts in risk assessment. In this case, severity analysis is undertaken using previous research conducted by domain experts in the transportation field (Eboli, Mazzulla & Pungillo 2017; Siami et al. 2018).

In this study, the research of Eboli, Mazzulla & Pungillo (2016) is followed to find the severity of each driving pattern. They explored the relationship between velocity and acceleration to distinguish dangerous driving conditions. They found a correlation between dangerous driving patterns, instantaneous velocity and acceleration. Based on their findings, a driver's behaviour is risky when the value of acceleration is larger than the defined threshold in the following equation.

$$|\bar{a}| = g.\left[0.198.\left(\frac{V}{100}\right)^2 - 0.592.\left(\frac{V}{100}\right)^2 + 0.569\right] \quad (6\text{-}7)$$

where $|\bar{a}|$ is the instantaneous acceleration norm, and V is the value of velocity (km/h). g denotes gravity, which is equal to 9.18 (m/s^2). According to this equation, when the value of acceleration is more than $|\bar{a}|$(m/s^2), the driver is engaging in risky behaviour.

This equation is used to find the percentage of the abnormal acceleration value for each driving event. The severity of driving events in mobile telematics data skews the distribution; thus, log transformation and min-max transformation is used to standardize these average number of dangerous events per second in a range between zero and one.

3) Event risk estimation

The FLS uses the developed membership functions as shown in Figure 6-2 and detailed in Tables 6-2 to 6-4 to calculate the risk levels. The relation between probability and severity variables with risk are shown in Table 6-5. For example, if the probability is U and the severity is M, then the risk is TA. Mamdani's fuzzy inference method is used to calculate the output risk score. Table 6-6 describes the functions used to obtain the fuzzy outcome from the input variables. Finally, the defuzzification process is used to transform the fuzzy risk set to a crisp risk score.

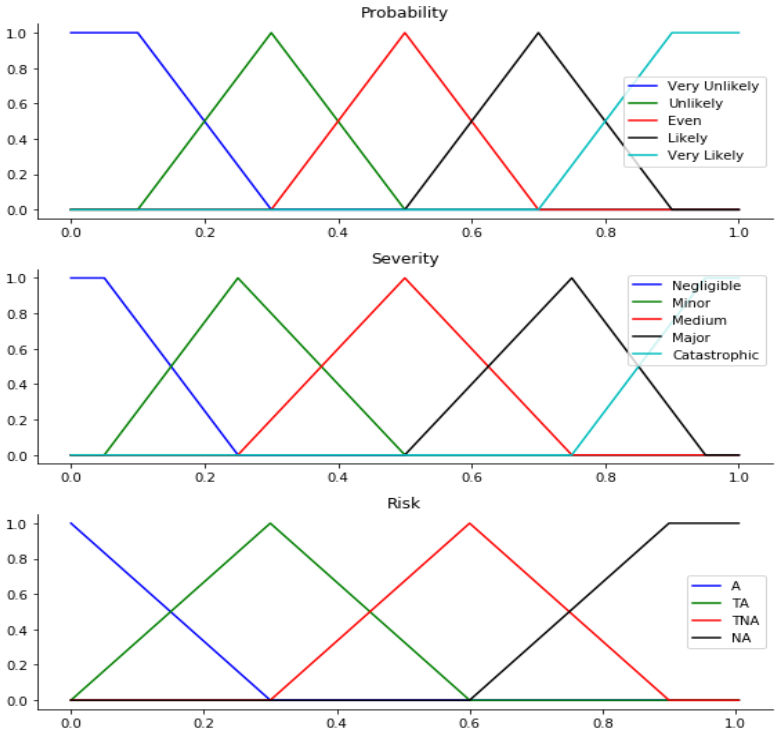

Figure 6-2 - Fuzzy Membership Functions Of Probability, Severity, And Risk

Table 6-2 - Probability Linguistic Variables

Fuzzy set	Linguistic term	α cut	
		Level 1	Level 0
VU	Very Unlikely	0,0.1	0.3
U	Unlikely	0.3	0.1,0.5
E	Even	0.5	0.3,0.7
L	Likely	0.7	0.5,0.9
VU	Very Unlikely	0.9,1	0.7

Table 6-3 - Severity Linguistic Variables

Fuzzy set	Linguistic term	α cut	
		Level 1	Level 0
N	Negligible	0, 0.1	0.3
MI	Minor	0.3	0.1,0.5
M	Medium	0.5	0.3,0.7
MA	Major	0.7	0.5,0.9
C	Catastrophic	0.9,1	0.7

Table 6-4 - Risk Linguistic Variables

Fuzzy set	Linguistic term	α cut	
		Level 1	Level 0
A	Acceptable	0	0.3
TA	Tolerable Acceptable	0.3	0,0.6
TNA	Tolerable not acceptable	0.6	0.3,0.9
NA	Not Acceptable	0.9,1	0.6,0.9

Table 6-5 - Risk Matrix

		Severity				
		N	MI	M	MA	C
Probability	VL	TNA	TNA	NA	NA	NA
	L	TA	TNA	TNA	NA	NA
	E	A	TA	TNA	NA	NA
	U	A	A	TA	TNA	NA
	VU	A	A	TA	TNA	TNA

Table 6-6 - Mamdani Model (Mamdani 1977)

Operation	Operator	Formula
Union (OR)	MAX	$\mu_c = \max(\mu_A(x), \mu_B(x))$
Intersection (AND)	MIN	$\mu_c = \min(\mu_A(x), \mu_B(x))$
Implication	MIN	$\mu_c = \min(\mu_A(x), \mu_B(x))$
Aggregation	MAX	$\mu_c = \max(\mu_A(x), \mu_B(x))$
Defuzzification	CENTROID	$COE = Z^* = \frac{\int z\, \mu_c(z) dz}{\int \mu_c(z) dz}$

We developed the proposed fuzzy risk estimation model to calculate the risk score of all driving events. Table 6-7 shows the calculated risk score for each driving category. The results show that cluster numbers 20 and 21 have the top two dangerous driving events. Cluster 20 is a very high-risk cluster and accounts for 0.96% of all driving events. This cluster represents those who drive dangerously during cornering. The other high-risk cluster is Cluster 21. This cluster reflects the behaviour of reckless drivers who drive faster than the speed limit without any significant changes. On the other hand, driving behaviours in Cluster 29 pose a low risk and the behaviour in this group shows changing lanes with low speed.

6.5.3 Stream Risk Calculation

After calculating the risk score of all the extracted driving patterns using fuzzy logic, we calculate the risk score of new driving events according to the provided knowledge. Table 6-8 shows the calculation procedure of one trip. In this table, the columns show the list of all the extracted driving behaviours in our decision support system, and the rows show the list of detected driving events in one trip. The cells show the similarity score between driving events in the trip and the extracted driving patterns in the knowledge-base.

Table 6-7 - Calculated Risk Scores

Cluster Number	Probability Score	Severity Score	Risk Score	Cluster Number	Probability Score	Severity Score	Risk Score
1	0.19	0.75	0.600	16	0.26	0.5	0.300
2	0.31	0.33	0.265	17	0.54	0.17	0.351
3	1.00	0	0.600	18	0.50	0.08	0.187
4	0.28	0.83	0.666	19	0.31	0.58	0.415
5	0.59	0.08	0.326	20	0.33	1	0.857
6	0.46	0.08	0.180	21	0.29	0.92	0.795
7	0.27	0.58	0.406	22	0.52	0.08	0.259
8	0.43	0.17	0.277	23	0.51	0.17	0.300
9	0.43	0	0.110	24	0.38	0.42	0.408
10	0.53	0.17	0.333	25	0.00	0.67	0.494
11	0.42	0.42	0.455	26	0.14	0	0.104
12	0.32	0.33	0.288	27	0.33	0.17	0.197
13	0.24	0.5	0.300	28	0.27	0.25	0.102
14	0.58	0.08	0.324	29	0.21	0.67	0.483
15	0.09	0.42	0.286				

Table 6-8 shows the calculation process for a selected trip. A short trip is selected. The trip has 31 detected events according to driver behaviour. The similarity score is calculated for all events with all risk factors which are extracted from our knowledge base. Afterwards, the risk score of each event was calculated according to the most similar driving behaviour risk score. Finally, the risk score of the selected trip according to the optimistic, pessimistic, and neutral strategies is 0.192, 0.646, and 0.318 respectively.

Table 6-8 - Trip Risk Calculation Process

E\RF	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	RE
1	0.002	0.025	0.050	0.002	0.001	0.003	0.038	0.019	0.129	0.029	0.059	0.057	0.175	0.006	0.071	0.015	0.030	0.003	0.033	0.030	0.014	0.002	0.018	0.008	0.004	0.089	0.005	0.007	0.074	**0.219**
2	0.002	0.058	0.022	0.002	0.001	0.003	0.033	0.011	0.170	0.016	0.031	0.026	0.058	0.004	0.091	0.009	0.016	0.003	0.076	0.016	0.009	0.002	0.011	0.006	0.003	0.033	0.004	0.005	0.280	**0.342**
3	0.001	0.196	0.014	0.001	0.001	0.002	0.036	0.008	0.046	0.012	0.015	0.016	0.033	0.003	0.094	0.006	0.011	0.002	0.246	0.010	0.006	0.002	0.007	0.004	0.003	0.020	0.003	0.004	0.197	**0.445**
4	0.002	0.235	0.015	0.001	0.001	0.002	0.070	0.008	0.027	0.015	0.012	0.016	0.034	0.004	0.119	0.007	0.012	0.002	0.281	0.011	0.007	0.002	0.008	0.005	0.003	0.021	0.003	0.004	0.075	**0.347**
5	0.002	0.070	0.023	0.002	0.001	0.003	0.289	0.011	0.027	0.025	0.014	0.023	0.059	0.004	0.178	0.009	0.018	0.003	0.093	0.016	0.009	0.002	0.011	0.006	0.003	0.034	0.004	0.005	0.057	**0.360**
6	0.002	0.029	0.045	0.002	0.001	0.003	0.201	0.018	0.031	0.053	0.019	0.043	0.148	0.006	0.109	0.014	0.034	0.003	0.038	0.027	0.013	0.002	0.016	0.008	0.004	0.077	0.005	0.006	0.042	**0.361**
7	0.002	0.012	0.103	0.001	0.001	0.003	0.056	0.026	0.022	0.114	0.020	0.084	0.109	0.006	0.035	0.018	0.067	0.003	0.015	0.047	0.017	0.002	0.023	0.009	0.004	0.169	0.005	0.007	0.021	**0.196**
8	0.002	0.009	0.135	0.001	0.001	0.003	0.026	0.032	0.021	0.082	0.026	0.098	0.083	0.006	0.022	0.021	0.084	0.003	0.011	0.067	0.020	0.002	0.028	0.010	0.004	0.176	0.005	0.007	0.016	**0.319**
9	0.001	0.005	0.230	0.001	0.001	0.003	0.014	0.037	0.012	0.059	0.017	0.157	0.030	0.005	0.011	0.022	0.112	0.002	0.006	0.094	0.020	0.002	0.032	0.009	0.003	0.097	0.004	0.006	0.009	**0.474**
10	0.003	0.014	0.097	0.002	0.001	0.005	0.028	0.038	0.039	0.053	0.060	0.085	0.103	0.009	0.032	0.027	0.067	0.004	0.016	0.068	0.025	0.003	0.035	0.014	0.006	0.124	0.007	0.011	0.026	**0.192**
11	0.002	0.014	0.090	0.002	0.001	0.004	0.029	0.031	0.067	0.037	0.068	0.144	0.080	0.008	0.034	0.023	0.048	0.004	0.018	0.053	0.021	0.003	0.028	0.012	0.005	0.124	0.006	0.009	0.034	**0.203**
12	0.003	0.015	0.099	0.002	0.002	0.005	0.061	0.035	0.026	0.104	0.024	0.084	0.077	0.009	0.038	0.026	0.077	0.004	0.019	0.057	0.024	0.003	0.032	0.013	0.006	0.111	0.007	0.011	0.025	**0.214**
13	0.002	0.007	0.087	0.002	0.001	0.005	0.020	0.056	0.012	0.195	0.016	0.056	0.031	0.009	0.014	0.037	0.153	0.004	0.008	0.087	0.034	0.003	0.050	0.016	0.006	0.058	0.007	0.012	0.010	**0.341**
14	0.002	0.003	0.048	0.001	0.001	0.004	0.007	0.174	0.007	0.038	0.012	0.034	0.013	0.010	0.006	0.088	0.111	0.003	0.004	0.130	0.076	0.002	0.150	0.021	0.005	0.024	0.007	0.013	0.005	**0.288**
15	0.002	0.002	0.028	0.001	0.001	0.004	0.005	0.208	0.005	0.021	0.010	0.022	0.008	0.010	0.004	0.138	0.053	0.003	0.003	0.075	0.116	0.002	0.209	0.024	0.005	0.015	0.007	0.014	0.004	**0.289**
16	0.002	0.003	0.039	0.002	0.001	0.005	0.006	0.184	0.007	0.025	0.016	0.032	0.012	0.012	0.006	0.114	0.065	0.004	0.004	0.104	0.100	0.003	0.171	0.027	0.006	0.022	0.009	0.017	0.005	**0.288**
17	0.004	0.010	0.080	0.003	0.002	0.007	0.017	0.067	0.031	0.037	0.103	0.082	0.039	0.015	0.018	0.050	0.066	0.006	0.011	0.093	0.047	0.005	0.062	0.024	0.009	0.063	0.012	0.019	0.018	**0.646**
18	0.003	0.017	0.073	0.002	0.002	0.005	0.027	0.031	0.097	0.032	0.128	0.095	0.075	0.009	0.036	0.024	0.043	0.004	0.021	0.049	0.023	0.004	0.029	0.013	0.006	0.091	0.007	0.011	0.041	**0.306**
19	0.002	0.035	0.046	0.002	0.001	0.004	0.062	0.018	0.099	0.030	0.037	0.053	0.135	0.006	0.108	0.014	0.029	0.003	0.047	0.028	0.013	0.003	0.017	0.008	0.004	0.077	0.005	0.007	0.105	**0.294**
20	0.002	0.038	0.034	0.001	0.001	0.003	0.105	0.013	0.051	0.027	0.021	0.036	0.151	0.004	0.216	0.010	0.022	0.002	0.053	0.020	0.010	0.002	0.012	0.006	0.003	0.060	0.004	0.005	0.089	**0.291**
21	0.002	0.053	0.031	0.002	0.001	0.003	0.070	0.013	0.063	0.025	0.025	0.032	0.134	0.005	0.202	0.011	0.021	0.003	0.074	0.020	0.010	0.002	0.013	0.007	0.003	0.052	0.004	0.005	0.113	**0.291**
22	0.002	0.079	0.020	0.001	0.001	0.003	0.038	0.010	0.097	0.015	0.023	0.022	0.055	0.004	0.117	0.008	0.014	0.002	0.108	0.014	0.008	0.002	0.009	0.005	0.003	0.030	0.003	0.005	0.300	**0.427**
23	0.001	0.203	0.014	0.001	0.001	0.002	0.037	0.008	0.044	0.012	0.014	0.015	0.033	0.003	0.096	0.006	0.011	0.002	0.264	0.010	0.006	0.002	0.007	0.004	0.003	0.020	0.003	0.004	0.174	**0.350**
24	0.001	0.507	0.006	0.001	0.000	0.001	0.020	0.003	0.014	0.006	0.006	0.006	0.014	0.002	0.044	0.003	0.005	0.001	0.293	0.004	0.003	0.001	0.003	0.002	0.001	0.008	0.001	0.002	0.041	**0.320**
25	0.001	0.563	0.005	0.001	0.000	0.001	0.017	0.003	0.012	0.005	0.005	0.005	0.012	0.001	0.037	0.002	0.004	0.001	0.268	0.004	0.002	0.001	0.003	0.002	0.001	0.007	0.001	0.001	0.035	**0.314**
26	0.000	0.641	0.004	0.000	0.000	0.001	0.012	0.002	0.009	0.004	0.004	0.004	0.009	0.001	0.028	0.002	0.003	0.001	0.229	0.003	0.002	0.001	0.002	0.001	0.001	0.005	0.001	0.001	0.028	**0.305**
27	0.000	0.702	0.003	0.000	0.000	0.001	0.010	0.002	0.007	0.003	0.003	0.003	0.007	0.001	0.022	0.002	0.003	0.001	0.195	0.002	0.001	0.000	0.002	0.001	0.001	0.004	0.001	0.001	0.021	**0.298**
28	0.000	0.750	0.003	0.000	0.000	0.000	0.009	0.002	0.006	0.002	0.002	0.003	0.006	0.001	0.019	0.001	0.002	0.000	0.164	0.002	0.001	0.000	0.001	0.001	0.001	0.004	0.001	0.001	0.018	**0.292**
29	0.000	0.740	0.003	0.000	0.000	0.000	0.009	0.002	0.006	0.003	0.003	0.003	0.006	0.001	0.019	0.001	0.002	0.000	0.171	0.002	0.001	0.000	0.001	0.001	0.001	0.004	0.001	0.001	0.019	**0.293**
30	0.000	0.879	0.001	0.000	0.000	0.000	0.004	0.001	0.003	0.001	0.001	0.001	0.003	0.000	0.009	0.001	0.001	0.000	0.082	0.001	0.001	0.000	0.001	0.000	0.000	0.002	0.000	0.000	0.008	**0.278**
31	0.000	0.881	0.001	0.000	0.000	0.000	0.004	0.001	0.003	0.001	0.001	0.001	0.003	0.000	0.008	0.001	0.001	0.000	0.081	0.001	0.001	0.000	0.001	0.000	0.000	0.002	0.000	0.000	0.008	**0.278**

6.3 EVALUATION

Evaluation is a critical step to assess the confidence of the proposed fuzzy risk assessment model. The model proposed in this chapter has been evaluated using sensitivity analysis through a usage-based insurance risk assessment case study in the big data

environment. We closely monitored all the used parameters of the models and assessed their impact on the risk score provided by the fuzzy risk assessment model.

The result of this step shows the confidence of the proposed model. The proposed model is based on a fuzzy clustering algorithm which extracts various criteria for decision making and the fuzzy risk matrix which is used for risk score calculation.

To validate the system proposed in this study, a sensitivity analysis is developed on various trips with different risk levels. The sensitivity analysis shows how much the model result can be affected by the uncertainty in input parameters. A partial validation using sensitivity analysis on these drivers is proposed with the following conditions:

- Condition 1: selecting top k similar driving patterns, where all possible values for k have been considered.
- Condition 2: defining three different aggregation strategies, namely optimistic, pessimistic, and neutral.

Three trips are selected with different risk levels and their risk is assessed according to the proposed fuzzy risk assessment model and the parameters are considered to assess the confidence of the proposed model.

6.3.1 SENSITIVITY ANALYSIS

Three trips are selected with three risk levels, namely Trips A, B, and C, which are low, medium and high risk respectively. The risk score of these trips is calculated using the proposed fuzzy risk assessment model. We tested the models' output by changing the input parameters. Figures 6-3 to 6-5 illustrate the risk score of these trips by considering three

different strategies. In these figures, the x-axis shows the number of k, and the y-axis shows the risk score.

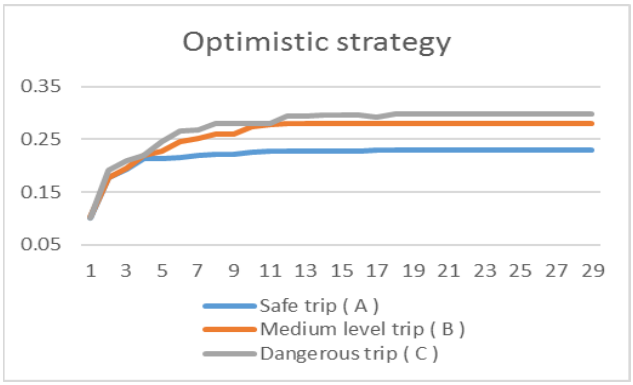

Figure 6-3 - Sensitivity Analysis Optimistic Strategy Risk

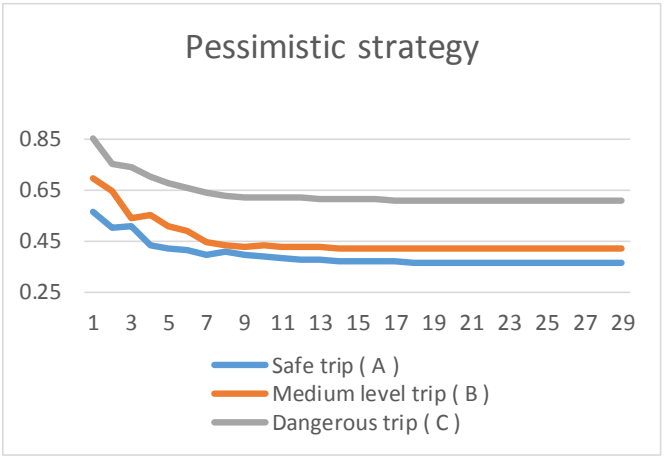

Figure 6-4 - Sensitivity Analysis Pessimistic Strategy Risk

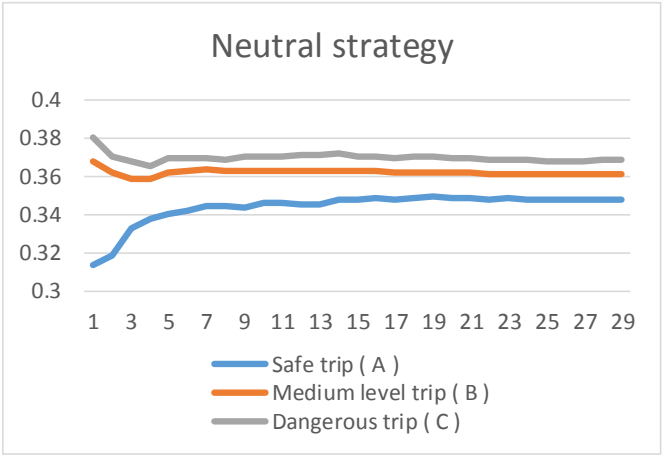

Figure 6-5 - Sensitivity Analysis Neutral Strategy Risk

An examination of this model in light of the three strategies using all possible values for k reveals that the calculated risk score for a dangerous trip is larger than a medium-level trip, and a safe trip has a lower risk level in comparison to the others. In addition, by increasing the value of k, the risk score becomes stable with less variations. For the optimistic strategy, the risk score has a very low domain and the difference between the minimum and maximum value is low, while for the pessimistic strategy, the domain is larger and the extracted value is more reliable.

The optimistic strategy in Figure 6-3 shows that by increasing the number of k, the risk value is increasing but it is more stable with less variation. On the other hand, the pessimistic strategy has different behaviour, and by increasing the value of k, it is decreasing and the risk value is not changed significantly. The neutral strategy behaviour is completely different from the others, and the risk score for a safe trip is increasing, while the score for other trips is decreasing. The sensitivity analysis shows that changing the parameters of the model does not have any impact on the final result because trip C is always dangerous regardless of the input parameters for the possible value for k in all strategies.

6.4 SUMMARY

In this chapter, the fuzzy risk assessment component with its sub-models are explained. We proposed a new fuzzy risk assessment methodology for the mobile telematics environment. We introduced an autonomous fuzzy decision support system using the capabilities of artificial intelligence and machine learning algorithms to extract decision-making criteria and the risk factors from mobile telematics data automatically. The proposed fuzzy risk assessment system extracted different patterns from big data streams automatically, and we proposed the fuzzy expert system to assess the risk of extracted criteria in an uncertain situation. Finally, we evaluated the performance of our proposed framework using a mobile telematics case study. Our results using sensitivity analysis show that the proposed decision

support system is consistent across various input parameters and optimistic, pessimistic and neural strategies.

Chapter 7:

MISSING DATA IMPUTATION

7.1 INTRODUCTION

Mobile telematics has been used in a number of road safety applications (Zhao 2002), intelligent transportation systems (Zhao 2000), and usage-based insurance (Bowne et al. 2013), but applying this technology in real-world businesses is problematic. According to the experts' opinion, risk assessment using driving style data is only one part of applying mobile telematics in real-world problems. There is a huge data gap in business-related data in mobile telematics. In particular, for the UBI industry, the premium calculation not only depends on the driving style risk score but also relates to other variables such as age range, gender, suburb in which they live, and etc. Mobile telematics is unable to fill these data gaps alone, thus missing data imputation is proposed to help decision makers to prepare an accurate calculation for customer premium in insurance.

The main purpose of the missing data imputation component is to decrease the rate of missing data in mobile telematics to help decision makers estimate the null fields in mobile telematics. For example, the gender of users is one of these unknown features, and in particular, the gender of drivers is an important feature for premium calculation. To the best

of our knowledge, mobile telematics has so far been unable to reliably answer this question: what is the gender of the driver behind the wheel, male or female?

The missing data imputation component is a new classification algorithm, which is proposed for the first time in this study. The proposed algorithm is the Choquet fuzzy integral vertical bagging classifier. The proposed algorithm is new and we have applied this algorithm to detect gender from mobile telematics data.

Figure 7-1 illustrates the proposed component. This component extracts the gender of drivers according to their driving styles to give some insights to decision makers for their decision making.

In this chapter, first we explain the fuzzy integral in Section 7.2. Section 7.3 describes the implementation process for data preparation. The Choquet fuzzy integral vertical bagging classifier is introduced in Section 7.4. Finally, Section 7.5 presents the experiment results of the proposed classifier.

7.2 DATA PREPARATION

The data preparation component has been explained in the previous chapters, but as the implementation process for labelled data is different from unlabelled data, this section provides complementary explanations regarding the data preparation process for the missing data imputation component. Table 7-1 introduces the driving characteristics stream data used by this component.

Table 7-1- Stream Data Introduction

Name of feature	Description
Speed	The value of the instantaneous velocity of the vehicle stored by a smartphone
Acceleration (x,y)	Two different stream data of a car's acceleration over x and y axis.
Yaw rate	The value of the vehicle's angular speed around its vertical axis.
Pitch rate	The lateral motion of a car is called the pitch rate. The pitch rate value shows the up or down forward tilt of the vehicle.
Roll rate	The longitudinal axis movement of the car shows the characteristics of the road.
GPS heading	The compass direction measured in degrees from north.

The sample rate of the input data is equal to 15 Hz. This sample rate is high for our methodology, thus we downsampled the data to one sample per second because our investigations revealed that one sample per second produces the best information for our analytics. As explained above, a windowing procedure is developed to select a driving time window, then the time windows are summarized using statistical features from the stream data for each time window. 14 statistical features including minimum, maximum, mean, median, first and third quantile, standard deviation, average absolute deviation, skewness, entropy, kurtosis, auto-correlation, zero crossing, and energy are calculated for each stream data feature. We calculated these features for all seven stream data, extracting 7×14 features for each time window.

In addition to feature extraction, feature selection techniques are performed on the extracted features to find the features with the highest correlation with class labels. Features with low variation and low correlation are removed from the data using the data preparation component.

7.3 CHOQUET FUZZY INTEGRAL VERTICAL BAGGING CLASSIFIER

The vertical bagging model is similar to traditional bagging, the difference being in the method of creating sub-models. In traditional bagging, sub-models are generated from sub-samples of data with the same attributes. In contrast, the sub-models in the vertical bagging models are trained by various combinations of predictive attributes with similar sample data (Zhang et al. 2010). In this chapter, we combine multiple random forest classifiers with the Choquet fuzzy integral to propose the Choquet fuzzy integral vertical bagging classifier. The maximum number of features (F) and the maximum number of iterations are the input parameters for training the algorithm.

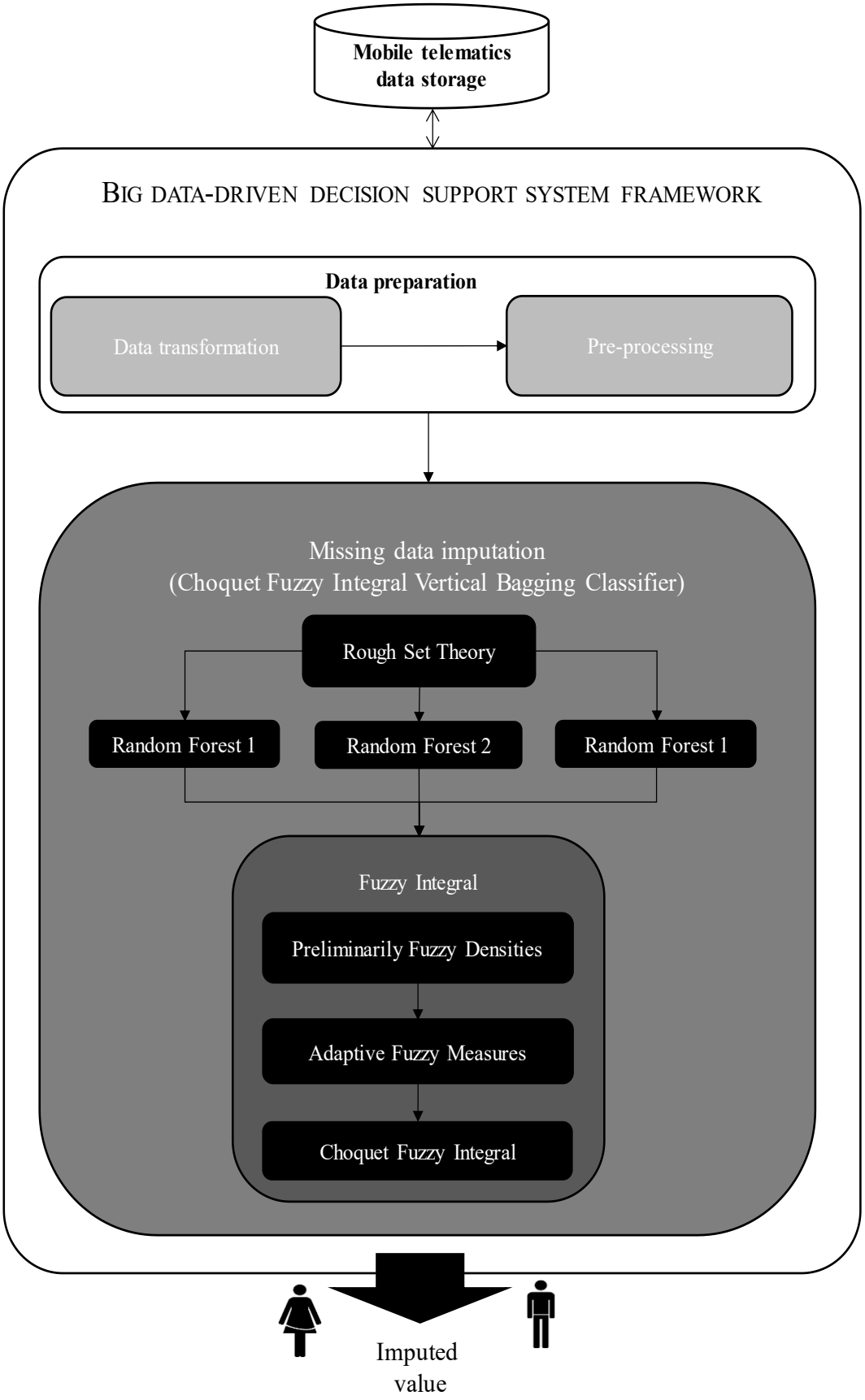

Figure 7-1- Missing Imputation Component (Choquet Fuzzy Integral Vertical Bagging)

Using rough set theory (Pawlak 1998), k training subsets Tr_1, Tr_2.... Tr_k were generated from the input training data. The maximum number of features (F) is equal to the total number of features in each Tr subset and k is equal to the maximum number of iterations. Random forest (RF) (Breiman 2001) classifiers were trained for each training subset. All RF models were sorted according to their performance in the training dataset, and the top three models were selected. In the next step, the top random forest classifiers were merged using the Choquet fuzzy integral which achieves outstanding performance in merging different classifiers (Namvar & Naderpour 2018).

7.3.1 FUZZY INTEGRAL

Classification results are not usually precise or certain, so fuzzy theory is useful for merging different classifiers into one prediction result. Fuzzy integral methods such as Sugeno and Choquet are popular and practical methods which have been used in a wide range of domains including mathematics, economics, machine learning and pattern recognition (Wang et al. 2015).

Although both of these integral methods are fuzzy and popular, Choquet fuzzy integrals have been more widely applied than Sugeno integrals (Krishnan, Kasim & Bakar 2015). A Choquet integral is an aggregation method that simultaneously considers the importance of a classifier and its interaction with other classifiers (Li, Wang & Chen 2015). It relies on the concept of fuzzy measures first introduced by Sugeno (1974). The definitions of Choquet integrals and fuzzy measures according to (Murofushi & Sugeno 1989) are as follows.

Assume X is a set of classifiers and the power of X is denoted by P(X).

Definition 5–1: The fuzzy measure of X is a set function $g: P(X) \rightarrow [0,1]$. This function satisfies the following conditions:

1) The boundary of g is : $g(\phi) = 0$, $g(X) = 1$

2) For each A, $B \in P(X)$ and $A \subset B$ then $g(A) \leq g(B)$

where $g(k)$ is the grade of the subjective importance of the classifier set k. The fuzzy singleton measure values for each classifier are $g(x_i) = g^i$ and are commonly called densities. Not only must the worth of each singleton be calculated, but also the value of function g for any combination of classifiers. The Sugeno λ-measure and fuzzy densities are used to calculate the fuzzy measure of any combination of classifiers. This measure is defined by the values of the fuzzy densities. The λ-measure has the following additional property:

$$\begin{cases} g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B) \\ \forall A, B \in P(X), A \cap B = \phi \end{cases} \quad (7\text{-}1)$$

where λ can be calculated by Eq. 7-2.

$$\lambda + 1 = \prod_{i=1}^{n}(1 + \lambda g^i), \lambda > -1 \quad (7\text{-}2)$$

Definition 5-2: g is the fuzzy measure of $X = \{x_1, x_2, \ldots, x_n\}$. Eq. 7-3 shows the Choquet integral function of $f: X \to R$ and its relation to g:

$$C_g(f) = \sum_{i=1}^{n} f_i \left[g(A_i) - g(A_{i-1}) \right] \quad (7\text{-}3)$$

A permutation of X is indicated by (i), and $f(x_{(1)}) \le f(x_{(2)}) \le \cdots \le f(x_{(n)})$ also $A_i = \{x_{(i)}, x_{(i+1)}, \ldots, x_{(n)}\}$, $A_0 = \phi$.

The prediction result of classifier x_i, is denoted by f_i, and $[g(A_i) - g(A_{i-1})]$ depicts the relative importance of the classifier x_i. The fuzzy integral of f with respect to g is the integration result.

7.3.2 PRELIMINARY FUZZY DENSITIES:

In the previous step, the top three random forest classifiers with the highest performance in the training data were selected and the confusion matrix for each classifier was determined. Eq. 7-3 is used to calculate the Choquet fuzzy integral on the fuzzy measures (g), reflecting the importance of each classifier and classifier combinations. Equation 7-1 is used to calculate the value of any combination of classifiers. These parameters are crucial in

the fuzzy integral vertical bagging classifier, and they play an essential role in the practical application of fuzzy integral in this algorithm and information fusion.

The confusion matrix of the i-th classifier is defined as

$$CM_i = \begin{bmatrix} n_{11}^i & \cdots & n_{1M}^i \\ \vdots & \ddots & \vdots \\ n_{M1}^i & \cdots & n_{MM}^i \end{bmatrix} \quad i = 1,2,\dots,P \tag{7-4}$$

where $j_1 = j_2$, $n_{j_1 j_2}^i$ depicts the number of samples in class c_{j1} which are correctly classified as c_{j1} by the i-th classifier. On the other hand, $j_1 \neq j_2$, $n_{j_1 j_2}^i$ represents the number of samples with class label c_{j1}, but they have been misclassified as c_{j2} by classifier i. Therefore, the probability of items which have been correctly classified is calculated by Eq. 7-5 for each classifier and class label.

$$p_{j1j}^i = p\left(s_k \in c_{j1} \middle| E_i(s_k) = c_j\right) = \frac{n_{j1j}^i}{\sum_{j=1}^{M} n_{j1j}^i} \tag{7-5}$$

$$(j_1 = 1,2,\dots,M; j = 1,2,\dots,M)$$

where i is the i-th classifier, M is the number of classes, and the probability matrix is

$$PM_i = \begin{bmatrix} p_{11}^i & \cdots & p_{1M}^i \\ \vdots & \ddots & \vdots \\ p_{M1}^i & \cdots & p_{MM}^i \end{bmatrix} i = 1,2,\dots,P \tag{7-6}$$

The p_{jj}^i elements in PM_i represent the percentage of items which are classified correctly by the E_i classifier. Let $g_j^i = p_{jj}^i$, then g_j^i depicts the preliminary fuzzy density value for the j-th class with respect to the i-th classifier. The fuzzy density value of each classifier and the different class labels is the output of this step.

7.3.3 ADAPTIVE FUZZY MEASURES:

The classification results of random forest differ according to the feature set. Some classifiers may have better performance than others, and one classifier may be more robust than others for classifying certain types of items or classes. Merging classifiers by a voting strategy or by assigning an equal value to all classifiers is therefore not an efficient approach,

and the fuzzy density measure (g_j^i) needs to be properly adjusted by considering all classifiers and all classes.

After the correct classification rates and misclassification errors within the classifiers have been calculated, the values are used to update the fuzzy densities. The fuzzy densities are then updated by considering the pairwise proportion of wrongly classified items between the selected classifiers and the others. The fuzzy density parameters can be updated using Eq. 7-7 (Pham 2002):

$$g_j^{*i} = g_j^i * \left(\prod_m \delta_j^{i/m}\right)^{w1} * \left(\prod_m \gamma_j^{i/m}\right)^{w2} \quad (7\text{-}7)$$

where g_j^{*i} is the updated fuzzy density for the i-th classifier for class j; $\left\{\delta_j^{i/m}\right\}, 0 < \delta_j^{i/m} < 1$, and $\left\{\gamma_j^{i/m}\right\}, 0 < \gamma_j^{i/m} < 1$ are the sets of updated parameters. g_j^{*i} is calculated for all classes. Each set of updated parameters could have a different impact on the final outcome, so to add flexibility to the final result, w_1 and w_2 in Eq. 7-7 are used in the updating process.

$\delta^{i/m}$ is used to update the initial fuzzy density when the output of two different classifiers do not have the same result. The initial fuzzy density of the classifier will be decreased by increasing the number of misclassified objects, while the correctly classified items will increase the power of the classifier using Eq. 7-8.

$$\delta_j^{i/m} = f(x) = \begin{cases} 1 & , k_i/i = k_2/m \\ \dfrac{p_{j/i,j/i}^i - p_{k/i,j/m}^i}{p_{j/i,j/i}^i} & , k_1/i \neq k_2/m \end{cases} \quad (7\text{-}8)$$

where k_1/i, and k_2/m show that class k_1 is given by classifier $\mathrm{E_i}$ and class k_2 is given by classifier $\mathrm{E_m}$. When $k_1/\mathrm{i} = k_2/\mathrm{m}$, this means that two classifiers have identified samples in similar classes. $k_1/\mathrm{i} \neq k_2/\mathrm{m}$ means that two different classifiers have categorized a sample into two different classes. One sample may be correctly classified by $\mathrm{E_i}$ in class C1 but misclassified by classifier E_m. The proportion of objects correctly classified by E_i for class j is

depicted by $P^{i}_{j/i,j/i}$, but the number of items correctly classified by other classifiers is $p^{i}_{k/i,j/m}$. The training dataset is used to obtain both $P^{i}_{j/i,j/i}$ and $p^{i}_{k/i,j/m}$. Once the number of items misclassified by E_i have increased, the corresponding fuzzy density measure of the E_i classifier will be decreased.

The reason for updating the parameters $\gamma_j^{i/m}$ is that the initial fuzzy density of a classifier should be reduced when error E_i is more than classifier E_m, but the fuzzy density value is not changed if classifier E_i has the same or fewer mistakes than classifier E_m. This concept is developed by Eq. 7-9.

$$\gamma_j^{i/m} = \begin{cases} 1 & : p^{i}_{k/i,q/m} \leq p^{m}_{k/i,q/m} \\ \frac{p^{m}_{k/i,q/m}}{p^{i}_{k/i,q/m}} & : p^{i}_{k/i,q/m} > p^{m}_{k/i,q/m} \\ \varepsilon & : \quad p^{m}_{k/i,q/m} = 0 \end{cases} \tag{7-9}$$

where ε is a very small value which prevents $\gamma_j^{i/m}$ from being zero.

Adjusted fuzzy density is the output of this step, which updates the importance of each classifier in the training dataset by considering its performance in the training dataset by classifying items correctly or misclassifying them. The fuzzy measures are calculated to depict the weight of all classification algorithm that we want to merge them together in a fuzzy concept. Therefore, these measures are defined to provide a more robust weight for models by considering models accuracy and confusion matrix.

7.3.4 CHOQUET FUZZY INTEGRAL

The performance of each classifier in the vertical bagging RF is variant. Some RF classifiers have insufficient power to predict the result correctly, while another RF model may achieve excellent performance on the same samples. We propose the Choquet fuzzy integral vertical bagging random forest to take advantage of different random forest models. The Choquet fuzzy integral fuses the results of multiple classifiers and provides a robust classifier with more consistent results.

Suppose in a sample data space S, data are divided into two classes by a classifier (E). A classifier index is specified by ($i = 1, \dots, P$); j is the class index ($j = 1, \dots, M$); and k is the instance index($k = 1, \dots, N$). For the k-th sample, the prediction result by the i-th classifier is $[h_{i1}(k), h_{i2}(k), \dots, h_{iM}(k)]$ where $h_{ij}(k)$ is the probability result of the ith classifier, which shows the probability of k-th data belonging to class j. $[h_{1j}(k), h_{2j}(k), \dots, h_{Pj}(k)]^T$ is defined as $h_j(s_k)$ which can be interpreted as :

$h_j: S \rightarrow [0,1]$,$h_j(s_k) = [h_{1j}(k), h_{2j}(k), \dots, h_{Pj}(k)]^T$ for sample s_k, we obtain a value for $h_j(s_k)$ as degree of support provided by each classifier with respect to the j-th class for sample s_k.

In addition to $h_j(s_k)$, the Choquet fuzzy integral operates on the fuzzy measures (g). Fuzzy measures include fuzzy densities and the fuzzy measure of any combination of classifiers, which are calculated by Eqs. 7-8 and 7-1.

By calculating the Choquet integral of $h_j(s_k)$, g, we can provide the degree of support given by the ensemble classifier with respect to the j-th class for sample s_k. The output class c_j for sample s_k is the class with the largest integral value:

$$c_j = arg(\max_{1 \le l \le M} \int h_l(s_k) dg) \qquad (7\text{-}10)$$

A summary of the Choquet fuzzy integral vertical bagging classifier is as follows:

Algorithm 7-1 - Choquet Fuzzy Integral Vertical Bagging (CFIVB) Classifier

Input: Data, maximum number of features (F), maximum iteration (K)

Output: imputed value

1) ***Generate a list of important features in feature engineering step***
2) ***For k in [1,2, … K] as iteration:***
   - ✓ ***Train RF with F number of random features***
   - ✓ ***Validate RF with training dataset***
   - ✓ ***End for k***
3) ***Select top three RF models for fusion with training dataset***
4) ***Construct the confusion matrix for each selected classifier (Eq. 7-4), with training dataset***
5) ***For each j in [1,2] as [male/female]:***
   - ✓ ***For each i in [1,2,3] as top RF classifiers:***
     - ***Calculate initial fuzzy densities by Eq.7-5.***
     - ***Update parameter*** $[\delta_j^{i/m}]$ ***by Eq. 7-8.***

- *Update parameter* $[\gamma_j^{i/m}]$ *by Eq.7-9.*
- *Update the initial fuzzy densities by Eq.7- 7.*
- *End for i*

✓ *Compute the* g_λ*- fuzzy measures with updated fuzzy densities*
✓ *Compute the fuzzy integral each class with Eq. 7-3.*
✓ *End for j*

6) Use Eq. 7-10 to detect the gender of a driver.

7.4 EXPERIMENT RESULTS

The primary goal of this section is to examine the prediction results of the methodology for gender detection from smartphone-generated data. We used almost 1 GB data containing the anonymized driving behaviour of 301 unique drivers. These data were collected by a usage-based insurance company in real-world conditions. The dataset consists of the streamed data of 408 trips. Each trip contains at least 15 minutes of driving data from 301 unique drivers, some of whom feature in more than one trip. The number of male drivers is 161 and the number of female drivers is 140. A brief description of the final dataset is summarized in Table 7-2.

Table 7-2- Data Description

Number of trips	Number of unique drivers	Total driving distance	Total driving time	Male	Female
408	301	9898 km	202 hours	161	140

The experiment started by decreasing the stream data sample rate. We found that the best sample rate in our data for gender detection was one sample per second, which is the average of all 15 samples in one second. After decreasing the sample rate, we developed a windowing process. We segmented the driving characteristics into time windows which were equal to 512 seconds, then extracted all the proposed features listed in Section III for each time window. Features with very low variance were removed, and we developed a correlation analysis between the extracted features and gender of the driver to find the features of highest importance. We selected features which had a correlation greater than 0.1. The correlation analysis report is depicted in Table 7-3.

Table 7-3- Correlation Analysis Results

Feature name	Correlation
Speed mean	-0.19708638
Speed_Q3	-0.19611402
Speed energy	-0.19610685
Pitch_rate_Q1	-0.19027125
Speed median	-0.1869981
Speed_Q1	-0.18227661
Pitch rate standard deviation	0.17657519
Speed max	-0.17616991
Pitch rate energy	0.1666721
Pitch rate kurtosis	-0.16354984
Pitch rate average absolute deviation	0.1609966
Pitch_rate_Q3	0.13283702
Acceleration lon zero-crossing	-0.11766974
Speed skewness	0.11405972
Yaw rate zero-crossing	-0.10055166

After selecting the most valuable attributes from the extracted features, we had a clean data source ready for analytics containing 15 features of 2048 windows for 1119 males and 929 females. We used these data to validate our proposed classifier. We developed a comparison analysis with three base classifiers: random forest, gradient boosting classifier, and logistic regression. Our model was trained by setting the maximum number of iterations to 100; the maximum number of features in rough set theory is 10. In addition to the input parameters for the vertical bagging classifier, we defined the exponential weights for w1 and w2 in Eq. 7-7 as W1=0.9 , W2=0.6 (Namvar & Naderpour 2018). The value of ε in Eq. 7-9 was set to 0.0001.

In order to assess the performance of each clustering methodology, we used a well-known cross-validation methodology. This kind of validation helps to prevent over-fitting and under-fitting in the model performance evaluation (Malekipirbazari & Aksakalli 2015). In this research, we applied 5-fold cross-validation and we divided the total number of records which are 394,833 into five sets of data. In the first step, we selected the first 4 slices

of data for training the model and the last set of records for testing the model. In the next iteration, we selected the second slice for testing the model and the remaining slices for model training. This procedure is repeated 5 times. This process is illustrated in Figure 7-3.

To calculate the final performance of the model, different statistical measures are calculated based on the performance of each fold.

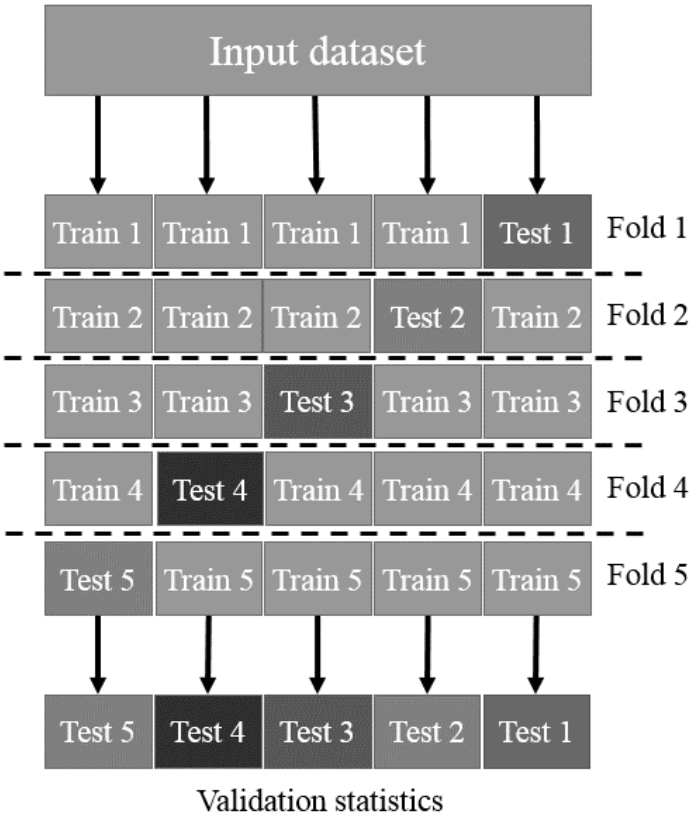

Figure 7-2 - 5-Fold Cross-Validation

5-fold cross-validation is developed to evaluate the performance of the proposed model. Accuracy and area under the curve (AUC) are two performance measures in this research. AUC shows the area under the ROC curve, and the accuracy score reflects the proportion of correct items in relation to all items. We compared our model performance with three other classifiers and the results are depicted in Table 7-4. We selected these three algorithms because of the following reasons: Logistic Regression is the most simple classification model. Our model is the improved version of random forest, so we should

compare its performance with RF, and Gradient Boosting is one of the state of the art algorithm, which is mostly used recently.

Table 7-4- Results

	AUC	Accuracy
Our model	**72.44**	**71.67**
Random Forest	68.23	64.16
Logistic Regression	60.32	55.85
Gradient Boosting Classifier	66.16	62.51

The results in Table 7-4 indicate that the Choquet fuzzy integral vertical bagging classifier achieves the best performance in terms of accuracy and AUC compared to the selected alternative algorithms. These results show that the final model not only improves the performance of the base classifier, random forest, but it also achieves better performance than logistic regression and gradient boosting.

In terms of accuracy, the proposed model achieves the best result for detecting gender. The accuracy score of the Choquet fuzzy integral vertical bagging classifier is 71.67, which is significantly higher than that of the classifiers selected for comparison. In addition to the accuracy score, we calculated the AUC score for each classifier. Our proposed model achieved 72.44, which is higher than the other three comparison models. The logistic regression classifier returned the worst results for both accuracy and AUC score for gender detection compared to the other methods.

To gain a comprehensive view of the performance of our model and to prevent overfitting or underfitting, we conducted 5-fold cross-validation. Figure 7-2 shows the performance of all four models for each run. The results show that our model achieves the highest performance in all folds.

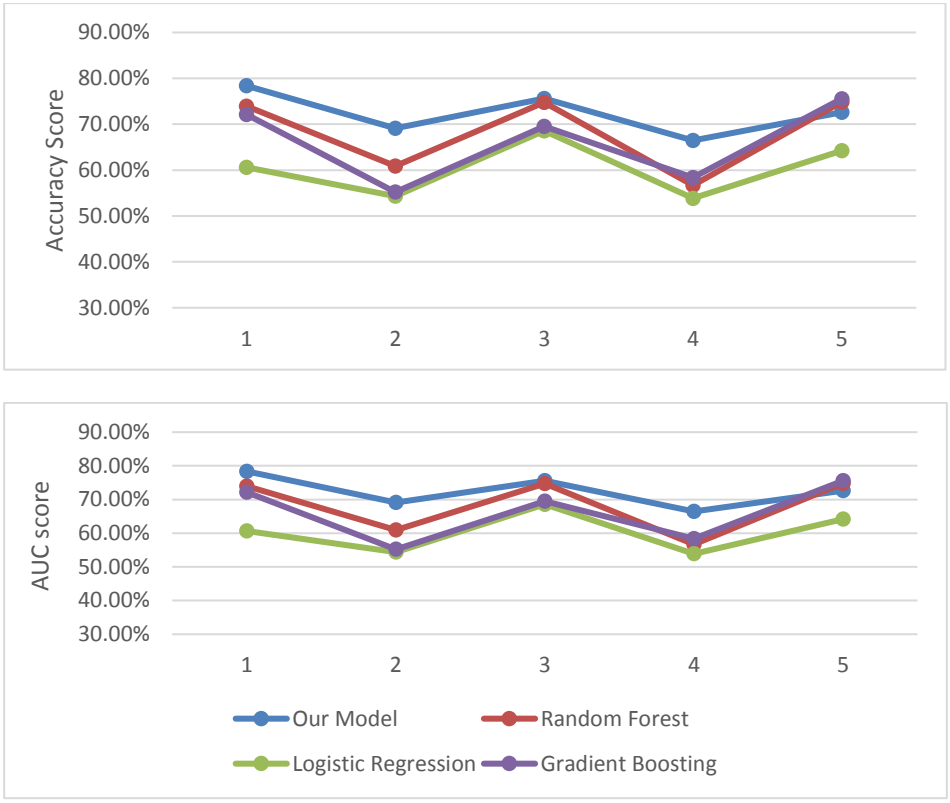

Figure 7-3 - Choquet Fuzzy Integral Vertical Bagging

7.5 SUMMARY

The application of mobile telematics for business is not limited to driving style risk assessment as the risk score is only one variable, hence decision makers need more variables to make a suitable decision in different domains. For example, in the UBI industry, demographic variables such as gender play an important role for the premium calculation process, but mobile telematics is unable to provide this information when these variables are not declared by the users. In this chapter, a new machine learning model is proposed to detect the gender of drivers from driving style data. The proposed model plays the role of missing data imputation in the proposed decision support system in this study.

A novel Choquet fuzzy integral vertical bagging classifier is proposed to detect the gender of the driver from smartphone-generated data. In addition, to the best of our

knowledge, this is the first model that uses smartphone data to detect the gender of a driver. The model uses randomly generated features to create several random forest models. Then, the Choquet fuzzy integral aggregates the results of the top classifiers to find the final result. The validation results show that the vertical bagging classifier with Choquet fuzzy integral achieves the best accuracy and AUC score for gender detection compared to the other classifiers.

Chapter 8:

CONCLUSION AND FUTURE WORK

8.1 CONCLUSIONS

This research has been motivated by the fact that mobile telematics devices can collect a huge amount of data which reveals human behaviour. These devices can provide useful applications for decision making. Thus, proposing a decision support system for mobile telematics using the capabilities of advanced analytical techniques and machine learning algorithms can help decision makers make better decisions in the mobile telematics environment.

The proposed DSS has superior performance in comparison with traditional forms of decision support systems. Traditional decision makers typically overlook the relationships among the involved criteria and are not able to identify the imprecise reasoning embedded in their criteria. Rule-based risk assessment methods are also time-consuming and challenging as they are heavily reliant on experts' knowledge, which also make them very subjective. The system proposed in this thesis relies upon new advancements in artificial intelligence and machine learning to provide new opportunities for mobile telematics data analytics to use supervised and unsupervised learning and automatic rule extraction algorithms to a build decision support system for risk assessment in mobile telematics. In this

study, we proposed an analytical platform to analyze mobile telematics data for risk assessment and also a novel supervised learning algorithm is proposed to apply to mobile telematics analytics once the labelled data is available.

The availability of labelled data is a major concern for modelling a supervised learning model in mobile telematics. The neural networks rule extraction, random forest classification algorithm, and deep learning models all achieve outstanding performance once the labelled data is available, but these models are not suitable to help analytical experts in the absence of true labels.

Because of the aforementioned weaknesses, in addition to the proposed supervised learning model, we introduced an unsupervised learning model for mobile telematics pattern recognition to understand the hidden patterns in large and complex datasets. These models try to cluster unlabeled data into groups. In addition, we provided a fuzzy risk assessment method to assess the risk of drivers by combining the capabilities of fuzzy inference systems and advanced machine learning techniques.

The proposed system can provide an outstanding contribution to mobile telematics data analytics and the big data environment. However, using this model in the practical sector needs a particular big data platform with a cloud computing feature, and all the capabilities of the proposed system will be available by implementing it on distributed computing platforms.

One of the most important limitation of this study is availability of labelled data which is suitable for research domain to assess the performance of the models using supervised techniques and model. The labelled data is rare in driving style analysis techniques, hence we proposed unsupervised assessment techniques.

This research makes the following main contributions:

- First, a decision support system is proposed to analyze driving behaviour for decision making using various advanced analytical techniques, including fuzzy logic, artificial intelligence and advanced analytics. The proposed framework

has four major components: 1) data preparation, 2) driving style pattern recognition, 3) fuzzy risk assessment, and 4) missing data imputation.

- Second, a risk assessment methodology which learns from big data is proposed. Most of the studies in the field of big data risk assessment only focus on supervised learning techniques, however this study aims to contribute to this growing area of research by exploring unlabeled data for automatic risk assessment using unsupervised learning and fuzzy logic. While the framework is proposed for risk assessment in mobile telematics risk assessment, it is still general and can also be used for other multi-criteria decision making.
- Third, an unsupervised learning framework is proposed to extract different risk factors for decision making automatically. The lack of labeled data is a fundamental challenge for all machine learning and artificial intelligence algorithms. Thus, in this study, we deal with the current challenges of unlabeled data in the big data domain which still remains an open problem. The proposed algorithm is a two-step clustering algorithm incorporating a self-organizing map (SOM) and fuzzy clustering. The SOM reduces the complexity of the data and fuzzy clustering categorizes the input dataset.
- Fourth, a fuzzy decision support system using the capability of a fuzzy inference system to handle uncertainty and the lack of confidence in the noisy data is proposed. Big data generated by digital devices, social media, and sensor technologies contain noise, and analyzing them can impair the analytical result, hence by increasing the volume, velocity, and variety of data, the final result will be more sensitive to noise. Therefore, we use fuzzy logic to decrease the uncertainty and lack of confidence in the data.
- Fifth, we evaluated the proposed framework using sensitivity analysis on big data collected by smartphones to assess the risk of car journeys for usage-based

insurance. We evaluate the results using sensitivity analysis to show the confidence of the final results.

- Sixth, an empirical assessment is made of five partitive clustering algorithms on SOM and deep auto-encoder results by the Davis Boulding and Calinski Harabasz indexes as well as execution time. The performance results show that a self-organizing map and k-means clustering are the best combination of the two-stage clustering of similar driving patterns into a set of driving behaviours.
- Seventh, an approach to identify the driving events with the highest rate of change according to three key characteristics, velocity, x-axis acceleration, and y-axis acceleration, using relative unconstrained least-squares importance fitting (RuLSIF) is proposed as data preparation.
- Eighth, a Choquet fuzzy vertical bagging classification algorithm is proposed to extract driving patterns from big data automatically using labelled data. The algorithm is completely new and combines the capability of the random forest algorithm with the Choquet fuzzy integral. Moreover, we proposed a new application for this algorithm to detect the gender of drivers from mobile telematics driving patterns.

8.2 FUTURE WORKS

The future directions of this research can be summarized from the following perspectives:

- One of the major contributions of this research is extracting driving behaviour from mobile telematics data using advanced analytics and unsupervised learning methods. We extracted 29 unique clusters of driving behaviours. These clusters can be used as a starting point for researchers in the field of transportation.

- Proposing a supervised learning algorithm for driving identification, risk assessment and behaviour detection is the next step for this research. The lack of labelled data was a significant challenge in this project. The labels provided in this research can be applied to other studies to provide new classification algorithms. Moreover, one of the most vexing challenges with using machine learning techniques for driving style analytics is the lack of labeled data. Thus, researchers in the field of transportation and road safety could also use this framework to label unlabelled datasets. Once labelled, the data could be used with a supervised learning technique with most state-of-the-art machine learning algorithms for various applications.

- The Choquet fuzzy vertical bagging classifier is a new algorithm with many opportunities to apply it in different domains. This algorithm is proposed for the first time and it has a wide range of applications in other areas of research. The performance of the Choquet fuzzy integral vertical bagging classifier is much higher than other classification algorithm, so the contribution is great.

- Many of the traditional decision support systems are proposed based the experts' opinion and they are very expensive and time-consuming. This research proposed an autonomous big data-driven decision support system for risk assessment. We applied this methodology for risk assessment in the mobile telematics environment. Other researchers can apply this methodology in other domains such as financial risk assessment in stock market time series data and transaction analysis for fraud detection.

- In addition to the practical application of the proposed models in other domains, the Choquet fuzzy integral vertical bagging classifier has great potential in improving other techniques. The performance of the algorithm can be improved using automatic feature extraction such as deep auto-

encoder and convolutional neural networks algorithms. Moreover, performance tuning and sensitivity analysis is another goal to evaluate the proposed algorithm. For example, the effects of variations in input variables, such as the maximum number of features and the maximum number of iterations, on the model's performance can be evaluated to assess model sensitivity.

- Fuzzy measures proposed in this study has great performance for merging classification results together. This concept and choquet fuzzy can be applied in other bagging and boosting classifiers such as random forest to improve the performance of them.
- XAI is one of the key areas of work being looked at in recent years. Working towards XAI is another future study for this research in trying to make the black box models in a more white box based approach.

BIBLIOGRAPHY

(ISO), I.S.O. 2011-12, *Road vehicles — Vehicle dynamics and road-holding ability — Vocabulary*.

AF Wåhlberg, A.E. 2004, 'The stability of driver acceleration behavior, and a replication of its relation to bus accidents', *Accident Analysis & Prevention*, vol. 36, no. 1, pp. 83-92.

Ahmadabadi, A.A. & Heravi, G. 2019, 'Risk assessment framework of PPP-megaprojects focusing on risk interaction and project success', *Transportation Research Part A: Policy and Practice*, vol. 124, pp. 169-88.

Aminikhanghahi, S., Wang, T. & Cook, D.J. 2018, 'Real-time change point detection with application to smart home time series data', *IEEE Transactions on Knowledge and Data Engineering*.

An, C., Zhu, R., Wang, X., Long, Y., Lu, Y., Chen, Y. & Zhong, H. 2020, 'The correlation analysis of RCPs impeller geometrical parameters and optimization in coast-down process', *Annals of Nuclear Energy*, vol. 142, p. 107283.

Arthur, D. & Vassilvitskii, S. 2006, *k-means++: The advantages of careful seeding*, Stanford.

Audi, R. 1999, 'The Cambridge dictionary of philosophy'.

Azzopardi, M. & Cortis, D. 2013, 'Implementing automotive telematics for insurance covers of fleets', *Journal of technology management & innovation*, vol. 8, no. 4, pp. 59-67.

Baecke, P. & Bocca, L. 2017, 'The Value of Vehicle Telematics Data in Insurance Risk Selection Processes', *Decision Support Systems*, pp. 69-79.

Bao, C., Wu, D. & Li, J. 2018, 'A Knowledge-Based Risk Measure From the Fuzzy Multicriteria Decision-Making Perspective', *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 5, pp. 1126-38.

Basseville, M. & Nikiforov, I.V. 1993, *Detection of abrupt changes: theory and application*, vol. 104, Prentice Hall Englewood Cliffs.

Bowne, B.F., Baker, N.R., Marzinzik, D.L., Riley, M.E., Christopulos, N.U., Fields, B.M., Wilson, J.L., Wilkerson, B.T. & Thurber, D.W. 2013, 'Methods to Determine a Vehicle Insurance Premium Based on Vehicle Operation Data Collected Via a Mobile Device', Google Patents.

Breiman, L. 2001, 'Random forests', *Machine learning*, vol. 45, no. 1, pp. 5-32.

Caliński, T. & Harabasz, J. 1974, 'A dendrite method for cluster analysis', *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1-27.

Céréghino, R. & Park, Y.-S. 2009, 'Review of the self-organizing map (SOM) approach in water resources: commentary', *Environmental Modelling & Software*, vol. 24, no. 8, pp. 945-7.

Chan, S.H., Song, Q., Sarker, S. & Plumlee, R.D. 2017, 'Decision support system (DSS) use and decision performance: DSS motivation and its antecedents', *Information & Management*, vol. 54, no. 7, pp. 934-47.

Chen, C.P. & Zhang, C.-Y. 2014, 'Data-intensive applications, challenges, techniques and technologies: A survey on Big Data', *Information sciences*, vol. 275, pp. 314-47.

Chen, Z., Yu, J., Zhu, Y., Chen, Y. & Li, M. 2015, 'D 3: Abnormal driving behaviors detection and identification using smartphone sensors', *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, IEEE, pp. 524-32.

Chi, M., Plaza, A., Benediktsson, J.A., Sun, Z., Shen, J. & Zhu, Y. 2016, 'Big data for remote sensing: challenges and opportunities', *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207-19.

Davies, D.L. & Bouldin, D.W. 1979, 'A cluster separation measure', *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224-7.

Desyllas, P. & Sako, M. 2013, 'Profiting from business model innovation: Evidence from Pay-As-You-Drive auto insurance', *Research Policy*, vol. 42, no. 1, pp. 101-16.

Dincer, H., Hacioglu, U., Tatoglu, E. & Delen, D. 2016, 'A fuzzy-hybrid analytic model to assess investors' perceptions for industry selection', *Decision Support Systems*, vol. 86, pp. 24-34.

Dong, W., Li, J., Yao, R., Li, C., Yuan, T. & Wang, L. 2016, 'Characterizing Driving Styles with Deep Learning', *arXiv preprint arXiv:1607.03611*.

Dong, W., Yuan, T., Yang, K., Li, C. & Zhang, S. 2017, 'Autoencoder regularized network for driving style representation learning', *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, AAAI Press, pp. 1603-9.

Du, H.-k., Cao, J.-x., Xue, Y.-j. & Wang, X.-j. 2015, 'Seismic facies analysis based on self-organizing map and empirical mode decomposition', *Journal of Applied Geophysics*, vol. 112, pp. 52-61.

Duri, S., Gruteser, M., Liu, X., Moskowitz, P., Perez, R., Singh, M. & Tang, J.-M. 2002, 'Framework for security and privacy in automotive telematics', *Proceedings of the 2nd international workshop on Mobile commerce*, ACM, pp. 25-32.

Eboli, L., Mazzulla, G. & Pungillo, G. 2016, 'Combining speed and acceleration to define car users' safe or unsafe driving behaviour', *Transportation research part C: emerging technologies*, vol. 68, pp. 113-25.

Eboli, L., Mazzulla, G. & Pungillo, G. 2017, 'How to define the accident risk level of car drivers by combining objective and subjective measures of driving style', *Transportation research part F: traffic psychology and behaviour*, vol. 49, pp. 29-38.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A.Y., Foufou, S. & Bouras, A. 2014, 'A survey of clustering algorithms for big data: Taxonomy and empirical analysis', *IEEE transactions on emerging topics in computing*, vol. 2, no. 3, pp. 267-79.

Fazeen, M., Gozick, B., Dantu, R., Bhukhiya, M. & González, M.C. 2012, 'Safe driving using mobile phones', *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1462-8.

Foresti, G.L., Farinosi, M. & Vernier, M. 2015, 'Situational awareness in smart environments: socio-mobile and sensor data fusion for emergency response to disasters', *Journal of Ambient Intelligence and Humanized Computing*, vol. 6, no. 2, pp. 239-57.

Gallupe, R.B. 2007, 'The tyranny of methodologies in information systems research 1', *ACM SIGMIS Database*, vol. 38, no. 3, pp. 20-8.

Gandomi, A. & Haider, M. 2015, 'Beyond the hype: Big data concepts, methods, and analytics', *International Journal of Information Management*, vol. 35, no. 2, pp. 137-44.

Ghasemaghaei, M. & Calic, G. 2019, 'Can big data improve firm decision quality? The role of data quality and data diagnosticity', *Decision Support Systems*, pp. 38-49.

Gionis, A., Mannila, H. & Tsaparas, P. 2007, 'Clustering aggregation', *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 4.

Guo, H., Ji, Y., Qu, T. & Chen, H. 2013, 'Understanding and modeling the human driver behavior based on MPC', *IFAC Proceedings Volumes*, vol. 46, no. 21, pp. 133-8.

Halim, Z., Waqas, M., Baig, A.R. & Rashid, A. 2017, 'Efficient clustering of large uncertain graphs using neighborhood information', *International Journal of Approximate Reasoning*, vol. 90, pp. 274-91.

Handel, P., Skog, I., Wahlstrom, J., Bonawiede, F., Welch, R., Ohlsson, J. & Ohlsson, M. 2014, 'Insurance telematics: Opportunities and challenges with the smartphone solution', *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 57-70.

Hariri, R.H., Fredericks, E.M. & Bowers, K.M. 2019, 'Uncertainty in big data analytics: survey, opportunities, and challenges', *Journal of Big Data*, vol. 6, no. 1, p. 44.

Hassan, M.M., Uddin, M.Z., Mohamed, A. & Almogren, A. 2018, 'A robust human activity recognition system using smartphone sensors and deep learning', *Future Generation Computer Systems*, vol. 81, pp. 307-13.

Heil, J., Häring, V., Marschner, B. & Stumpe, B. 2019, 'Advantages of fuzzy k-means over k-means clustering in the classification of diffuse reflectance soil spectra: A case study with West African soils', *Geoderma*, vol. 337, pp. 11-21.

Henriksson, M. 2016, 'Driving context classification using pattern recognition', Chalmers University of Technology.

Hofmann, E. 2017, 'Big data and supply chain decisions: the impact of volume, variety and velocity properties on the bullwhip effect', *International Journal of Production Research*, vol. 55, no. 17, pp. 5108-26.

Huber, J., Müller, S., Fleischmann, M. & Stuckenschmidt, H. 2019, 'A data-driven newsvendor problem: From data to decision', *European Journal of Operational Research*, pp. 904-15.

Husnjak, S., Peraković, D., Forenbacher, I. & Mumdziev, M. 2015, 'Telematics system in usage based motor insurance', *Procedia Engineering*, vol. 100, pp. 816-25.

Itoh, N. & Kurths, J. 2010, 'Change-point detection of climate time series by nonparametric method', *Proceedings of the world congress on engineering and computer science*, vol. 1, Citeseer, pp. 20-3.

Jun, J., Guensler, R. & Ogle, J. 2011, 'Differences in observed speed patterns between crash-involved and crash-not-involved drivers: Application of in-vehicle monitoring technology', *Transportation research part C: emerging technologies*, vol. 19, no. 4, pp. 569-78.

Kaisler, S., Armour, F., Espinosa, J.A. & Money, W. 2013, 'Big data: Issues and challenges moving forward', *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, IEEE, pp. 995-1004.

Katal, A., Wazid, M. & Goudar, R. 2013, 'Big data: issues, challenges, tools and good practices', *Contemporary Computing (IC3), 2013 Sixth International Conference on*, IEEE, pp. 404-9.

Kawahara, Y. & Sugiyama, M. 2012, 'Sequential change-point detection based on direct density-ratio estimation', *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 5, no. 2, pp. 114-27.

Khanmohammadi, S., Adibeig, N. & Shanehbandy, S. 2017, 'An improved overlapping k-means clustering method for medical applications', *Expert Systems with Applications*, vol. 67, pp. 12-8.

Kohonen, T. 1990, 'The self-organizing map', *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464-80.

Krishnan, A.R., Kasim, M.M. & Bakar, E.M.N.E.A. 2015, 'A short survey on the usage of Choquet integral and its associated fuzzy measure in multiple attribute analysis', *Procedia Computer Science*, vol. 59, pp. 427-34.

Kuechler Jr, W.L. & Vaishnavi, V.K. 2011, 'Promoting Relevance in IS Research: An Informing System for Design Science Research', *Informing Sci. Int. J. an Emerg. Transdiscipl.*, vol. 14, pp. 125-38.

Kurita, T. 1991, 'An efficient agglomerative clustering algorithm using a heap', *Pattern Recognition*, vol. 24, no. 3, pp. 205-9.

Lamm, R., Psarianos, B. & Mailaender, T. 1999, *Highway design and traffic safety engineering handbook*.

Lara, O.D. & Labrador, M.A. 2013, 'A survey on human activity recognition using wearable sensors', *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192-209.

Lee, J. & Jang, K. 2017, 'A framework for evaluating aggressive driving behaviors based on in-vehicle driving records', *Transportation Research Part F: Traffic Psychology and Behaviour*, pp. 610-9.

Li, X., Wang, F. & Chen, X. 2015, 'Support vector machine ensemble based on choquet integral for financial distress prediction', *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 29, no. 04, p. 1550016.

Liu, H., Taniguchi, T., Tanaka, Y., Takenaka, K. & Bando, T. 2017, 'Visualization of driving behavior based on hidden feature extraction by using deep learning', *IEEE Transactions on Intelligent Transportation Systems*.

Liu, S., Yamada, M., Collier, N. & Sugiyama, M. 2013, 'Change-point detection in time-series data by relative density-ratio estimation', *Neural Networks*, vol. 43, pp. 72-83.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. & Alsaadi, F.E. 2017, 'A survey of deep neural network architectures and their applications', *Neurocomputing*, vol. 234, pp. 11-26.

Liu, Y., Li, Z., Xiong, H., Gao, X. & Wu, J. 2010, 'Understanding of internal clustering validation measures', *2010 IEEE International Conference on Data Mining*, IEEE, pp. 911-6.

Lu, J., Yan, Z., Han, J. & Zhang, G. 2019, 'Data-Driven Decision-Making (D 3 M): Framework, Methodology, and Directions', *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 4, pp. 286-96.

Lu, N., Lin, H., Lu, J. & Zhang, G. 2014, 'A customer churn prediction model in telecom industry using boosting', *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1659-65.

Malalur, P.G., Balakrishnan, H. & Madden, S.R. 2013, 'Telematics using personal mobile devices', Google Patents.

Malekipirbazari, M. & Aksakalli, V. 2015, 'Risk assessment in social lending via random forests', *Expert Systems with Applications*, vol. 42, no. 10, pp. 4621-31.

Mamdani, E.H. 1977, 'Application of fuzzy logic to approximate reasoning using linguistic synthesis', *IEEE transactions on computers*, no. 12, pp. 1182-91.

Mangiameli, P., Chen, S.K. & West, D. 1996, 'A comparison of SOM neural network and hierarchical clustering methods', *European Journal of Operational Research*, vol. 93, no. 2, pp. 402-17.

Maulik, U. & Bandyopadhyay, S. 2002, 'Performance evaluation of some clustering algorithms and validity indices', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650-4.

Moreira-Matias, L. & Farah, H. 2017, 'On developing a driver identification methodology using in-vehicle data recorders', *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2387-96.

Mulliner, E., Malys, N. & Maliene, V. 2016, 'Comparative analysis of MCDM methods for the assessment of sustainable housing affordability', *Omega*, vol. 59, pp. 146-56.

Murofushi, T. & Sugeno, M. 1989, 'An interpretation of fuzzy measures and the Choquet integral as an integral with respect to a fuzzy measure', *Fuzzy sets and Systems*, vol. 29, no. 2, pp. 201-27.

Naderpour, M., Lu, J. & Zhang, G. 2013, 'A fuzzy dynamic bayesian network-based situation assessment approach', *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, IEEE, pp. 1-8.

Naderpour, M., Lu, J. & Zhang, G. 2014, 'An intelligent situation awareness support system for safety-critical environments', *Decision Support Systems*, vol. 59, pp. 325-40.

Namvar, A., Ghazanfari, M. & Naderpour, M. 2017, 'A customer segmentation framework for targeted marketing in telecommunication', *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*, IEEE, pp. 1-6.

Namvar, A. & Naderpour, M. 2018, 'Handling Uncertainty in Social Lending Credit Risk Prediction with a Choquet Fuzzy Integral Model', *arXiv preprint arXiv:1804.10796*.

Namvar, A., Siami, M., Rabhi, F. & Naderpour, M. 2018, 'Credit risk prediction in an imbalanced social lending environment', *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 925-35.

Nguyen, T.T., Krishnakumari, P., Calvert, S.C., Vu, H.L. & van Lint, H. 2019, 'Feature extraction and clustering analysis of highway congestion', *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 238-58.

Niu, L., Lu, J. & Zhang, G. 2009, 'Cognition-driven decision support for business intelligence', *Models, Techniques, Systems and Applications. Studies in Computational Intelligence, Springer, Berlin*.

Ohlsson, E. & Johansson, B. 2010, *Non-life insurance pricing with generalized linear models*, vol. 21, Springer.

Pawlak, Z. 1998, 'Rough set theory and its applications to data analysis', *Cybernetics & Systems*, vol. 29, no. 7, pp. 661-88.

Pham, T.D. 2002, 'Combination of multiple classifiers using adaptive fuzzy integral', *Artificial Intelligence Systems, 2002.(ICAIS 2002). 2002 IEEE International Conference on*, IEEE, pp. 50-5.

Power, D.J. 2002, *Decision support systems: concepts and resources for managers*, Greenwood Publishing Group.

Power, D.J. & Sharda, R. 2007, 'Model-driven decision support systems: Concepts and research directions', *Decision Support Systems*, vol. 43, no. 3, pp. 1044-61.

Saiprasert, C., Pholprasit, T. & Thajchayapong, S. 2017, 'Detection of driving events using sensory data on smartphone', *International journal of intelligent transportation systems research*, vol. 15, no. 1, pp. 17-28.

Saraee, M., Vahid Moosavi, S. & Rezapour, S. 2011, 'Application of Self Organizing Map (SOM) to model a machining process', *Journal of Manufacturing Technology Management*, vol. 22, no. 6, pp. 818-30.

Sculley, D. 2010, 'Web-scale k-means clustering', *Proceedings of the 19th international conference on World wide web*, pp. 1177-8.

Seiti, H., Hafezalkotob, A., Najafi, S.E. & Khalaj, M. 2019, 'Developing a novel risk-based MCDM approach based on D numbers and fuzzy information axiom and its

applications in preventive maintenance planning', *Applied Soft Computing*, vol. 82, p. 105559.

Sengupta, A., Bandyopadhyay, D., Van Westen, C. & Van Der Veen, A. 2016, 'An evaluation of risk assessment framework for industrial accidents in India', *Journal of loss prevention in the process industries*, vol. 41, pp. 295-302.

Shen, Y., Pedrycz, W., Chen, Y., Wang, X. & Gacek, A. 2019, 'Hyperplane Division in Fuzzy C-Means: Clustering Big Data', *IEEE Transactions on Fuzzy Systems*.

Shim, J.P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda, R. & Carlsson, C. 2002, 'Past, present, and future of decision support technology', *Decision support systems*, vol. 33, no. 2, pp. 111-26.

Shou, Z. & Di, X. 2018, 'Similarity analysis of frequent sequential activity pattern mining', *Transportation Research Part C: Emerging Technologies*, vol. 96, pp. 122-43.

Shouno, O. 2018, 'Deep unsupervised learning of a topological map of vehicle maneuvers for characterizing driving styles', *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, pp. 2917-22.

Shukla, A.K., Muhuri, P.K. & Abraham, A. 2020, 'A bibliometric analysis and cutting-edge overview on fuzzy techniques in Big Data', *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103625.

Siami, M., Namvar, A., Naderpour, M. & Lu, J. 2018, 'A fuzzy telematics data-driven approach for vehicle insurance policyholder risk assesment', *Data Science and Knowledge Engineering for Sensing Decision Support*, vol. Volume 11, WORLD SCIENTIFIC, pp. 1407-14.

Song, Y., Lu, J., Lu, H. & Zhang, G. 2019, 'Fuzzy clustering-based adaptive regression for drifting data streams', *Accepted by IEEE Transactions on Fuzzy Systems*.

Sugeno, M. 1974, 'Theory of fuzzy integrals and its applications', *Doctorial Thesis*.

Tan, W., Blake, M.B., Saleh, I. & Dustdar, S. 2013, 'Social-network-sourced big data analytics', *IEEE Internet Computing*, vol. 17, no. 5, pp. 62-9.

Toledo, T., Musicant, O. & Lotan, T. 2008, 'In-vehicle data recorders for monitoring and feedback on drivers' behavior', *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 320-31.

Vaia, G., Carmel, E., DeLone, W., Trautsch, H. & Menichetti, F. 2012, 'Vehicle Telematics at an Italian Insurer: New Auto Insurance Products and a New Industry Ecosystem', *MIS Quarterly Executive*, vol. 11, no. 3.

Vesanto, J. & Alhoniemi, E. 2000, 'Clustering of the self-organizing map', *IEEE Transactions on neural networks*, vol. 11, no. 3, pp. 586-600.

Von Luxburg, U. 2007, 'A tutorial on spectral clustering', *Statistics and computing*, vol. 17, no. 4, pp. 395-416.

Wahlström, J., Skog, I. & Händel, P. 2017, 'Smartphone-based vehicle telematics: A ten-year anniversary', *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2802-25.

Wahlström, J., Skog, I., Händel, P., Bradley, B., Madden, S. & Balakrishnan, H. 2019, 'Smartphone Placement Within Vehicles', *IEEE Transactions on Intelligent Transportation Systems*.

Wang, K., Qi, X., Liu, H. & Song, J. 2018, 'Deep belief network based k-means cluster approach for short-term wind power forecasting', *Energy*.

Wang, Q., Zheng, C., Yu, H. & Deng, D. 2015, 'Integration of Heterogeneous Classifiers Based on Choquet Fuzzy Integral', *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2015 7th International Conference on*, vol. 1, IEEE, pp. 543-7.

Wang, W. & Xi, J. 2016, 'A rapid pattern-recognition method for driving styles using clustering-based support vector machines', *American Control Conference (ACC), 2016*, IEEE, pp. 5270-5.

Xiao, Y. & Yu, J. 2012, 'Partitive clustering (K-means family)', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 3, pp. 209-25.

Yu, J., Chen, Z., Zhu, Y., Chen, Y.J., Kong, L. & Li, M. 2017, 'Fine-grained abnormal driving behaviors detection and identification with smartphones', *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2198-212.

Zadeh, L.A. 1965, 'Fuzzy sets', *Information and control*, vol. 8, no. 3, pp. 338-53.

Zhang, D., Zhou, X., Leung, S.C. & Zheng, J. 2010, 'Vertical bagging decision trees model for credit scoring', *Expert Systems with Applications*, vol. 37, no. 12, pp. 7838-43.

Zhang, H., Chow, T.W. & Wu, Q.J. 2016, 'Organizing books and authors by multilayer SOM', *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2537-50.

Zhang, Q., Yang, L.T., Chen, Z. & Li, P. 2018, 'A survey on deep learning for big data', *Information Fusion*, vol. 42, pp. 146-57.

Zhang, T., Ramakrishnan, R. & Livny, M. 1996, 'BIRCH: an efficient data clustering method for very large databases', *ACM sigmod record*, vol. 25, no. 2, pp. 103-14.

Zhang, X., Zhao, X. & Rong, J. 2014, 'A study of individual characteristics of driving behavior based on hidden markov model', *Sensors & Transducers*, vol. 167, no. 3, p. 194.

Zhao, Y. 2000, 'Mobile phone location determination and its impact on intelligent transportation systems', *IEEE Transactions on intelligent transportation systems*, vol. 1, no. 1, pp. 55-64.

Zhao, Y. 2002, 'Telematics: safe and fun driving', *IEEE Intelligent systems*, vol. 17, no. 1, pp. 10-4.

Zhou, M., Liu, X.-B., Chen, Y.-W., Qian, X.-F., Yang, J.-B. & Wu, J. 2020, 'Assignment of attribute weights with belief distributions for MADM under uncertainties', *Knowledge-Based Systems*, vol. 189, p. 105110.

Zhou, Z., Dou, W., Jia, G., Hu, C., Xu, X., Wu, X. & Pan, J. 2016, 'A method for real-time trajectory monitoring to improve taxi service using GPS big data', *Information & Management*, vol. 53, no. 8, pp. 964-77.

Zhu, G.-N., Hu, J. & Ren, H. 2020, 'A fuzzy rough number-based AHP-TOPSIS for design concept evaluation under uncertain environments', *Applied Soft Computing*, p. 106228.

Zhu, L., Yu, F.R., Wang, Y., Ning, B. & Tang, T. 2018, 'Big data analytics in intelligent transportation systems: A survey', *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383-98.

Zimmermann, H. 1998, 'Fuzzy Decision Support Systems.', in Z.L.A. Kaynak O., Türkşen B., Rudas I.J. (ed.), *Computational intelligence: soft computing and fuzzy-neuro integration with applications*, Springer, Berlin, Heidelberg, pp. 198-229.

APPENDIX: ABBREVIATIONS

AggC	Agglomerative Clustering
ACC	Accuracy
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AUC	Area Under Curve
ARNet	Auto-encoder regularized network
BirC	Birch Clustering
BTS	Base Transceiver Station
CFIVB	Choquet Fuzzy Integral Vertical Bagging
CHI	Calinski-Harabasz index
CM	Confusion Matrix
CNN	Convolutional Neural Networks
COV	Covariance
D3M	Data-Driven Decision making
DAE	Deep Auto-Encoders
DBI	Davis-Boulding index
DM	Decision making
DSAE	Deep Sparse Auto-Encoders
DSS	Decision Support System
DT	Decision Tree
FCM	Fuzzy C means clustering
FDSS	Fuzzy Decision Support System
FE	Features Extraction
FI	Fuzzy Integral
FLS	Fuzzy Logic System
FN	False Negative
FP	False Positive

GBC	Gradient Boosting Classifier
GBM	Gradient Boosting Machine
GPS	Geographic position system
IV	Imputed Value
IVDR	In-Vehicle Data Recorder
LR	Logistics Regression
MADM	Multi-Attribute Decision Making
MBK-means	Mini-Batch K-means Clustering
MCDM	Multi-Criteria Decision Making
MAD	Mean Absolute Deviation
MD	Missing Data
ML	Machine Learning
MSQ	Mean Squared Error
PCA	Principal Component Analysis
RF	Random Forest
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
RS	Risk Score
RST	Rough Set Theory
RuLSIF	Relative unconstrained Least-Squares Importance Fitting
ParC	Partitive Clustering
SD	Standard Deviation
SpeC	Spectral Clustering
SME	Subject Matter Expert
SOM	Self-Organizing Map
SSE	Sum of Squares Error
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

VB	Vertical Bagging
Wi–Fi	Wireless Local Area Network