

2014

A Method for Classifying Driver Performance

Ishika Zonina Towfic

University of Windsor, ishikatowfic@gmail.com

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Towfic, Ishika Zonina, "A Method for Classifying Driver Performance" (2014). *Electronic Theses and Dissertations*. Paper 5164.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

A Method for Classifying Driver Performance

By

Ishika Zonina Towfic

A Thesis
Submitted to the Faculty of Graduate Studies
through the Department of Mechanical, Automotive, and Materials Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada

2014

© 2014 Ishika Zonina Towfic

A Method for Classifying Driver Performance

by

Ishika Zonina Towfic

APPROVED BY:

Dr. C. Lee
Department of Civil and Environmental Engineering

Dr. B. Minaker
Department of Mechanical, Automotive, and Materials Engineering

Dr. J. Johrendt, Advisor
Department of Mechanical, Automotive, and Materials Engineering

July 17, 2014

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Driving performance can be directly related to the driver behaviour in terms of the mental workload and risk perception. No generally accepted model or system exists that can model the driving task or driver performance in a comprehensive manner. The purpose of this research is to develop a methodology using a series of modelling techniques to evaluate driving performance under naturalistic driving contexts. Exploratory statistical techniques and artificial neural network have been used as the backbone of the work presented in this thesis to determine and classify driver performance in different categories by identifying underlying natural sub-sets in the driving data set. A safe and experienced driver should possess the knowledge and the experience about his/her driving skills along with an acute awareness of the surrounding driving environment. The methodology proposed in this thesis can be used for various applications including evaluation of driving performance of emergency ambulance drivers.

*To my loving parents, Dina and Towfic,
For their unwavering support, unconditional love,
And continuous encouragement.*

ACKNOWLEDGEMENTS

There are a number of people without whom this journey would have not been possible. I am forever indebted to them. I would also like to thank the AUTO21 Network of Centres of Excellence for their funding support for this research.

Foremost, I would like to express my deepest gratitude and appreciation for my supervisor, Dr. Johrendt, for her patience, knowledge, motivation, technical guidance, and kindness. I consider myself extremely fortunate to have her as my mentor and I am very thankful to her for believing in me. I would also like to thank my committee members, Dr. Minaker and Dr. Lee, for their technical guidance and support throughout the course of this research. Dr. Minaker has always offered me valuable advice, and Dr. Lee has helped me understand and learn the importance of statistical techniques for conducting this research. Much of what I have learnt and achieved during my Master's program is due to my supervisor and my committee members.

I would also like to take this opportunity to thank the Safety Ambulance Monitoring Unit (SAMU) research team and Université Laval for providing me with the resources for successfully conducting this research.

There are some very special people that I would like to acknowledge and express my sincerest gratitude. I would like to thank my sisters, Adiba and Tasnuva, for believing in me and being there for me through thick and thin. I would like to specially thank my partner, Luv, for his patience and understanding. He has always kindly lent me his ears and patiently listened to all my worries and troubles even when it made no sense. I am very grateful for his support and his advice. He has been an integral part of my journey.

"I have no special talents. I am only passionately curious."

- Albert Einstein

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iii
ABSTRACT	iv
DEDICATION	v
ACKNOWLEDGEMENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF APPENDICES	xii
LIST OF ABBREVIATIONS/NOMENCLATURE	xiii
LIST OF SYMBOLS	xiv
CHAPTER 1 INTRODUCTION	1
<i>1.1 Research Background</i>	1
<i>1.2 Research Objectives</i>	2
<i>1.3 Research Applications</i>	3
CHAPTER 2 LITERATURE REVIEW	5
<i>2.1 Theoretical Models</i>	5
<i>2.2 Mathematical Models</i>	8
CHAPTER 3 STATISTICAL METHODS	12
<i>3.1 Background Mathematics</i>	12
<i>3.1.1 Statistical Concepts</i>	13

3.1.2 <i>Matrix Algebra Concepts</i>	14
3.2 <i>Cluster Analysis</i>	16
3.2.1 <i>Hierarchical Agglomerative Clustering</i>	16
3.2.2 <i>Ward's Method</i>	17
3.3 <i>Factor Analysis</i>	20
3.3.1 <i>Factor Model</i>	20
3.3.2 <i>Principal Component Method</i>	23
CHAPTER 4 ARTIFICIAL NEURAL NETWORKS	25
4.1 <i>Background</i>	25
4.2 <i>Multi-Layer Perceptrons</i>	25
4.3 <i>Processing Unit</i>	27
4.3.1 <i>Activation Functions</i>	28
4.4 <i>Network Training</i>	30
4.4.1 <i>Learning Algorithms</i>	32
4.5 <i>Design Considerations and Validation</i>	34
CHAPTER 5 MULTIVARIATE DATA SET	36
5.1 <i>Data Collection</i>	36
5.1.1 <i>ECEF Coordinates</i>	38
5.2 <i>Raw Data Processing</i>	39
5.2.1 <i>Latitude and Longitude Correction</i>	39
5.2.2 <i>Vehicle Speed Determination</i>	40
5.2.3 <i>Vehicle Acceleration Determination</i>	42
5.2.4 <i>Distance Travelled</i>	43
5.3 <i>Data Extraction for Modelling</i>	44
5.4 <i>Outlier Detection</i>	46

5.5 Data Standardization	46
CHAPTER 6 DRIVER PERFORMANCE CLASSIFICATION	48
6.1 Unsupervised Classification.....	48
6.2 ANN Model for Classification of Driver Performance	55
6.2.1 Network Architecture.....	56
6.2.2 Network Training and Validation.....	57
6.2.3 Network Results	58
6.3 Identification of Significant Variables	62
CHAPTER 7 DRIVER PERFORMANCE CLASS DESCRIPTION	66
7.1 Data Dimensionality Reduction	66
7.2 Factor Model for Driver Performance Classification	68
7.3 Interpretation of Factors.....	69
CHAPTER 8 CONCLUSIONS AND FUTURE WORK	74
8.1 Conclusions	74
8.2 Future Work	76
REFERENCES.....	77
APPENDICES.....	80
Appendix A: Driving Data Set.....	80
Appendix B: Cluster Analysis Results	82
Appendix C: ANN Results	84
Appendix D: Factor Analysis Results	87
VITA AUCTORIS	88

LIST OF TABLES

Table 4.1: Commonly Used Activation Functions for ANNs.....	29
Table 5.1: List of Input Variables	45
Table 6.1: Results of Hierarchical Agglomerative Clustering with Five Classes	54
Table 6.2: Distance between Cluster Centroids	55
Table 6.3: Example of Transforming Classes to Binary Target Values	55
Table 6.4: Final ANN Architecture for Driver Performance Classification.....	56
Table 6.5: ANN Training Parameters and Performance Results	58
Table 6.6: Sensitivity Analysis Results for Driver Performance Class	65
Table 7.1: Eigenvalue Analysis of the Covariance Matrix for Driving Parameters	67
Table 7.2: Factor Loading Values and Communalities for Factor Model	68
Table 7.3: Class Description Based on Factor Model Results.....	72
Table 8.1: Summary of Results for Driver Performance Classification	74
Table A.1: Data Set of Driving Parameters	80
Table A.2: Standardized Values for Final Data Set.....	81
Table B.1: Results for Gap Criterion	82
Table B.2: Assignment of Data to Individual Classes	82
Table B.3: Cluster Centroids with respect to Each Variable	83
Table C.1: Error Values Generated by the ANN Network	84
Table C.2: Input Layer Weights for Driver Performance Classification ANN	86
Table C.3: Hidden Layer Weights for Driver Performance Classification ANN...	86
Table D.1: Factor Values Corresponding to Each Test Drive	87

LIST OF FIGURES

Figure 1.1: Research Methodology	3
Figure 2.1: Comprehensive Overview of a Sample Driver Model	6
Figure 3.1: Hierarchical Agglomerative Clustering Method Overview	17
Figure 3.2: Hierarchical Agglomerative Clustering using Ward’s method	19
Figure 3.3: Sample Dendrogram to Visualize Clusters.....	19
Figure 3.4: Sample Scree Plot to Determine Number of Factors	24
Figure 4.1: Feedforward MLP Network Architecture	26
Figure 4.2: Processing Unit.....	27
Figure 5.1: Visual Information from Internal and External Environment of the Vehicle Synchronized and Fused Together	37
Figure 5.2: ECEF Coordinates	38
Figure 5.3: (a) Raw Signal from Latitude Channel (b) Latitude Channel after “Zero” Correction	39
Figure 5.4: Example of a Final Corrected Latitude Channel obtained from raw GPS data.....	40
Figure 5.5: Speed Profile for a Prius Test Drive.....	41
Figure 5.6: Comparison between Speed and Acceleration Profile of a Prius Test Drive (a) Speed Profile (b) Acceleration	42
Figure 5.7: Total Distance Travelled for a Prius Test Drive.....	43
Figure 6.1: Dendrogram of Driving Data Set for Initial Cluster Analysis.....	49
Figure 6.2: Evaluation of Optimal Number of Clusters using C-H Criterion.....	51
Figure 6.3: Evaluation of Optimal Number of Clusters using Gap Criterion.....	52
Figure 6.4: Dendrogram Showing Assignment of Data Set in to Four Classes	54
Figure 6.5: Neural Network Architecture for Classifying Driver Performance	57
Figure 6.6: ANN Performance Curves for Training, Validation, and Testing	59
Figure 6.7: Error Histogram for Designed ANN Network	60
Figure 6.8: Confusion Matrix for Driver Performance Classification using ANN.....	62
Figure 7.1: Scree Plot for Determining the Number of Factors for Factor Model	67
Figure 7.2: Classification of Driver Performance based on Factor Model	71
Figure C.1: Regression Plots for Developed ANN Model	85

LIST OF APPENDICES

Appendix A: Driving Data Set	80
Appendix B: Cluster Analysis Results	82
Appendix C: ANN Results	84
Appendix D: Factor Analysis Results	87

LIST OF ABBREVIATIONS/NOMENCLATURE

3D	Three dimensional
AANN	Auto Associative Neural Networks
ANN	Artificial Neural Networks
ANOVA	Analysis of Variance
BSLM	Boosting Sequential Labelling Method
C-H	Calinski and Harabasz Criterion for determining Optimal Number of Clusters
ECEF	Earth Centered, Earth Fixed Coordinate System
EPS	Environmental Perceptual System
GADGET	Guarding Automobile Drivers through Guidance Education and Technology
GPS	Global Positioning System
GRAME	Research Group in Motion Analysis and Ergonomics
LM	Levenberg Marquardt Training Algorithm for ANN
LPA	Linear Prediction Analysis
MATLAB	Numerical Computing and Simulation Software
Minitab	Statistical Software Package
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error

LIST OF SYMBOLS

R^2	Proportion of variation by a cluster
SS_B	Between cluster variance
SS_{Error}	Error sum of squares
SS_{Total}	Total sum of squares
SS_W	Within cluster variance
V_{10}	Percentage of Test Drive where Vehicle Speed Exceeds 10% of the Posted Speed Limit
V_{res}	Resultant vehicle speed in km/h
\bar{X}	Sample mean
x_{std}	Standardized value
σ_x	Sample standard deviation
σ_x^2	Variance
b	Bias value for ANN
C	Variance-covariance matrix
d	Euclidean distance
der	Sum of derivatives of error with respect to the ANN inputs
$dist$	Haversine distance in km
E	Error between actual and predicted values
I	Identity matrix
L	Factor loading matrix
n	Number of observations in data set
r	Sample Pearson correlation coefficient
RD	Residual matrix
u	Weighted input
v	Eigenvector corresponding to eigenvalue λ
w	ANN weight
x, y	Variables
z	ANN output

α	Damping factor
β	Factor loading
λ	Eigenvalue
a	Vehicle acceleration
$cov(x, y)$	Co-variance between variables x and y

CHAPTER 1

INTRODUCTION

1.1 Research Background

Driving is a complex task that requires the driver to employ a wide range of skills in order to interact with a complex environment, while simultaneously managing different driving task demands. The driving task can be considered as the interaction between various vehicle, driver, and environment characteristics. Nowadays, the magnitude of information available to drivers through various technological systems and advancements is simply overwhelming. Each individual driver is unique and thus, their level of performance greatly relies on their driving behaviour through successful processing of available information. Every driver has a different perception of acceptable risk levels. These risk levels are subjective in nature and might be influenced by driver age, gender, lifestyle, social background, etc., which in turn can dictate the driver performance to a certain extent. A good driver must therefore possess an adequate level of mental and physical skills to control the vehicle within the environment that they are expected to function.

No generally accepted model or system exists that can model the driving task or driver performance in a comprehensive manner. Driving performance can be directly related to the driver behaviour in terms of the mental work load and risk perception. The driver can choose to follow different strategies for a given risky scenario by adjusting the various driving parameters (e.g. choice of vehicle speed) to constantly adjust the perceived level of risk to an acceptable value. Moreover, there is no standard set of variables defined that can be used to develop a comprehensive model encompassing driver performance and behaviour. The nature of data sets available for analysis varies greatly amongst research groups since each research group develops driver models tailored to their research needs.

Extensive research has been conducted to understand the reasons leading to roadway collision events and as such, drivers have been identified as one of the major contributors for such events. Driving is a dynamic event with continuously changing

demands and levels of interaction or attention required by the driver. According to Transport Canada's National Collision Database (NCDB), 73% of recorded injury collisions in 2010 occurred in urban areas, which includes metropolitan streets and residential areas [1]. Reducing the number of collisions that lead to such events are of key interest to policy makers around the world. Any such traffic event is the interaction between the driver, vehicle, and the road. The work done for this thesis involves the use of factors involved with the vehicle and the road to help identify an evaluation criteria for driver performance. It is very important to understand and classify driving behaviour under different categories based on driving performance and associated levels of risk involved. Such analysis is also very important in understanding and identifying factors that can lead to risky driving scenarios. The work presented in this thesis can have a wide range of applications for developing a better understating of the underlying characteristics inherent to the driving task.

1.2 Research Objectives

A safe and experienced driver should possess the knowledge and the experience about their driving skills along with an acute awareness of the surrounding driving environment. However, it is very challenging to objectify and quantify such phenomena mathematically, based on naturalistic driving data. Often collision scenarios are a result of driver errors, such as errors in risk perception, driving distractions, etc. Therefore, there is a need to identify possible factors and scenarios contributing to risky driving behaviour so that mitigating solutions (e.g. tailored driver training modules) can be further developed to help prevent risky driving events or collisions.

The purpose of this research is to develop a methodology using a series of modelling techniques to evaluate driving performance under naturalistic driving contexts, as presented in Figure 1.1. The methodology was developed using an iterative approach where every step was verified to ensure accurate model results. This thesis proposes a classification model based on driver performance that is indicative of the task demand and risk perception of the driver. In particular, the presented work analyzes the driving performance under an urban driving setting (metropolitan streets and residential streets).

The objective is develop a driver classification model, using artificial neural networks (ANN), that can classify driver performance using different classes or categories.

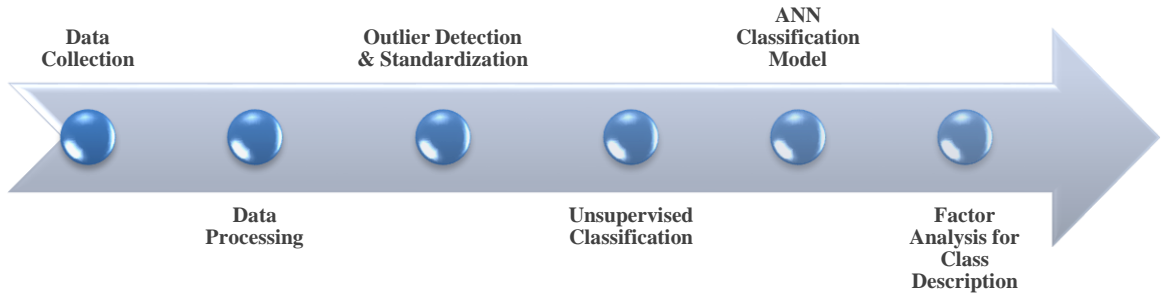


Figure 1.1: Research Methodology

Neural network is a very powerful mathematical tool that can determine, predict, and classify complex non-linear relationships without any prior assumptions. Based on analysis of a driving data set, the proposed work will attempt to determine the relationship between the driver performance and the associated risk factors using ANNs and statistical methods. Statistical modelling techniques will be used extensively used throughout this thesis to determine and interpret the natural subsets within the driving data set. Emphasis will also be placed on data processing techniques for tackling complex data sets containing quantitative information. In particular, the variables presented in this thesis assume continuous numerical values. Determining associated trends in a given data set is very important for building successful models using ANNs. Furthermore, the developed model will be analyzed to determine how individual input parameters affect individual outcomes of the model, by performing a sensitivity analysis. It should be realized that the proposed framework developed in the following thesis is based on information available in the data set. It serves as a guideline and provides an overview of the methodology that will be implemented to model driver performance, which can be used for various applications.

1.3 Research Applications

Apart from evaluating driving performance under regular driving scenarios, the work done for this thesis can also be extended to include various other purposes. One such

application is the use of such analysis and modelling techniques for evaluating the driving performance of emergency services such as ambulances. Recently, there has been growing concern due to the increased number of ambulance collisions, which can cause serious injuries to occupants and cause extensive damage to expensive emergency vehicles. For instance, over 370 ambulance collisions were reported in Quebec City alone in the past five years [2]. Research indicates that ambulance drivers are one of the most important factors influencing ambulance collisions. Thus, it is very important to understand individual driver behaviour and characteristics that can lead to risky behaviours such as speeding and overtaking other vehicles. Although driving an ambulance in emergency situations will be different from regular driving conditions, similar methodology and modelling procedures can be followed to develop an evaluation tool for driver performance. An emergency situation can be described as a scenario where an ambulance needs to respond immediately to an emergency call within a specific time frame. It is a dangerous activity that involves very risky and hazardous situations. The driver is often required to make immediate decisions based on the available information at that instant. Such analysis will greatly assist in understanding ambulance collisions and will help in decreasing the number of ambulance collisions in the future. A similar methodology can also be implemented for evaluating drivers for effective fleet management.

The work presented in this thesis can be applied to evaluate the performance of elderly drivers. The age structure of the Canadian population demographics is changing, with a significant proportion of the population falling under the age group of 65+ years. Moreover, elderly drivers have a higher risk of being involved in vehicle collisions. Although aging affects every individual in a different manner; driving skills, in general, gradually deteriorate due to an increase in reflex or response times. Instead of deciding arbitrarily to restrict the driving capabilities of an elderly driver, a tool can be developed to evaluate the performance using a similar methodology to the one outlined in this thesis. Such a tool will help in developing a quantitative measure for decisions with respect to driving restrictions. This tool in turn can have a positive impact on the quality of life for the aging population in Canada.

CHAPTER 2

LITERATURE REVIEW

A driver performance classification model can serve as the foundation for understanding driver characteristics and performance that lead to vehicle collisions. Driver modelling is an ongoing research topic and there exist various models that address various aspects of driver performance and behaviour. However, a generally acceptable framework is not available that encompasses all the behavioural effects of a driver that can comprehensively describe the driving task. Limited research has been conducted in this area using exploratory statistical methods combined with neural networks. This chapter highlights some of the important work conducted in this field of research. The knowledge learnt from previous literature served as a guideline for developing the classification model for this thesis. The chapter is divided into two sections. The first section introduces some key theoretical models that have served as the foundation for driver models. The second section of the chapter presents some relevant work conducted in developing a mathematical model using the knowledge gained from the theoretical models.

2.1 Theoretical Models

A comprehensive way of developing a driver model can be viewed as a combination of three major elements: inputs, information processing or behaviour, and outputs [3] (Figure 2.1). These parameters can be used to determine measures of effectiveness, or to evaluate driver performance. According to researchers, the occurrence of vehicle collisions can be reduced significantly if technology were available to detect the driver's potential for risky behaviour. If a technology existed that could identify individual driver characteristics responsible for poor driving performance, it would allow for the minimization of driver errors (through development of tailored training programs or advanced vehicle assistance systems), which could play a key role in automotive safety improvement. Based on Wilde's risk homeostasis theory, drivers attempt to maintain a target level of risk per unit time. If the driver is provided with additional safety measures, the driver will exhibit more risky behaviour to compensate and return to the target level of risk [3]. Every individual driver has a different level of risk perception; hence, it is very important to understand and analyze individual driving performance. If similar trends were

categorized in different levels, a more comprehensive model for performance evaluation could be developed. One of the major drawbacks of such theories pertaining to driver modelling is that they only explain the factors and motivations affecting the driver behaviour without describing the process involved in determining such behavioural characteristics. Also, such models cannot quantify factors associated with risk levels which can be further utilized to develop a mathematical model.

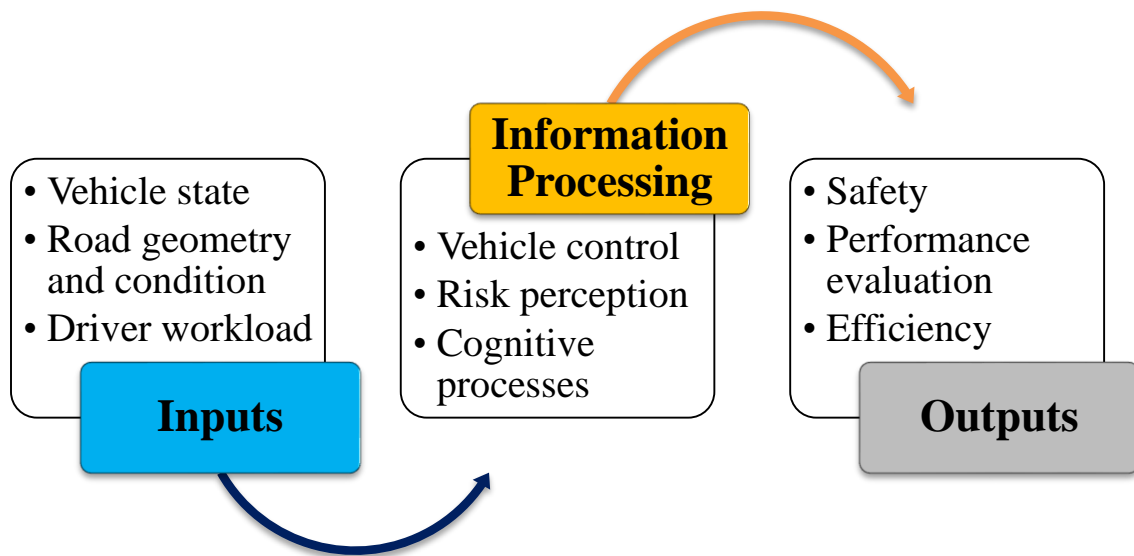


Figure 2.1: Comprehensive Overview of a Sample Driver Model

One of the most popular models for driver behaviour is Michon's Hierarchical Control Model [4]. Michon proposed a simple two way classification model for describing driving behaviour. This one dimensional model is appropriate for distinguishing behaviour using a simple input-output oriented model and internal state models. The second dimension differentiates between functional and taxonomic models, where model components may or may not interact with each other [4]. His model was further subdivided to show that there are three major factors involved in driver decision making: strategy, maneuvering/tactics, and control [4]. Michon's driver behavioural model has served as the basis of various studies over the years in an attempt to develop an effective driver model. For the purpose of this research, more emphasis will be given to the maneuvering/tactical

and control levels of the model, since they are associated with the maneuver execution and decision making processes of the driver. The maneuvering or tactical level is primarily associated with the driver's interaction with the traffic environment, where the driver's actions related to vehicle maneuvering are dependent on his/her level of expertise and interaction with the surrounding environment.

Another notable model that has been extensively used in the literature is the Guarding Automobile Drivers through Guidance Education and Technology (GADGET) matrix [3]. The GADGET matrix was initially developed to assess and structure post license driver education in the European Union. The GADGET matrix is also based on Michon's driver model and consists of four categories for describing driver behaviour – goals for life and skills for living, driving goals and context, mastery of traffic situations, and vehicle manoeuvring [3]. The levels or cells of the GADGET matrix are not mutually exclusive due to the inherent complexity of the driving task. It is possible that some subtasks might be conducted at different levels simultaneously (e.g. speed control, accelerating, braking, etc.). This model is very important since various driving safety laws and regulations in Europe and United States were developed based on the findings from this particular research. For instance, the GADGET matrix was used as the base for developing key driver competencies, which were then integrated into the driver (category B) education program in the European Union.

Overall, the driver uses cognitive, perceptual, and motor abilities to successfully carry out the driving task. Driving can therefore be described as a hierarchy of navigation, guidance, and control conducted simultaneously with visual search, recognition, and monitoring operations [5]. Thus based on the theoretical models and assumptions for risk perception, a comprehensive driver model that can relate the driver performance, capability, and behaviour to the level of associated risk can be proposed using the following five categories – attitudes/personality, experience, driver state, task demand/workload, and situation awareness [3].

2.2 Mathematical Models

Various concepts and theories regarding driver behaviour were explored in Section 2.1. Researchers have used knowledge based on these theories and developed mathematical models for determining driver behaviour and performance. One of the most important factors to consider when attempting to develop driver behaviour models is the vehicle speed. Aarts and van Schagen [6] have highlighted the importance of vehicle speed on road and traffic safety. According to their research, speed not only affects the severity of a vehicle collision, but also increases the risk of being in a collision event. The authors conducted an extensive review of empirical studies relating vehicle speed and the risk of collision. Based on their review of previous literature, they found that the risk of being involved in a collision event is higher for vehicles driving in “minor” or urban roadways when compared to rural or “major” roadways [6]. The authors have also inferred that a higher average speed in urban or minor roadways leads to a higher risk of crash. These observations are very important for the purpose of this research because speed was recorded during the data collection process. Moreover, only the driving data from urban and residential roadways were selected for modelling the driving performance.

Othman et al. [7], conducted a study on driver behaviour and obtained data from a driving simulator using a predetermined computer simulated driving course. In order to extract relevant data from the raw data set, the authors used a linear prediction analysis (LPA) technique to extract relevant features that could best describe the driver operation behaviour [7]. Through LPA techniques, parameters were identified using local data sets. These parameters were used as feature vectors of the driving operation. Feature vectors contain sets of numerical features that can help describe a particular scenario or object. Using Auto Associative Neural Networks (AANN), Othman et al. performed an identity mapping of the feature vectors for each driver and tested the capabilities of the developed network using sets of features from the same and different drivers. The model was primarily developed to identify the driver performance; the proposed method had an overall accuracy of 81.70% [7]. Pedal position, speed, and acceleration were used as the input parameters for the analysis. The model of each driver essentially captured the distribution of features of that driver, and the overall performance of the model was evaluated through the driver identification process. One of the major drawbacks of using such technique is

that it is challenging to interpret the results since it only helps to identify driving patterns; it does not provide information regarding whether the driving performance is satisfactory or not.

Constantinescu, Marinoiu, and Vladoiu [8] also investigated the driving styles of various drivers by classifying the drivers based on their “risk proneness” to group drivers according to their behaviours. The authors used exploratory statistical methods to identify the groups of driving styles based on data collected from an in-house built GPS system. However, the work was only limited to providing some description for each group identified from the set of experimental results. No attempt was made to develop a mathematical model to determine the relationship between each set of identified group and the individual driving parameters.

Another interesting technique was presented by Macadam et al. [9], where the driver behaviour was classified under five different categories using range and range rates of longitudinal closures. The data was trained and classified using ANNs. Once the network was developed, an aggressivity index was defined to reflect the frequency or willingness of a particular driver to overtake and pass other vehicles [9]. The numbering system, combined with the age of the drivers, revealed the associated trends and patterns of behaviour observed in different age groups. However, when a similar technique was applied to represent the longitudinal control behaviour associated with closing-in and tracking of a preceding vehicle, the technique did not yield favourable results. One possible explanation of not obtaining favourable results could be due to the limited number of input parameters for the ANN network. Since driving behaviour is affected by other road and environment characteristics, the network could be trained using a different set of parameters in order to be able classify and predict driver behaviour.

Apart from the use of ANNs, various driver models exist which utilize the concepts of control theory, vehicle dynamics, fuzzy logic, and genetic algorithms. Models developed using vehicle dynamics and control theory are very complex in nature and require several prior assumptions. An example is the model proposed by Sharp, Casanova, and Symonds [10] where a steering control model is developed using linear optimal discrete time preview

control theory. It should be noted that the proposed model is non-linear in nature and the driver model is joined to the vehicle dynamics model to demonstrate the path tracking performance [10]. The developed model requires priori assumptions and the availability of a large number of parameters to satisfy the mathematical expressions. The model demonstrates reliable results but is highly dependent on the assumptions made during the development of the model. One advantage of using neural networks to model such behaviour is that no prior assumptions are required, and results may be achieved using a smaller set of input parameters.

Another example of such a model is the one presented by Raksincharoensak et al. [11] for modelling naturalistic driving behaviour in traffic scenarios. The authors use a combined driver behaviour model using a state transition feature. The driver behaviour model was essentially based on longitudinal vehicle dynamics with a particular focus on vehicle accelerations and braking. The longitudinal driver model was further categorised into five driving states from a viewpoint of active safety [11]. The framework of the work was based on a non-generative method known as the Boosting Sequential Labelling Method (BSLM) which was used to train the model for driver behaviour recognition. Using BSLM, the conditional probability was calculated to determine the relationship between the sensor data and the driving states. Similar to the concept of ANNs, the described technique (BSLM) is a statistical machine learning technique for real time driver state recognition. Using such method, varying levels of accuracies were obtained for the five driving states. A lower accuracy (73%) was observed when determining the driver braking behaviour. Moreover, the mode results deviate significantly when the training data set for the algorithm is altered [11], thus showing some inconsistencies in the model.

On the contrary, neural networks have been particularly successful in the field of driver behaviour modelling since they are able to capture various driving characteristics through an iterative training process along with iterative parameter adjustments to obtain the desired results. One of the major challenges for neural networks is to find a method to interpret the final results. Often it can be challenging to determine the relationship between the network input and output parameters based on an analysis of the network weights.

The literature presented in this chapter has demonstrated the wide range of techniques and applications of driver behaviour and performance modelling. The variables considered for analysis were different for each study. Moreover, the level of detailed information available on the driver-vehicle-environment varies greatly along with the experimental setup. It is thus often challenging to summarize a set of variables that can provide a comprehensive overview of driver behaviour. The work conducted for this thesis utilizes variables (e.g. speed, position, etc.) that can be recorded and obtained in a straightforward manner to determine the various driving characteristics under naturalistic driving conditions.

CHAPTER 3

STATISTICAL METHODS

The following chapter presents an overview of core statistical concepts relevant to this work's analysis of driver performance. Often, such analysis involves multi-dimensional data with three or more variables. Thus, it is sometimes challenging to interpret and analyze data in higher dimensions. The chapter begins with an introduction to mathematical concepts that will be utilized for different statistical modelling techniques. Two analysis techniques will be introduced in this chapter – cluster analysis and factor analysis. Cluster analysis is a widely used technique for exploratory data mining and pattern recognition tasks, amongst many others. Factor analysis is then presented as a dimension reduction technique to model a given multivariate data set, with minimal loss of information. The analyses techniques presented in this chapter will be used for the research presented. The techniques will provide insight into the structure of the data set and help reduce the complexity of the data by identifying factors that are of key importance for the developed model.

3.1 Background Mathematics

This section attempts to provide some common mathematical terminology that will be required to understand the different statistical techniques presented throughout this thesis. This section is divided into two different parts – statistical concepts and matrix algebra concepts. Before proceeding, a multivariate data set will be described that will form the basis of all analysis and techniques mentioned henceforth.

Multivariate Data

Statistical analysis is based on observations and variables. A variable, for the purpose of this thesis, is described as a unique character or quantity that is measured for analysis (e.g. vehicle velocity, distance travelled, etc.). Similarly, an observation is defined as a set of variables or measurements that describes a particular scenario or case. Thus, a multivariate data set is considered as a data set where two or more variables of interest are present for analysis and modelling. The statistical measures presented in the following

sections are for a sample data set of interest. A sample data set is a subset of the entire population, which encompasses all possible scenarios or cases.

3.1.1 Statistical Concepts

Some basic statistical measures and computations are presented in this section that will help to analyze and explore the relationships between different observations and variables in a given data set.

Sample Mean

The mean or average value of a given variable, \bar{X} , in a sample data set can be computed using Equation 1.

$$\text{Sample mean, } \bar{X} = \sum_{i=1}^n \frac{x_i}{n} \quad (1)$$

where, $x_i = i^{\text{th}}$ observation of variable x ,

$n =$ number of observations present in the data set

Sample Standard Deviation and Variance

The standard deviation, σ_x , and variance, σ_x^2 , provide a measure of the spread or dispersion of a given variable from its mean value and can be computed using Equations 2 and 3.

$$\text{Sample standard deviation, } \sigma_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n - 1}} \quad (2)$$

$$\text{Variance, } \sigma_x^2 = \frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n - 1} \quad (3)$$

Covariance

Covariance is a statistical measure of the variance between two given variables, x and y . The covariance can be computed using Equation 4.

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{n - 1} \quad (4)$$

The sign of the covariance is of particular importance in determining the relationship between two different variables. A positive sign indicates that the magnitudes of the two variables increase/decrease simultaneously. On the other hand, a negative value

indicates that the magnitude of the first variable increases while the magnitude of the other variable decreases. If the covariance value is zero, it indicates that the two variables are independent, i.e. their magnitudes are not dependent on each other.

For a multivariate data set consisting of more than two variables, a covariance matrix can be formed which contains all the possible covariance values between the different variables in a data set. A general expression of a covariance matrix, C , of three variables (x , y , and z) can be presented as follows:

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix} \quad (5)$$

It is important to note that the diagonal of matrix C is the variance of each individual variable itself. Thus, C will also be referred to as the variance covariance matrix. Also, it should be noted that the matrix, C , is symmetrical along the diagonal with $cov(x, y) = cov(y, x)$.

Correlation

Correlation is another important statistical measure of dependence between two given variables. The sample Pearson correlation coefficient, r , is used to determine the correlation coefficient between variables, as expressed in Equation 6.

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{X}}{\sigma_x} \right) \left(\frac{y_i - \bar{Y}}{\sigma_y} \right) \quad (6)$$

An r value of +1 indicates a perfectly positive correlation, while a value of -1 indicates a perfectly negative correlation between the x and y variables under consideration. An r value of zero indicates that the values are independent of each other. In a similar manner as the variance covariance matrix, a correlation matrix can also be formed for a multivariate data set consisting of more than two variables.

3.1.2 Matrix Algebra Concepts

The aim of this section is to provide some important background, pertinent to matrix algebra, required for the statistical techniques presented in this chapter and applied throughout the thesis. It is assumed that the reader has a good understanding about the basic

concepts of matrix manipulation. Special emphasis will be provided on some important properties of eigenvalues and eigenvectors since they will be used extensively for the principal component method, introduced later in this chapter.

Eigenvalues and Eigenvectors

Eigenvectors and eigenvalues are important tools for analysis of system of linear equations. Consider a non-zero vector, A , of dimension $(n \times 1)$ multiplied with a square matrix, B , of dimension $(n \times n)$. Vector A will be known as the eigenvector of B only if there exists a solution (real or complex) such that:

$$BA = \lambda A \quad (7)$$

where, λ is an eigenvalue of B . Eigenvectors are only specific to square matrices and will always have a length equivalent to 1. The length of a vector, $\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$ can be defined using Equation 8.

$$length(\lambda_1, \lambda_2) = \sqrt{\lambda_1^2 + \lambda_2^2} \quad (8)$$

Hence, to obtain an eigenvector of unit length, each element of the vector can be divided by the length of the vector, as shown below.

$$\text{Eigenvector } \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \text{ can be re-written as } \begin{pmatrix} \lambda_1 / \sqrt{\lambda_1^2 + \lambda_2^2} \\ \lambda_2 / \sqrt{\lambda_1^2 + \lambda_2^2} \end{pmatrix}$$

A set n of eigenvalues for a given matrix, B , can be essentially determined using Equation 9.

$$B - \lambda I = 0 \quad (9)$$

where, I = identity matrix of dimension $n \times n$.

Once the eigenvalues, λ_i , are computed, the corresponding eigenvectors, A_i , can be determined using Equation 10.

$$(B - \lambda_i I)A_i = 0 \quad (10)$$

Thus, an $n \times n$ matrix will always have n eigenvectors. It is also important to note that all eigenvectors of a matrix are orthogonal to each other, for principal component analysis (i.e. they are linearly independent of each other).

3.2 Cluster Analysis

Cluster analysis can be described as a set of tools to identify clusters or groups, based on natural trends, in a multivariate data set. The clusters or groups are determined based on some similarity criterion such that, the observations within the group have the closest proximity; while, the observations between different groups have the largest proximity. Thus, cluster analysis is considered as an unsupervised learning technique to group data without having information about the similarities within the data set a priori.

Consider a multivariate data set matrix ($m \times x$), consisting of x variables and m observations. The proximity between pairs of observations (m_i, m_j) can be determined by calculating the Euclidean distance, d_{ij} , of the data set matrix using Equation 11 [12]. The Euclidean distance is the most commonly used distance method and is defined as the geometrical distance between two points, derived from the Pythagoras theorem. The greater the distance values, the more dissimilar are the observations and vice versa.

$$d_{ij} = \left(\sum_{k=1}^x |m_{ik}^2 - m_{jk}^2| \right)^{1/2} \quad (11)$$

Once the proximity between the observations are computed, an algorithm can be selected to group the observations, based on their proximities.

Clustering algorithms essentially fall into two categories: partitioning algorithms and hierarchical algorithms. The key differences between the two algorithms rely on the fact that partitioning algorithms require initial assumptions related to the number of clusters and cluster centers. On the other hand, hierarchical clustering algorithms do not require the specification of initial number of clusters. Moreover, hierarchical clustering provides more meaningful and subjective division of clusters based on natural trends in the data set. The following sub-sections will focus on a specific type of hierarchical clustering algorithm, since it was selected as the most appropriate algorithm for the data set to be analyzed.

3.2.1 Hierarchical Agglomerative Clustering

The hierarchical clustering method aims to build a hierarchy of clusters. The method can be further described by two techniques: the agglomerative (“bottom-up” approach) technique and the divisive (“top-down” approach) technique. The agglomerative

method relies on a “bottom-up” approach where each observation in the data matrix is assigned to an individual cluster. The method progresses in stages by merging the closest clusters together, based on a selected algorithm, until one single cluster remains. An overview of the agglomerative clustering method is presented in Figure 3.1.

Most clustering algorithms utilize the computation of distances (e.g. Euclidian distance) to determine the criteria for merging two clusters together. However, after performing cluster analysis with various clustering algorithms, Ward’s method was selected for analysing the multivariate data set because it demonstrated better clustering results than the other algorithms. Ward’s clustering method is described more in details in the following section.

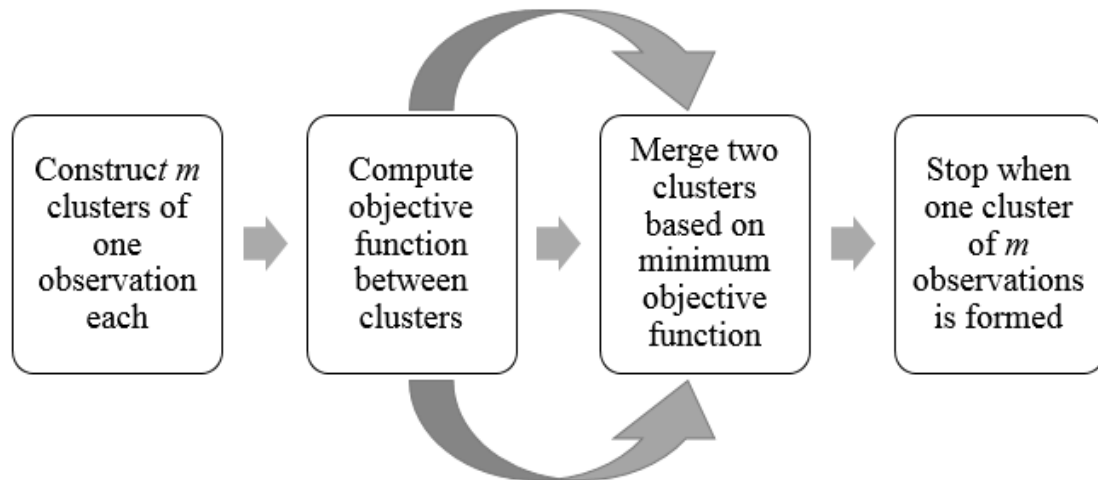


Figure 3.1: Hierarchical Agglomerative Clustering Method Overview

3.2.2 *Ward’s Method*

Cluster analysis using Ward’s method does not involve the computation of proximities or distances like traditional methods. Rather, Ward’s method considers it as an analysis of variance (ANOVA) problem to evaluate the distance between clusters and provides an alternative approach for cluster analysis [13]. Unlike other clustering algorithms, Ward’s method merges groups by ensuring that the variation within the groups does not change significantly. The error sum of squares, total sum of squares, and R-squared values are very important for this method and are computed using Equations 12,

13, and 14 [13], respectively. Let Y_{imx} denote the variable x in observation m belonging to cluster i .

Error sum of squares, SS_{Error} :

$$SS_{Error} = \sum_i \sum_m \sum_x |Y_{imx} - \bar{y}_{ix}|^2 \quad (12)$$

where, \bar{y}_{ix} is the mean of variable x present in cluster i .

The error sum of squares value helps to evaluate individual observations for each variable against the cluster mean of that variable. SS_{Error} is computed for each cluster, and a small value is indicative of the individual observation or data to be very close to the cluster mean, \bar{y}_{ix} . Hence, a relatively small value suggests that a single observation of data is a member of that individual cluster.

Total Sum of Squares, SS_{Total} :

$$SS_{Total} = \sum_i \sum_j \sum_k |Y_{imx} - \bar{y}_x|^2 \quad (13)$$

The total sum of squares helps to evaluate individual observations of each variable against the grand mean, \bar{y}_x , of that variable across all clusters.

Once SS_{Error} and SS_{Total} is computed, the proportion of variation explained by a particular cluster can be explained using the R-squared value:

$$R^2 = \frac{SS_{Total} - SS_{Error}}{SS_{Total}} \quad (14)$$

The agglomerative method starts with m clusters (one cluster for each observation) using the hierarchical approach. The algorithm progresses by determining a pair of observations which yield the smallest SS_{Error} or largest R^2 . Based on the values, $m - 1$ clusters are formed; i.e., only one cluster contains two observations while the rest of the clusters contain one observation each. The process is repeated, at each stage of the algorithm, by clustering together pairs with the highest R^2 value. Since the analysis is done using hierarchical agglomerative technique, the algorithm continues till one large cluster of m observations is formed. An overview of the hierarchical agglomerative using Ward's method is presented in Figure 3.2.

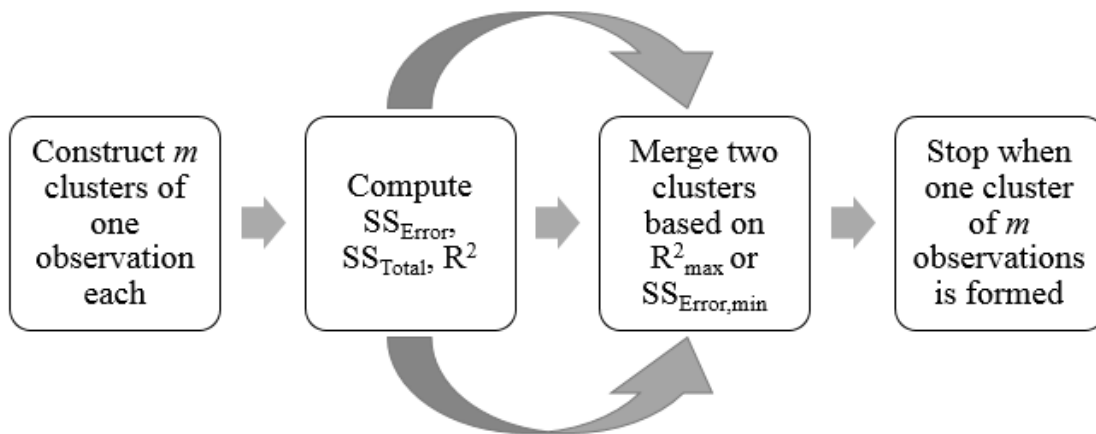


Figure 3.2: Hierarchical Agglomerative Clustering using Ward's method

The results of a cluster analysis can be effectively summarized visually with the aid of a dendrogram. A dendrogram is type of tree diagram with U-shaped links, which helps to visualize the clusters produced by the unsupervised classification method described above. The dendrogram is constructed by plotting the distance versus the observation number, as shown in Figure 3.3.

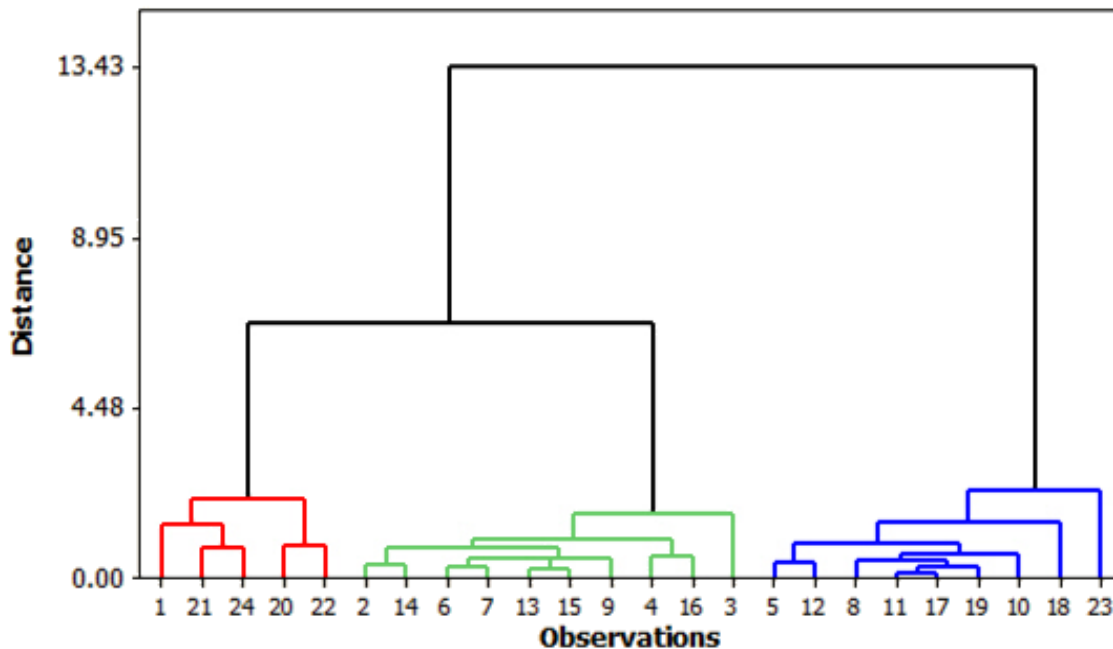


Figure 3.3: Sample Dendrogram to Visualize Clusters

Figure 3.3, shows three distinct groups identified through cluster analysis since the distance values (y axis) of the individual groups are significantly different, relative to each

other. The distance between two individual groups, C_i and C_j can be determined using the combinatorial expression presented in Equation 15 [14].

$$D(C_i, C_j) = \frac{(n_j + n_k)d_{kj} + (n_j + n_l)d_{lj} - n_j d_{kl}}{n_i + n_j} \quad (15)$$

where, d is the Euclidean distance between the two vectors, n_i, n_j, n_k, n_l are the number of observations in C_i, C_j, C_k, C_l . The new cluster C_j is considered to be formed by merging two clusters C_k and C_l , respectively.

The distance between two respective clusters is sum of squared deviations from points to cluster centroids [15]. It should be noted that the relative ranking of the distance in the dendrogram is more important in the formation of clusters rather than the magnitude itself.

3.3 Factor Analysis

In addition to performing a cluster analysis to determine natural groups or clusters in a data set, it is often beneficial to identify the underlying characteristics of the collected data. This can be achieved through factor analysis by defining a small number of factors, n , that can explain most of the variation observed in the data set. The main objective of factor analysis is to provide logical interpretation of a multivariate data set by reducing complexity through identification of factors, which can essentially explain most of the model behaviour. Thus, a data set with x variables can be explained with the help of n variables instead; where, n will always be smaller than x .

3.3.1 Factor Model

Assume a data set of variables y_1, y_2, \dots, y_x . Also, assume that all the variables in the data set are linearly related to a small number of common factors (f_1, f_2, \dots, f_n) . These factors are considered to be inferred from the relationship inherent to the data set, instead of being collected directly from the data. Thus, each variable in the data set can be expressed as a function of the underlying factors, as shown in Equation 16.

$$\begin{aligned}
y_1 &= \beta_{10} + \beta_{11}f_1 + \beta_{12}f_2 + \dots + \beta_{1n}f_n + e_1 \\
y_2 &= \beta_{20} + \beta_{21}f_1 + \beta_{22}f_2 + \dots + \beta_{2n}f_n + e_2 \\
y_x &= \beta_{x0} + \beta_{x1}f_1 + \beta_{x2}f_2 + \dots + \beta_{xn}f_n + e_x
\end{aligned} \tag{16}$$

In equation 16, e_1, e_2, \dots, e_x are considered as the errors between the actual values and the predicted values of a given variable, by the factor model. Moreover, the terms $\beta_{10}, \beta_{21}, \dots, \beta_{fn}$ are the coefficients for the respective factors and are referred to as the factor loadings. Hence, the abovementioned model is analogous to a regression model where each variable can be modelled using n factors, which can explain the variation in the data set. At this point, it should be realized that $n \ll x$.

Before a detailed discussion is presented on how to interpret the factor model, the following assumptions are necessary to uniquely estimate the parameters for the model [13]:

1. The mean and variance of random errors, e_i , are zero: $\bar{e} = 0$, and $\sigma_e^2 = 0$, where $i = 1, 2, \dots, x$.
2. The mean and variance of common factors, f_i , is zero and one respectively: $\bar{f} = 0$, and $\sigma_f^2 = 1$, where $i = 1, 2, \dots, n$.
3. There is no correlation within common factors, errors, and between common factors and errors: $\text{cov}(f_i, f_j) = 0$, $\text{cov}(e_i, e_j) = 0$, and $\text{cov}(e_i, f_j) = 0$.

Based on the assumptions presented for the model, the variance of any given variable x_i can be calculated using Equation 17.

$$\begin{aligned}
\sigma_{x_i}^2 &= \gamma_{i1}^2 \sigma_{f_1}^2 + \gamma_{i2}^2 \sigma_{f_2}^2 + \dots + \gamma_{in}^2 \sigma_{f_n}^2 + (1^2) \sigma_{e_i}^2 \\
\sigma_{x_i}^2 &= \gamma_{i1}^2 + \gamma_{i2}^2 + \dots + \gamma_{in}^2 + \sigma_i^2
\end{aligned} \tag{17}$$

where, the terms $\gamma_{i1}^2 + \gamma_{i2}^2 + \dots + \gamma_{in}^2$ are referred to as the communality, and σ_i^2 is referred to as the specific variance, of any give variable i . Communality represents the portion of the variable that is explained by the common factors; whereas, the specific variance accounts for the portion of $\sigma_{x_i}^2$ that is not explained by the communality.

Communality serves as a good assessment tool to determine how well the developed factor model behaves with respect to a set of variables. Moreover, the covariance between any two variables (x_i, x_j) can be determined using Equation 18.

$$cov(x_i, x_j) = \gamma_{i1}\gamma_{j1} + \gamma_{i2}\gamma_{j2} + \dots + \gamma_{in}\gamma_{jn} \quad (18)$$

Once the variance and covariance are computed for all the variables using Equations 17 and 18, the results can be organized in a variance-covariance matrix, C, as explained in Section 3.1.1. Matrix C will have a dimension of $x \times x$, since there are x variables in the data set.

The variance for each variable is organized along the major diagonal of the matrix, while the covariance between the individual variables is arranged in the remainder elements of the symmetric matrix, C. A matrix computed using original variables from the data set leads to the formation of theoretical variance-covariance matrix. On the other hand, computation of a matrix using the predicted variables from the factor model leads to the formation of an observed variance-covariance matrix. The difference between the theoretical and the observed matrices is stored in a new matrix known as the residual variance-covariance matrix (RD), as shown in Equation 19. The structure of matrix RD is similar to that of matrix C.

$$Residual\ Matrix, RD = C_{theoretical} - C_{predicted} \quad (19)$$

The residual matrix helps to assess the fit of the factor model. The lower the values of the residual matrix, the better the factor model performs in modelling the initial set of variables, p , present the data set.

There are two methods widely used in factor analysis to determine the factor loading values for the model – principal component method and maximum likelihood estimation method. The following section provides an overview of the principal component method, since this method will be further used to analyse the multivariate driving data set. Maximum likelihood estimation method requires the data set to be obtained from a multivariate normal distribution data. Since not all variables necessarily follow a normal distribution, principal component method was selected as the most suitable method for analysis.

3.3.2 Principal Component Method

The objective of the principal component method is to determine the factor loadings in such a manner that the total communality of the model is as close as possible to the total of the predicted variables. Principal component analysis will help to reduce the number of variables in a data set for better interpretation, using linear combinations. Before the principal component method is applied, it is very important for the data to be standardized. The data standardization process and its implications are explained in further detail in Chapter 5.

The first step, in determining the factor loadings using the principal component method, is to construct the variance covariance matrix, C , of the data set using Equations 3 and 4. Since the variance covariance matrix is a square matrix, it can be re-represented using eigenvalues and eigenvectors (Equation 8), as discussed in Section 3.1.2. Since the idea is to reduce the dimension of the data set matrix, the eigenvector and the eigenvalues are arranged in a descending order. Thus, the eigenvector with the highest eigenvalue forms the first principal component and so on.

Once the principal components are identified, the factor loading can be determined using the spectral decomposition (SD) theorem, as shown in Equation 20.

$$SD = \sum_{i=1}^x \lambda_i v_i v_i^T \cong \sum_{i=1}^n \lambda_i v_i v_i^T = LL^T \quad (20)$$

where, v is the eigenvector corresponding to the eigenvalue λ_i , v_i^T is the transpose of v , and L is the factor loading matrix. Thus, the estimator of factor loadings [13] can be expressed using Equation 21 as follows:

$$l_{ij} = v_{ij} \sqrt{\lambda_i} \quad (21)$$

Finally, to determine the number of factors required for the analysis, a scree plot can be generated by plotting the eigenvalues vs. the number of principal components, as shown in Figure 3.4. The number of factors is determined at the point beyond which the eigenvalues are comparably small and do not change significantly with respect to each other. For instance, by observing Figure 3.4, it can be seen that beyond the third component, there is no large change in eigenvalues between the components. As a result, three principal

components can be selected for use in the factor model. The interconnecting lines for scree plots serve as a visual aid for determining the trend, the eigenvalues cannot assume any values along the interconnecting lines (blue lines). The set of eigenvalues calculated for each data set is discrete in nature.

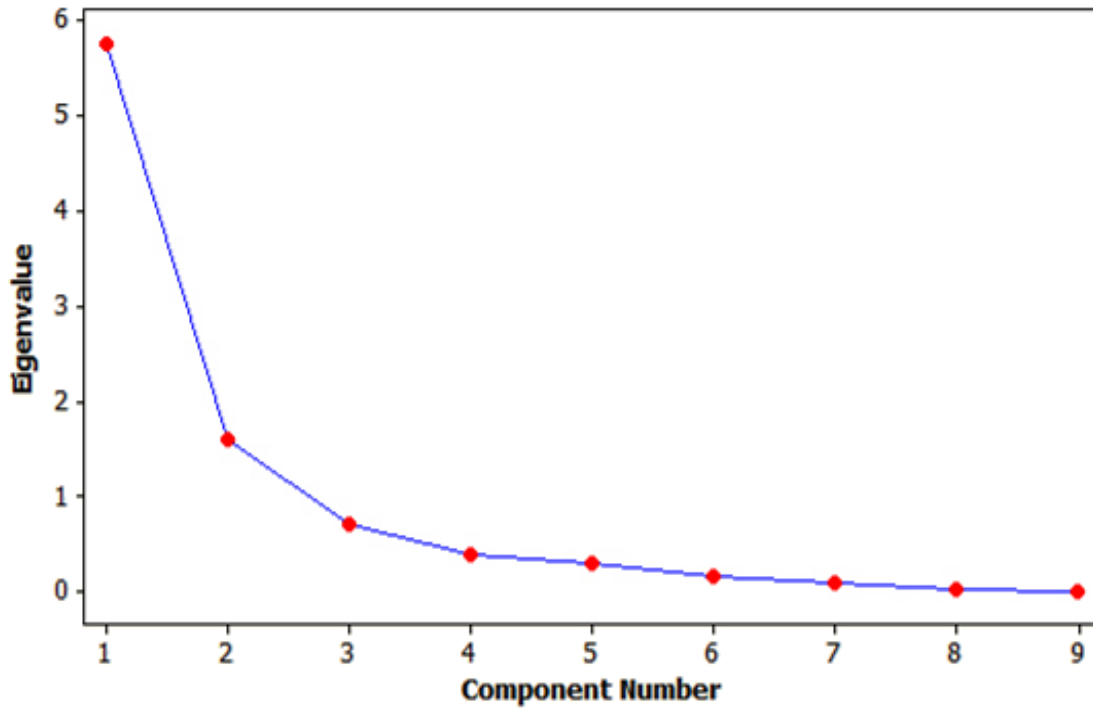


Figure 3.4: Sample Scree Plot to Determine Number of Factors

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS

4.1 Background

Artificial neural networks (ANN), as the name suggests, are inspired by biological neurons and the information processing capabilities of the human brain. ANNs can be described as massive interconnected processing elements (neurons) that can obtain and store knowledge from an external environment or data set. From a mathematical standpoint, neural networks can be considered as “black boxes”, and serve as an essential analysis and modelling tool for multivariate data sets. ANNs are capable of performing a variety of tasks including prediction (function approximation), pattern recognition, and forecasting [16]. They are versatile tools used across multiple disciplines and areas of research ranging from engineering systems and stock market predictions to speech pattern recognition.

Traditional methods for determining the relationship between input and output parameters require a set number of rules, equations, or assumptions for describing the system. One of the biggest advantages of ANNs are that no prior assumptions or rules are required to determine the underlying relationships between the input and the output parameters. This thesis will focus on developing a classification ANN model for categorizing driving performance using a known set of inputs and outputs. This technique is known as supervised learning, where the network attempts to approximate the relationship between the inputs and the different classes of driver performance using a known set of targets or classes. The aim of this chapter is to provide introduction to the concepts associated with the design and construction of classification neural networks. The chapter concludes with some general design guidelines and evaluation methods to ensure the quality, in terms of network accuracy and generalization capabilities, of any desired network of choice.

4.2 Multi-Layer Perceptrons

As mentioned earlier, ANNs are built by interconnecting processing units called neurons. These neurons are interconnected with corresponding parameters known as

weights that help the ANN to learn and map inputs to outputs. The weights are an integral part of a network and help to describe the effect of each single unit on the network output. For the purpose of this research, a specific type of neural network known as multi-layer perceptrons (MLP) will be discussed. MLPs are one of the most widely used types of neural networks [17], consisting of a set of input units and a set of output units connected together through one or more processing hidden units. The hidden units will be primarily used as non-linear classifiers to categorize driver performance. Such networks are arranged in layers and will be represented henceforth as a series of three numbers in the following format: Input layer – Hidden layer(s) – Output layer.

The number of inputs, hidden layer(s), and outputs will be expressed in numeric format to provide an overview of the network architecture. For example, a basic feedforward MLP network with a 4-2-1 architecture is presented in Figure 4.1. The term feedforward indicates that information flows only in one direction in the network (i.e. from inputs to outputs). No feedback loops are present in such networks.

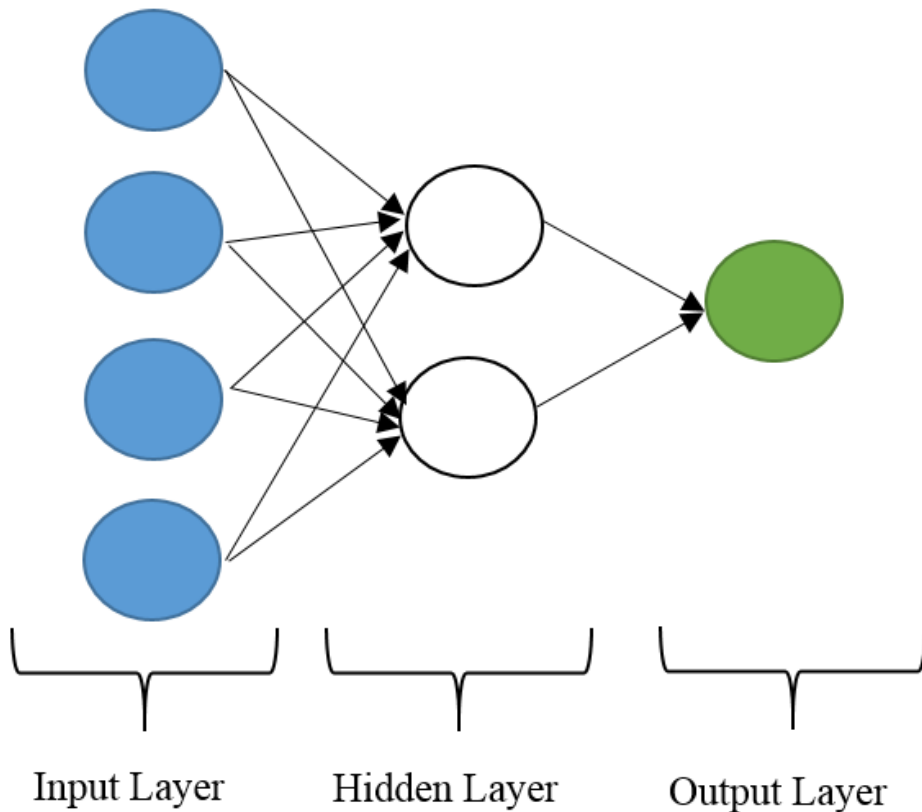


Figure 4.1: Feedforward MLP Network Architecture

The input layer, as seen in Figure 4.1, consists of parameters received from an external environment. The inputs are connected to the hidden layer with the help of weights, which in turn helps to process the input values to determine the relationship between a given set of known output variables. Often, MLP networks also have bias units present in the network. These units are always connected to all processing units except the input layer. Bias values are always set to one, and accounts for the effects that are not explained by the input variables in the model. Thus, bias values can be considered analogous to the intercept value in a statistical regression analysis [16]. The following section discusses the individual characteristics about MLPs and neural networks in detail.

4.3 Processing Unit

The purpose of a processing unit, present in hidden layers and output layers, is to process information and compute an output signal based on the information incoming into the unit from a previous layer (input layer or hidden layer). An example of a processing unit is presented in Figure 4.2.

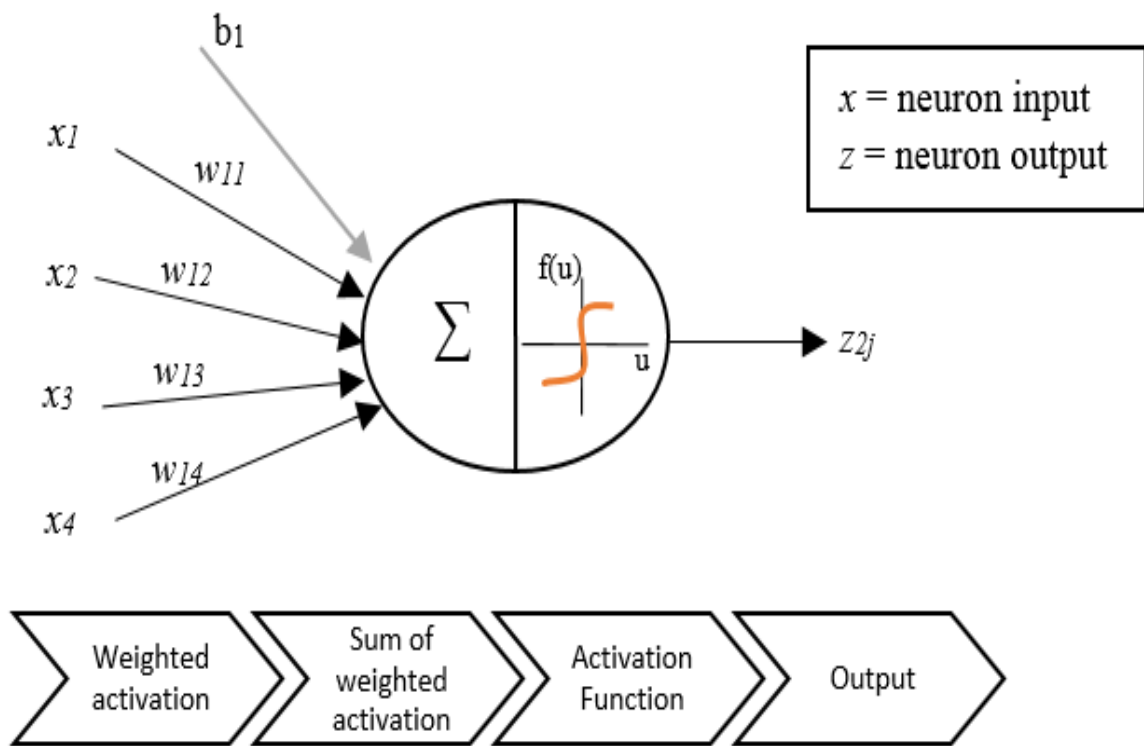


Figure 4.2: Processing Unit

where, w is the weight of the j^{th} neuron of the i^{th} layer, and b_i is the bias value.

The total input to a processing unit is essentially a vector sum of the weighted inputs including the bias value, if any (Figure 4.2). The total input is then transformed by the processing unit by a function called the activation function. A general expression for computing the unit output is presented in Equations 22 and 23.

$$u = \sum_{i=1}^p w_{ij}x_i + b_{ij} \quad (22)$$

$$z_j = f(u) \quad (23)$$

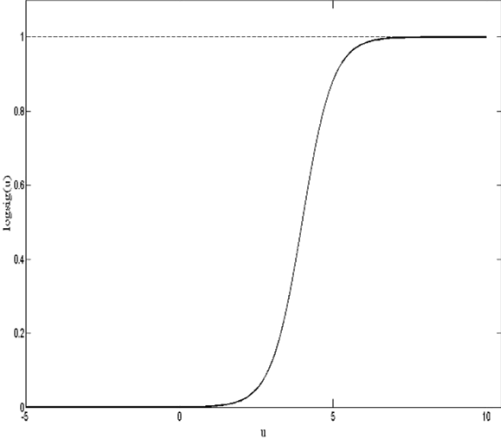
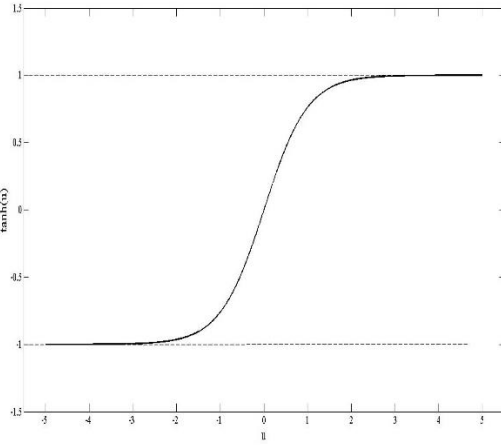
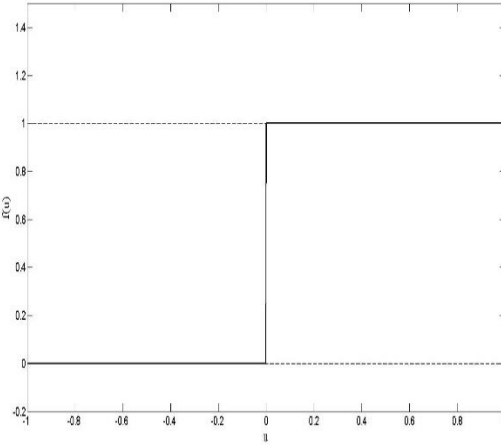
where, p is the number of units in a given layer and u is the weighted inputs.

The choice of activation function is dependent on the model requirements and the data set. Since the objective is to design a classification network with binary outputs, the choice of activation function for the processing units play an important role in the design of the network and is discussed in detail in the following section.

4.3.1 Activation Functions

Non-linearity is introduced to the network through activation functions and hence, allows for complex non-linear mapping of inputs to outputs. Activation functions are mostly continuous differentiable functions. This property is important because weight adjustments are achieved by backpropagating errors through the network. Backpropagation will be discussed further at a later section. Depending on the requirements of the data, the activation functions can be either linear or non-linear in nature. Irrespective of the nature of data entering a neuron, each activation function is bound within a certain operating range, except the linear activation function. Thus, the output from the neuron will never go beyond its operating range value. One of the major reasons why ANNs are able to predict complex non-linear functions is due to the fact that activation functions help to map the data from input to output of a processing unit non-linearly. Three commonly used activation functions for MLPs are presented in Table 4.1, along with their corresponding equations.

Table 4.1: Commonly Used Activation Functions for ANNs

Activation Function	Graphical Representation	Operating Range	Mathematical Representation
<p>Logistic Sigmoid</p>		<p>0 to 1</p>	$\text{logsig}(u)$ $= \frac{1}{1 + e^{-u}}$
<p>Tan Sigmoid</p>		<p>-1 to 1</p>	$\text{tanh}(u)$ $= \frac{1 + e^{-u}}{1 - e^{-u}}$
<p>Step Function</p>		<p>0 or 1</p>	$f(u)$ $= \begin{cases} 1 & \text{if } u \geq \Theta \\ 0 & \text{if } u < \Theta \end{cases}$ <p>where, Θ is a predefined threshold value</p>

The selection of the most suitable activation function for the processing units is vital for designing a neural network. Each activation function has specific properties which should be considered when choosing an appropriate function. The objective of this thesis is to design a classification ANN binary classifiers as outputs. As a result, a suitable activation function for the network can either be the logistic sigmoid activation function, hyperbolic tangent sigmoid activation function, or the step function. The step function is predominantly used for linear networks with single layers [16]. Moreover, step functions are not continuous in nature, thus making them non-differentiable. This makes them unsuitable for computing network gradients and determining trends where non-linearities exist in a given data set. A suitable activation function for a binary non-linear model is the logistic sigmoid function which has an upper bound of +1 and a lower bound of 0 (Table 4.1). On the contrary, tanh activation function reduces network performance and decreases computational time considerably when compared with the logistic sigmoid function [18]. Even though the hyperbolic tangent function (tanh) has an operating range of $[-1, 1]$, it will be preferred over logistic sigmoid function due to the rational mentioned above.

4.4 Network Training

Once the basic structure of the processing units are determined, emphasis is placed on determining a suitable method for training the network. ANNs learn by adjusting a set of weights present in a network in order to successfully build a classification model. An ANN network can learn through either supervised training or unsupervised training.

- **Supervised Learning:** The network is presented with a known set of inputs and corresponding outputs. These known outputs are often referred to as network targets. The error between the calculated network outputs and targets are used by the learning algorithm to update and adjust the weights of the network. MLPs always learn when a set of desired outputs or targets is presented to the network. The ANN network developed for the purpose of this research will be trained using the supervised learning technique.
- **Unsupervised learning:** The network is presented with a set of input parameters without a priori information about the desired outputs. The network adjusts its weights using relevant algorithms to identify underlying properties of the data to be modelled. However, MLPs cannot be trained with this technique.

For a network to learn from a desired output, an error criterion must be selected for evaluation of the network's accuracy. Each input set is presented to the network several times in an iterative manner, so that weights undergo adjustment until the ANN learns to perform the task as desired. The simplest way of determining the error (E) for the j^{th} data point is to compute the difference between the network output (z) and the target output (t) for each unit, as shown in Equation 24.

$$E_j = t_j - z_j \quad (24)$$

To calculate the error of the entire network output layer, the Mean Squared Error (MSE) method is utilized, as shown in Equation 25. MSE is the average of errors and is based on the Pythagoras theorem. MSE sums the squares of the individual errors so that the effect of their positive and negative signs is not taken into consideration. The objective is to obtain a low enough absolute error value which is suitable for a given analysis.

$$MSE = \frac{1}{2p} \sum_{i=1}^p E_i^2 \quad (25)$$

where p is the number of units in a given layer.

The goal of any given MLP network, during the training phase, is to minimise the overall error between the network outputs and the target outputs. This in turn is dependent on how data is presented to the network, and how the network weights are updated. There are essentially two ways in which the network can be trained.

- **Online Learning:** For this method, the error is calculated and the network weights are updated after each observation in the data set is presented to the network.
- **Batch Learning:** This method calculates the error and updates the network weights once the entire data set is passed through the network. Every time the entire set of data passes through the network once, it is referred to as an epoch. At the end of each epoch, the average error for the network is calculated and the weights are updated accordingly.

There are certain things that should be taken into consideration before selecting the training mode for the designed network. The batch learning mode is comparatively faster than the

online learning mode since the weight updates occur less frequently. Moreover, the batch learning method provides a better representation of the required weight updates [17] otherwise this method of network training might cause the network weights to reach a local minimum error instead of a global minimum error; hence, care should be taken when designing and training a network with batch learning mode. The designed ANN for driver performance classification will utilize the concept of batch learning since it is more efficient than the online learning mode.

So far, the material presented in this chapter has provided an overview of the different training methods for designing a neural network. One important part of designing a network is to determine a suitable algorithm to update the weights based on error backpropagation in the MLP network. The following sub-section provides an overview of the different training algorithms that should be taken in to consideration when designing an MLP network.

4.4.1 Learning Algorithms

The learning process takes place by incrementally adjusting the weights of the network such that the activation functions in the processing neurons can generate the desired response [16], i.e. correct classification of performance for the task at hand. During the learning process, the network error eventually decreases, since the difference between the target and the network output gradually decreases. This process of adjusting the weights while reducing the error to a specified level is more generally referred to as the training phase of the network. The training phase can be initiated by assigning random weight values across the network. MSE will be treated as the error indicator for adjusting the weights of the network.

The technique of backpropagating the errors forms a crucial part in the network learning phase. The backpropagation technique involves calculating the first partial derivatives of outputs with respect to inputs. The first derivative of error is often referred to as the sensitivity of error. Essentially the first derivative of error with respect to the output neuron can be computed using Equation 26.

$$\frac{\partial E}{\partial z} = \frac{\partial}{\partial z} \left(\frac{1 + e^{-z}}{1 - e^{-z}} \right) \quad (26)$$

The objective is to determine the partial derivative of error with respect to the hidden neurons and input neurons (Figure 4.2). This can be achieved by utilizing the chain rule of differentiation as shown in Equation 27.

$$\frac{\partial E}{\partial z} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial u} \cdot \frac{\partial u}{\partial x} \quad (27)$$

where, v is the output from the hidden layer neuron and $\frac{\partial v}{\partial u}$ is essentially the derivative of the activation function(s) used in the hidden layer(s).

The above mentioned technique of backpropagating errors can then further be utilized to update the network weights. Some of the more common techniques for network learning algorithm are – delta bar delta learning, steepest descent method, quick propagation, Gauss Newton method, Levenberg Marquardt method, etc. For the purpose of this thesis, only the Levenberg Marquardt (LM) technique will be discussed in detail, since it will be used extensively for the development of the ANN model.

The LM technique computes the second derivative of the error to update the weights, which essentially indicates the rate at which the gradient of error in the designed network changes [16]. At any given point on the error surface, the second derivative of error, with respect to the weight (w), can be expressed as $\frac{\partial^2 E}{\partial w^2}$. Computing the second derivative provides a more efficient method to determine the optimum set of weights for the desired network. For the LM technique, the required weight adjustment is computed using Equation 28.

$$\Delta w_\varepsilon = - \frac{der_\varepsilon}{der_\varepsilon^2 + der^\alpha} \quad (28)$$

where, ε is the epoch number, der is sum of derivatives of error with respect to the inputs, and α is the damping factor. The damping factor, α , is altered accordingly for each epoch in an attempt to minimize the overall error of the network.

The LM technique is a hybrid learning technique which has shown superior results to other learning techniques due to its increased efficiency in converging to a global minimum error as compared to the other methods mentioned earlier [16].

4.5 Design Considerations and Validation

Building a neural network is an iterative and experimental process. This section will summarize the information presented in this chapter, and will present a general set of guidelines for designing networks. No set rules or generic architectures exist for obtaining the desired results. Each network is different in its inherent properties and requires appropriate selection of algorithms and techniques that are tailored to a given set of data. When designing and building an MLP network, some decisions need to be made regarding the network size and architecture. In general, before an ANN can be built, the data needs to be processed and coded using a suitable method to obtain better network performance. More details about data processing methods will be presented in Chapter 5. Apart from that, the number of hidden layers also needs to be determined for preliminary analysis. Designing ANNs is an iterative process and more than one architecture is usually tested before determining the final structure of the network. Some analytical techniques are presented below which serve as guidelines for determining the preliminary network architecture:

- According to Kolmogorov's theorem, the upper bound for the number of hidden units in a network can be represented by the expression $2x + 1$, where x is the number of input variables.
- The number of training patterns should approximately equal the number of weights present in the network multiplied by the error limit [17].
- The number of hidden units are dependent on the properties of the input variables. A complex multivariate data might require more hidden units/layers to learn the underlying relationships.
- There should be a reasonable trade-off between network generalization and network accuracy [17].
- Hyperbolic tangent (tanh) is an asymmetric function and leads to faster learning with fewer epochs than a non-symmetric activation function such as the logistic sigmoid function.

Once the network has been trained using the desired architecture, the error performance and quality of the designed network can be validated by plotting the error and weight distributions of the network. An ideal distribution should follow a normal distribution curve with peaks near the center of the curve (near the zero region). Another good indicator of good network learning performance is the initial and final values of the weights. If the weights have varied significantly from their initial points then it indicates that the desired network has learnt from the data presented [17]. Various other plots can be used to test the performance of a designed network, and will be explored in detail in a later chapter.

CHAPTER 5

MULTIVARIATE DATA SET

The aim of this chapter is to introduce the data set that was used for driver performance classification. Before the techniques described in Chapters 3 and 4 can be utilized for modelling purposes, it is vital to understand the nature of the data set to be used for such analyses. The chapter begins with a general overview of the data collection and data extraction processes. Some explorative and descriptive analyses are conducted on the raw data set to provide an insight into the nature of the raw data. The remainder of the chapter explores methods for transforming the data set in preparation for modelling purposes. Throughout the course of this chapter, the limitations of the data set as well the limitations of the scope of analysis will be discussed so that an in-depth understanding of the nature of the data set is obtained. In the chapter that follows, a processed multivariate data set will be modelled by determining the unobservable natural patterns inherent in the data set.

5.1 Data Collection

The data was collected by the Research Group in Motion Analysis and Ergonomics (GRAME) from Université Laval, Quebec, Canada. The data collection process involved equipping the test vehicles with vision systems, GPS, and a data acquisition system (GPS). The vision systems implemented in the instrumented vehicles served two purposes. The first purpose was to monitor the driver and passenger (if applicable) activities inside the vehicle cabin, and secondly to monitor the external environment of the vehicle through the use of a dual stereoscopic system with accurate three dimensional (3D) forward vision and a 360° field of view [2]. This system for monitoring the external environment is referred to as the Environmental Perceptual System (EPS) and consists of forward calibrated cameras that were mounted in the instrumented vehicles. The EPS serves to obtain visual information about the traffic and road conditions surrounding the vehicle. All videos obtained from the vision system were synchronized and fused together for providing visual information about the vehicle cabin and the external environment. Figure 5.1 shows an example of how two separate videos obtained for the vehicle cabin and external environment were synchronized and fused together. Due to limitations with respect to the

equipment, the analysis presented in this thesis will rely heavily on information recorded by the GPS and the EPS system.



Figure 5.1: Visual Information from Internal and External Environment of the Vehicle
Synchronized and Fused Together

The test studies were conducted in the province of Quebec using two types of vehicle – a heavy duty transport truck and a hybrid electric vehicle (Toyota Prius). The video samples for each test drive were collected at 16Hz while the GPS data was sampled at 4Hz. Both the video acquisition systems and the GPS system were synchronized to ensure that reliable and accurate information about the vehicle and the environment were collected. The sample set for the study consisted of twenty-nine test drives, nineteen of which were obtained from the transport truck, and ten from the hybrid vehicle. The following raw data was available for each test drive:

1. Latitude and longitude of the vehicle, obtained from GPS, in degrees
2. Axial speeds of the vehicle in Earth Centered, Earth Fixed (ECEF) coordinate system, obtained from GPS, in cm/s
3. Video frame number
4. Video file for the external and internal environment of the vehicle

It should be noted that the test drives for the study were not conducted on any specific or fixed routes. The data was collected under naturalistic driving contexts and most

of the samples consisted of driving in various rural and urban roadways including trans-Canada highways and Quebec AutoRoute. However, the data collected from the hybrid vehicle consisted driving data mostly in urban and residential areas of Quebec. Since the driving conditions were not consistent (e.g. similar test route, similar duration, similar weather conditions, etc.), certain comparable sections of the test drives were isolated for further analysis. A detailed explanation on the data extraction and trimming process is provided in Section 5.2.

The following sub-section gives a brief overview of the ECEF coordinate system to provide more insight into the nature of data collected.

5.1.1 ECEF Coordinates

ECEF coordinates describes a Cartesian coordinate system that is used to define an object's location and is often found in GPS systems and satellites. ECEF is a fixed coordinate system with respect to the Earth, with its center of origin (0, 0, and 0) placed at the mass center of Earth. It often provides very precise information without having to model the Earth as an ellipsoid [19]. The coordinate frame is oriented in such a way that the Z axis points toward the North Pole while the X and Y axes are placed on the equatorial plane of the Earth, as shown in Figure 5.2. It should be noted that ECEF coordinate system is not dependent on the position or orientation of the vehicle in consideration.

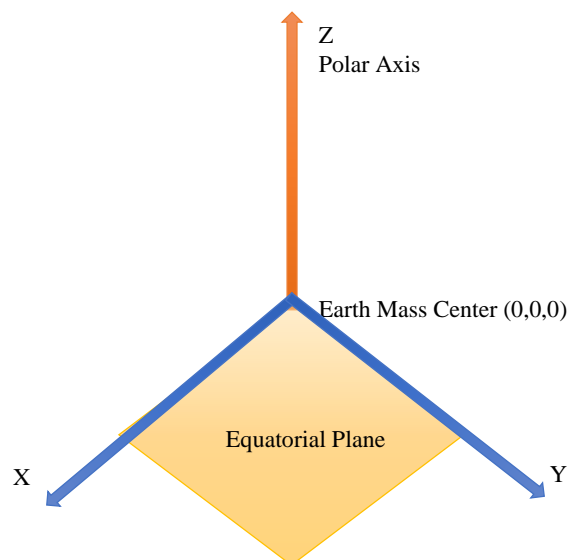


Figure 5.2: ECEF Coordinates

5.2 Raw Data Processing

Once the raw data was recorded for each test drive, the data was stored in an .xml file for further treatment and processing. ‘Analyse’ software is a data treatment and analysis toolbox, compatible with MATLAB software, developed by Mr. M. E. Kaszap from GRAME at Université Laval. This software package was used to initially process and filter the raw GPS data.

5.2.1 Latitude and Longitude Correction

Several preliminary steps are required before the raw latitude and longitude signals can be used for further analysis. The raw data channels containing recorded latitude and longitude values have missing data or “zeroes” which are not useful for analysis purposes. These false “zeroes” are caused by temporary GPS signal dropout during data recording. Thus, the zero values were required to be corrected first in the latitude and longitude channels using Analyse software. Figure 5.3 shows a sample set before and after the zero correction was applied to a sample latitude channel from a test drive.

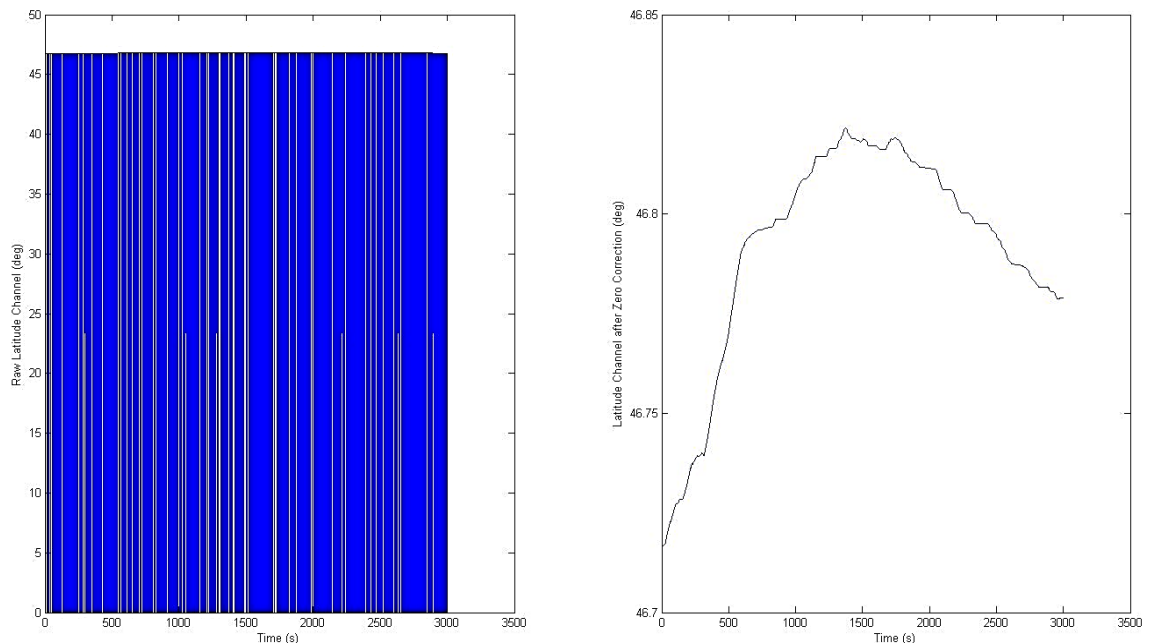


Figure 5.3: (a) Raw Signal from Latitude Channel (b) Latitude Channel after “Zero” Correction

To remove unwanted signal drops during GPS recording, when the vehicle is slowing down or coming to a stop, the latitude and longitude needed to be further corrected. The data was then passed through a moving window weighted average filter and a

polynomial filter to obtain the final corrected latitude and longitude values. Both the filters aid in noise reduction of the recorded signals. The moving window method helps to smooth the signal in order to exclude random noise in elevation measurements while the polynomial filter helps to reduce the effect of signal white noise in the recorded data. A completely corrected latitude channel is presented in Figure 5.4 for reference. This procedure is a standard pre-processing procedure developed by Université Laval for raw GPS data treatment for naturalistic driving applications and was applied to all the recorded test drives considered for this research.

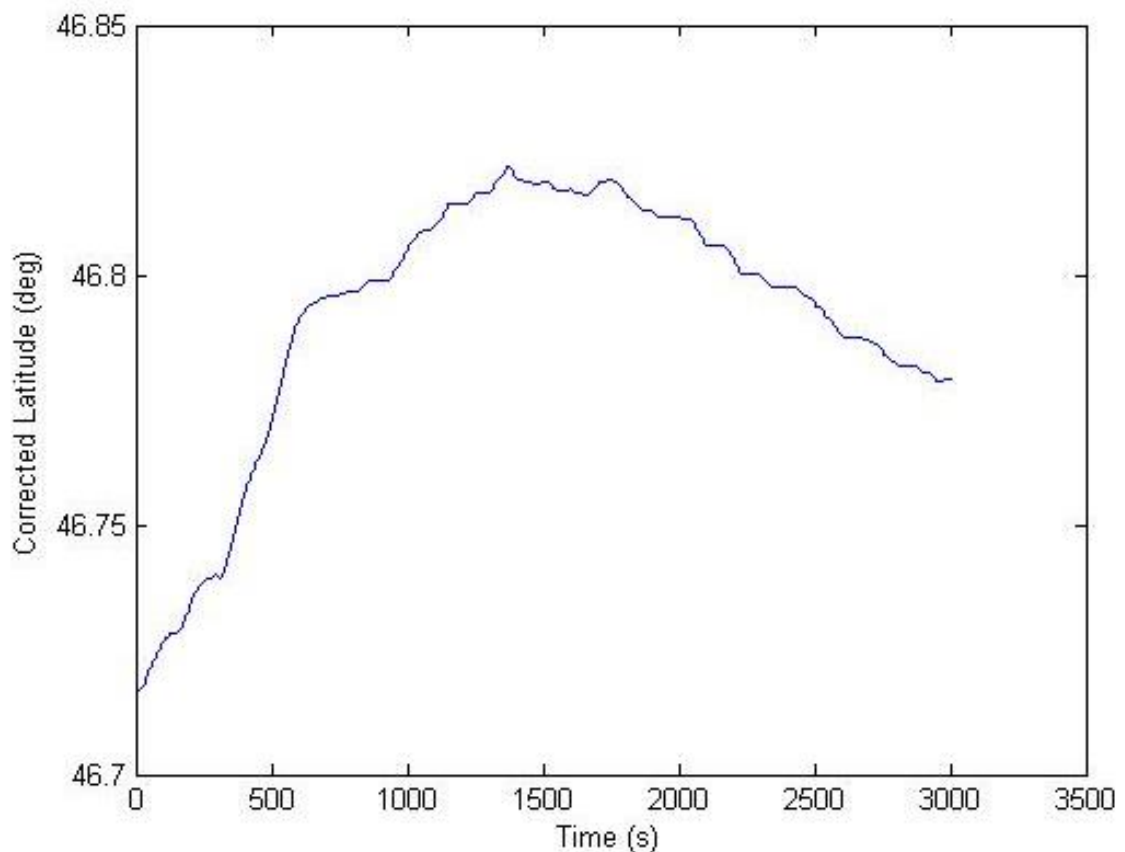


Figure 5.4: Example of a Final Corrected Latitude Channel obtained from raw GPS data

5.2.2 Vehicle Speed Determination

Once the corrected latitude and longitude values were obtained, the resultant vehicle speed, in km/h, was calculated from the ECEF coordinate speeds using Equation 29.

$$V_{res} = 0.036 \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (29)$$

where, v_x , v_y , v_z are the ECEF velocities in cm/s and 0.036 is the conversion factor for converting cm/s to km/h.

Figure 5.5 shows the vehicle speed pattern observed for a Prius test drive with a mean speed of 30.45 km/h and a maximum vehicle speed of approximately 106 km/h. A quick observation of Figure 5.5 reveals that the vehicle travelled mostly in urban or residential areas. This can be deduced by observing that the vehicle was travelling at speeds lower than 80km/h for majority of the test drive duration and performed numerous stops; both factors are indicative of residential or urban driving. On urban roadways, speed profiles fluctuate due to frequent stoppages because of traffic signals, stop signs, traffic congestion, etc.

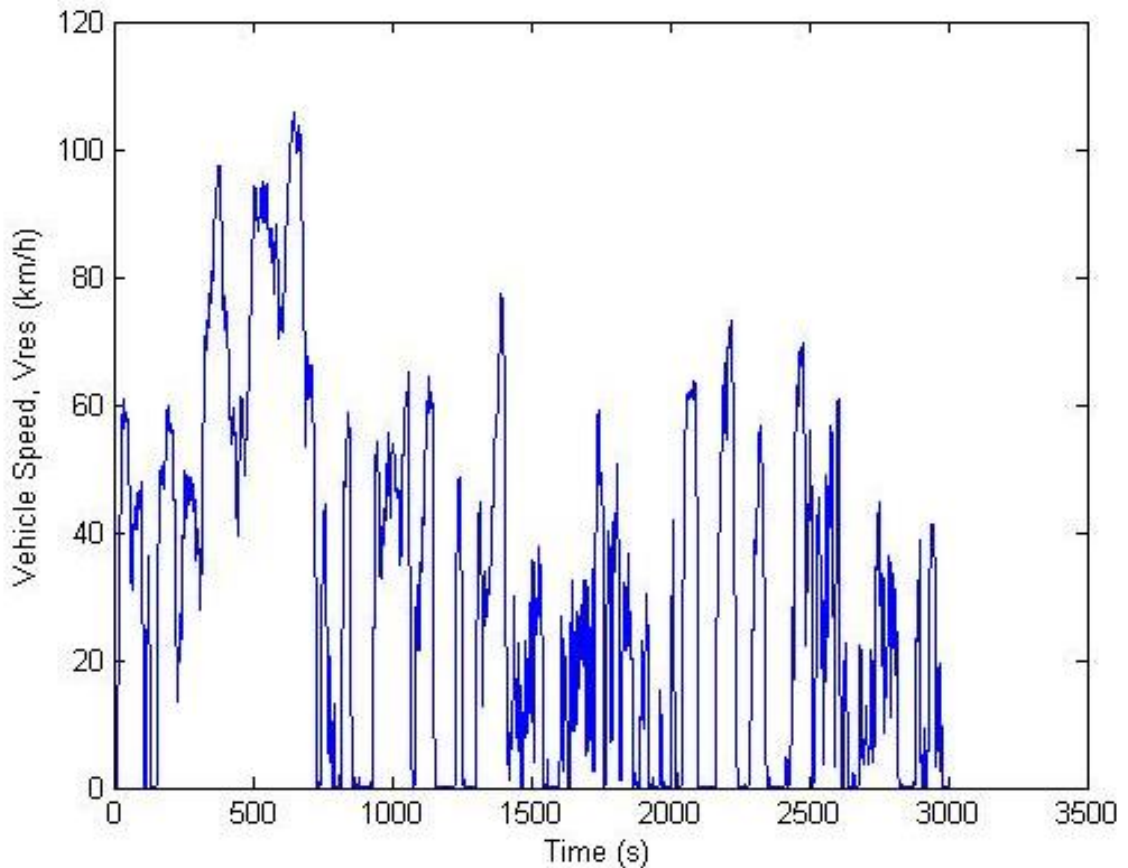


Figure 5.5: Speed Profile for a Prius Test Drive

5.2.3 Vehicle Acceleration Determination

The next step in data processing was to determine the vehicle acceleration. The acceleration of the vehicle, a , cannot be directly obtained from the raw data set, but rather has to be computed from the obtained vehicle speed in km/h. This is because the vehicles were not equipped with sensors to record the vehicle acceleration directly. The acceleration, in km/h^2 , between two successive points in the data set can be computed using Equation 30.

$$a = \frac{V_{res2} - V_{res1}}{(0.25)\left(\frac{1}{3600}\right)} \quad (30)$$

A simple numeric manipulation was used to compute the acceleration, knowing that the GPS data was sampled at 4Hz. Figure 5.6 presents plot of speed and acceleration for a sample Prius test drive. The graphs in Figure 5.6 only represent a section of the test drive to increase the resolution. It should be noted that there will be a certain degree of error associated with the calculated acceleration due to its indirect computation from vehicle speed. The ten sharp peaks in acceleration (positive and negative) seen in Figure 5.6(b) correspond to the ten instances in Figure 5.6(a) where the vehicle accelerated or decelerated over a very short period of time.

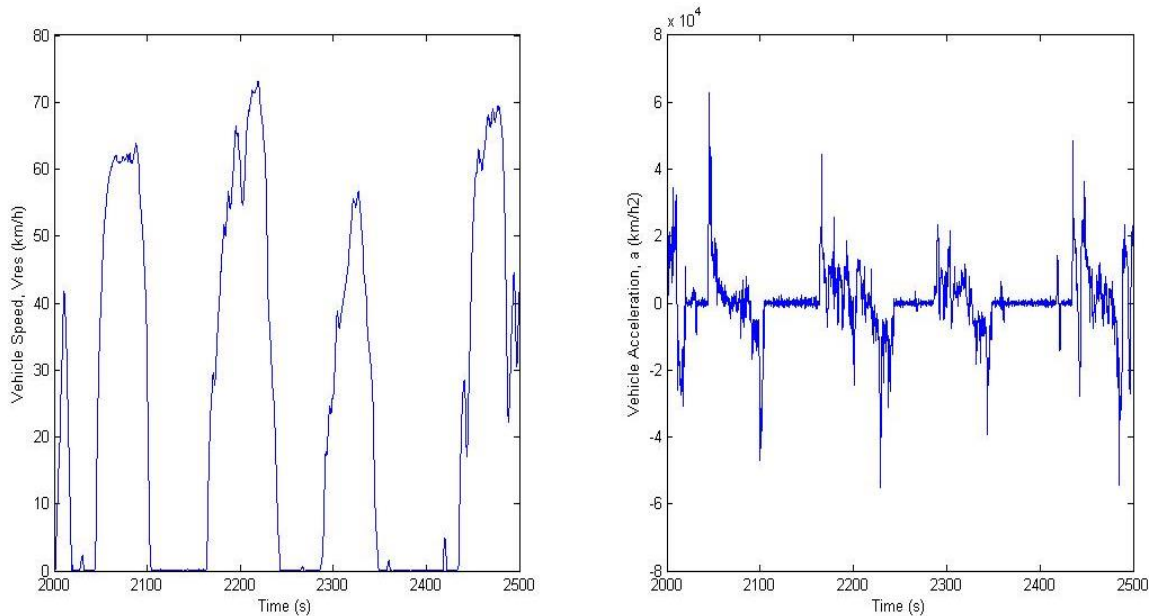


Figure 5.6: Comparison between Speed and Acceleration Profile of a Prius Test Drive (a) Speed Profile (b) Acceleration Profile

5.2.4 Distance Travelled

It was important to determine the distance that the vehicle travelled in each test drive, since each test drive had a different duration and route associated with it. The distance, d , travelled between two sets of latitude and longitude points was determined using the Haversine formula, as shown in Equation 31. The Haversine formula is used for navigation purposes to compute the shortest distance between two points on the Earth's surface. The formula is based on the assumption that the Earth is spherical in shape.

$$h = \sin^2\left(\frac{lat_2 - lat_1}{2}\right) + \cos(lat_1) \cos(lat_2) \sin^2\left(\frac{lon_2 - lon_1}{2}\right)$$
$$c = 2 \operatorname{atan2}(\sqrt{h}, \sqrt{1-h}) \quad (31)$$
$$\text{dist} = 6371c$$

where, lat is the latitude value, lon is the longitude value, and 6371 is the radius of the Earth in km. Note that (lat_1, lon_1) and (lat_2, lon_2) are consecutive data points in the data set. The Haversine formula is well conditioned for computations of distances even as small as a few metres [20]. An example of the total distance travelled by Prius during a sample test drive is presented in Figure 5.7.

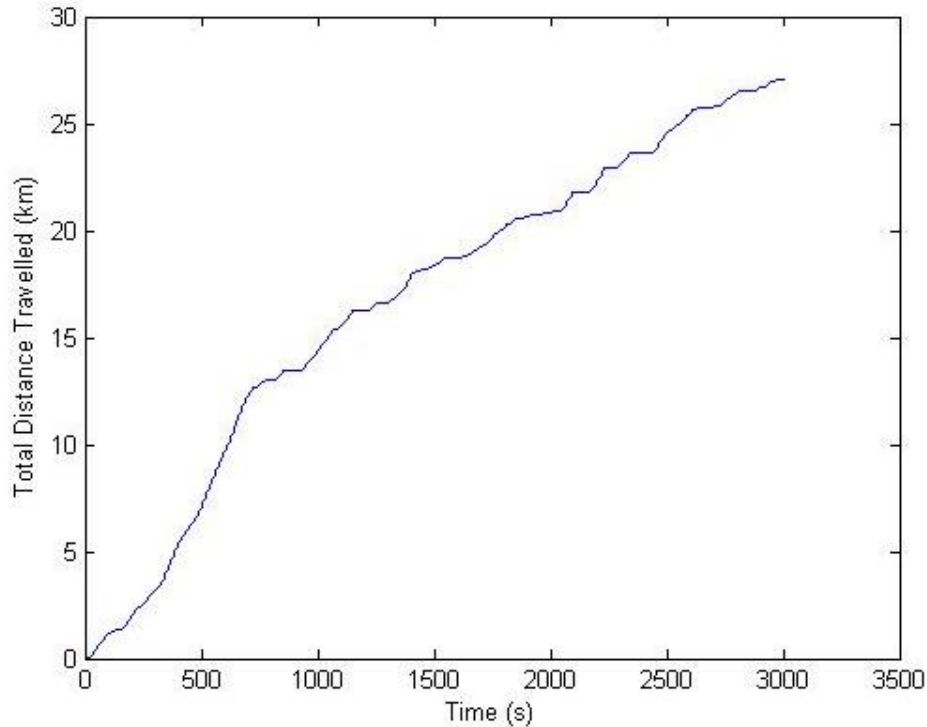


Figure 5.7: Total Distance Travelled for a Prius Test Drive

5.3 Data Extraction for Modelling

For the purpose of consistency in modelling, it is essential to trim the data set for each test drive such that the driving conditions are comparable. Urban/residential driving is very complex in nature due to the various levels of interaction of the driver with the environment. Moreover, most driving events are often observed in urban/residential settings where the vehicle frequently has to accelerate/decelerate, stop, perform sharp turns, etc. As a result, each test drive was trimmed accordingly to isolate portions of the test drive where the driver was travelling in urban/residential roadways, with a posted speed limit of 50 km/h. The data was further trimmed to exclude all data points where the vehicle was stationary (e.g. at stop signs, intersections, etc.). Scenarios where the vehicle remains stationary do not provide valuable information for initial model development. Rather it diminishes the quality of the data set since it might skew or alter the trends observed.

For instance, a stationary vehicle does not have any speed value associated with it. If these instances were to be included in the final data set, it would provide a misrepresentation of the average speed value of the vehicle during that test drive. Once all the test drives were trimmed to include scenarios where the vehicle was moving in urban/residential roadways, some preliminary statistical calculations (as described in Chapter 3) were performed on each test drive set to obtain the final set of variables for analysis and modelling purposes. Over 38,000 data points or observations were obtained for the 29 test drives. The set of input variables extracted from each test drive is presented in Table 5.1. The data set consisted of twenty nine test drives and can be found in Table A.1 (Appendix A). Thus, the obtained data set consisted of quantitative variables which were continuous in nature, i.e. can assume any numerical value along a continuum. The only exception was the variable, V_{10} , which was essentially a ratio or percentage where a value of zero indicated an absence of the feature or measurement under consideration.

Table 5.1: List of Input Variables

Variable Description	Variable	Units
Mean Vehicle Speed	\bar{V}	km/h
Standard Deviation of Vehicle Speed	σ_V	km/h
Percentage of Test Drive where Vehicle Speed Exceeds 10% of the Posted Speed Limit	V_{10}	n/a
Standard Deviation of Vehicle Acceleration	σ_a	km/h ²
Mean Vehicle Acceleration (Positive)	\overline{Acc}	km/h ²
Standard Deviation of Positive Acceleration	σ_{Acc}	km/h ²
Mean Vehicle Acceleration (Negative)	\overline{Brk}	km/h ²
Standard Deviation of Negative Acceleration	σ_{Brk}	km/h ²
Total Distance Travelled	D_{tot}	km

The variables presented nine in Table 5.1 will serve as the basis for analysis and classification of driving performance for this thesis. The variables related to standard deviation help to measure the variation in a given variable for each test drive. The calculated vehicle acceleration was divided into two separate variables, *Acc* and *Brk*, based on their respective positive or negative signs. A positive acceleration value was used as an indicator of the driver accelerator pedal position in the vehicle, while a negative acceleration value was used as an approximate indicator of driver brake pedal position in the vehicle. It should be noted that these are only approximate measures since no direct measurement of the pedal positions were recorded from the vehicles. Another important thing to note from Table 5.1 is the V_{10} variable. V_{10} helps to evaluate the percentage of time the vehicle was travelling 10% above the posted speed limit in each test drive. This variable will serve as a measure of risky driving performance. Although 10% (55 km/h) is a conservative approach for sedans travelling in urban/residential areas, recall that the data set consists of two different classes of vehicles: a transport truck and a passenger car. Moreover, if the model is to be extended to include other roadway types with different speed limits, 10% provides a more consistent method for evaluation rather than absolute speed values for evaluating risky driver performance.

5.4 Outlier Detection

Before the data set can be processed and used for modelling and analysis purposes, it is crucial to check for outliers present in the data set. Outliers are essentially data points or observations that are significantly different than the rest of the data set. Such values have an effect on the statistical model and the ANN model by generating unwanted errors, and thus are often excluded from the data set for analysis [17]. A simple check for outliers was conducted by determining the mean and standard deviations for each variable in the data set using Equation 32.

$$x_{check} = \begin{cases} \text{not outlier,} & x_{ij} - \bar{x}_i < 2\sigma_{x_i} \\ \text{outlier,} & x_{ij} - \bar{x}_i > 2\sigma_{x_i} \end{cases} \quad (32)$$

where x_{ij} is any variable i belonging to observation j , \bar{x}_i is the mean of variable i , and σ_{x_i} is the standard deviation of variable i .

A given variable i in observation j is an outlier if x_{check} is greater than two standard deviations. A check for outliers was performed for each variable in a given observation. Based on the obtained results, the variable D_{tot} of observation 23 was found to be an outlier (Table A.1 in Appendix A). As a result, the entire observation, 23, was excluded from the data set to reduce the associated errors. Thus, the final data set consists of 28 observations of driving parameters which will be used henceforth for further analysis.

5.5 Data Standardization

Once the final data set of 28 observations was obtained for multivariate analysis, it was very important for the variables to be transformed in a manner such that each variable had a comparable effect on statistical analyses and neural networks. Since the variables in the data set were measured in different scales and units, it was very important to transform the data so that variables with higher magnitudes or scales (e.g. \overline{Acc}) did not outweigh the effect of variables with smaller magnitudes (e.g. V_{10}). Data normalization and data standardization are two linear transformation techniques that are widely used for transforming or scaling the data set so that the relative effect of different variables can be interpreted in a meaningful manner irrespective of their units. The data set for this work was standardized instead of being normalized within a specific range (usually -1 to 1). Standardization was primarily done because the development of a factor model requires

the data set to be standardized. Moreover, instead of normalizing the data set between a defined range, standardization scales each variable such that the transformed variable has zero mean with unit variance. The general expression for standardizing any variable, x , is presented in Equation 33.

$$x_{std} = \frac{x - \bar{x}}{\sigma_x} \quad (33)$$

The standardized final data set is presented in Table A.2 (Appendix A) for reference. Each variable can be converted back to its original value and unit for ease of interpretation of the model.

CHAPTER 6

DRIVER PERFORMANCE CLASSIFICATION

Driving is a complex task affected by various factors related to the driver, vehicle, and environment. No generally accepted mathematical model, or defined set of variables exists for providing comprehensive information about driving behaviour and performance. This lack of direction leads to questions requiring further in-depth investigation of the data set. What is the importance of the variables present in the data set for this research? How can the data set be used to draw some inferences about driver performance in urban/residential areas? These are some of the questions that will be addressed through the course of this chapter. The objective is to classify driver performance based on the variables presented in the final data set (Table 5.1), as described in Chapter 5. Although the set of variables used for this analysis is limited in nature, an attempt is made to provide meaningful interpretation of the information available about driver performance and behaviour.

The following chapter is divided into two major sections. The first section focuses on an unsupervised hierarchical clustering technique to determine the natural subsets within the driving data set. Emphasis will be placed on determining the optimum number of clusters or groups which can provide meaningful interpretation of driver performance. Once the specific groups are identified, an attempt is made to develop a supervised classification neural network that can establish the relationship between the input variables and the identified groups. This topic will be the focus of the second section of this chapter and will provide a basis for driver performance evaluation under similar circumstances (i.e. urban/residential roadways with a posted speed limit of 50 km/h). At this stage, it should be noted that all analyses and modelling will be conducted using the standardized data set presented in Table A.2 (Appendix A). Minitab Statistical Software v16.1.1 and MATLAB 2014a were used extensively for carrying out the analyses presented in this chapter.

6.1 Unsupervised Classification

The driving data set relevant for this research has no priori information available about different driving behaviour, style, or performance. Hence, before a model for evaluating driver performance can be developed, it is important to determine the outcome

or dependent variable for the data set. This can be achieved by performing unsupervised clustering analysis, as described in Chapter 3.2. Through this technique, underlying patterns are identified which in turn help in evaluating the different groups of driving performance.

The hierarchical clustering algorithm with Ward's linkage was used to perform an initial cluster analysis on the standardized data set. The first step in the process was to generate a dendrogram to determine the observable patterns in the data set using Ward's linkage method. The initial dendrogram for the analysis is shown in Figure 6.1.

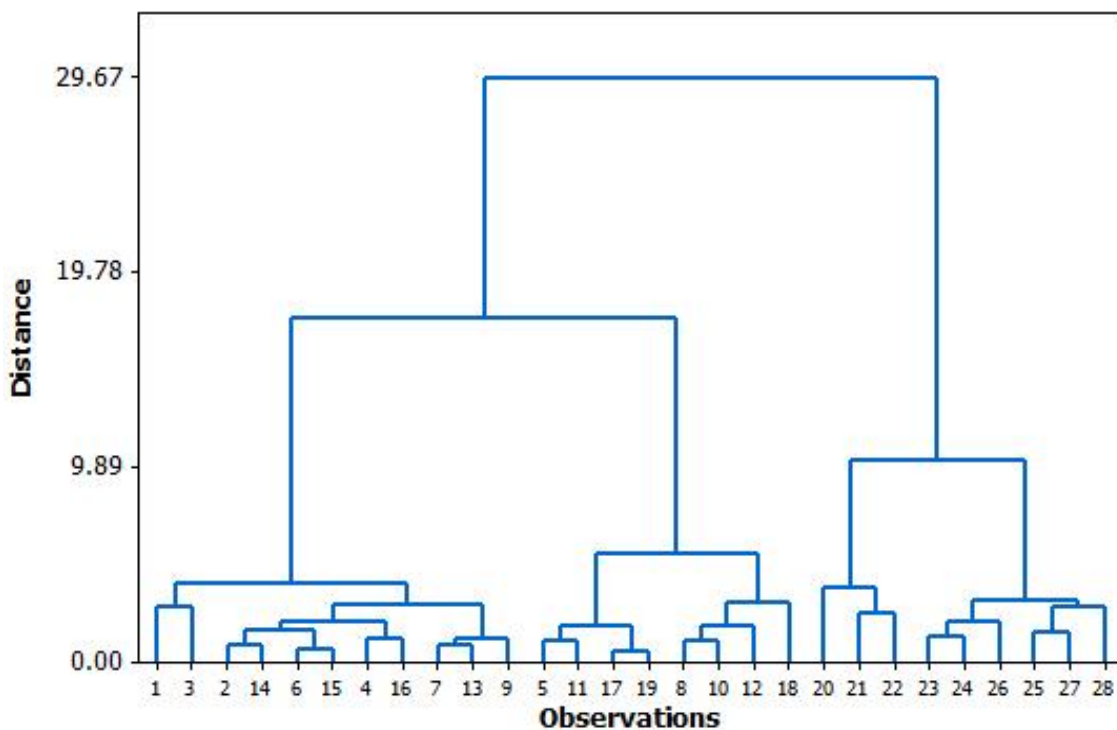


Figure 6.1: Dendrogram of Driving Data Set for Initial Cluster Analysis

As mentioned in Chapter 3, the hierarchical agglomerative algorithm starts by assigning each observation to a certain group. The algorithm progresses till one single group, containing all observations, remains. There was no information about the classes present in the data set prior to the investigation. Thus, one of the biggest challenges for cluster analysis was to determine the optimum number of clusters that represent the natural division in the data set. This aspect of statistical analysis has been widely explored by researchers from various disciplines. However, the use of such analysis techniques has

been limited in the field of engineering. Real data sets have a high level of complexity and no generally acceptable technique exists to estimate the number of clusters [21]. The interpretation of results is dependent on the analyst. Milligan and Cooper [22] conducted an extensive study for evaluating over 30 techniques to determine the number of clusters present in a simulated data set. Based on the findings of the study, the Calinski and Harabasz (C-H) criterion was considered as one of the most efficient methods for determining optimal number of clusters.

The C-H criterion is based on the within cluster variance (SS_W) and between cluster variance (SS_B), and can be computed using Equations 34 and 35 [23]. A cluster is considered well defined when the between cluster variance is large and the within cluster variance value is small.

$$SS_W = \sum_{i=1}^k \sum_{x \in c_i} d_{x, \bar{c}_i}^2 \quad (34)$$

$$SS_B = \sum_{i=1}^k n_i d_{\bar{X}, \bar{c}_i}^2 \quad (35)$$

where, d is the Euclidean distance, k is the number of clusters, x is an observation, \bar{c}_i is the cluster centroid of the i^{th} cluster, n_i is the number of observations in cluster c_i , and \bar{X} is the overall mean of the data set. Thus, the C-H can be computed using Equation 36, where the optimal number of clusters is determined by the highest C-H value [23].

$$C - H = \frac{SS_B(n - k)}{SS_W(k - 1)} \quad (36)$$

The C-H criterion is used to evaluate the quality of the classes and to determine the compactness or ‘tightness’ of each class. For the given data set, the C-H values were evaluated using ten initial clusters. To determine the number of optimum clusters, a plot similar to the scree plot can be generated using the calculated results. Hence, the C-H values, computed using Equation 36, were plotted against the number of clusters, as shown in Figure 6.2. A quick observation of Figure 6.2 reveals four optimal clusters for the data set (highest value of C-H). The interconnecting lines serve as a visual aid for determining

trends, no intermediate values are possible between two consecutive discrete points (indicated as points in Figure 6.2).

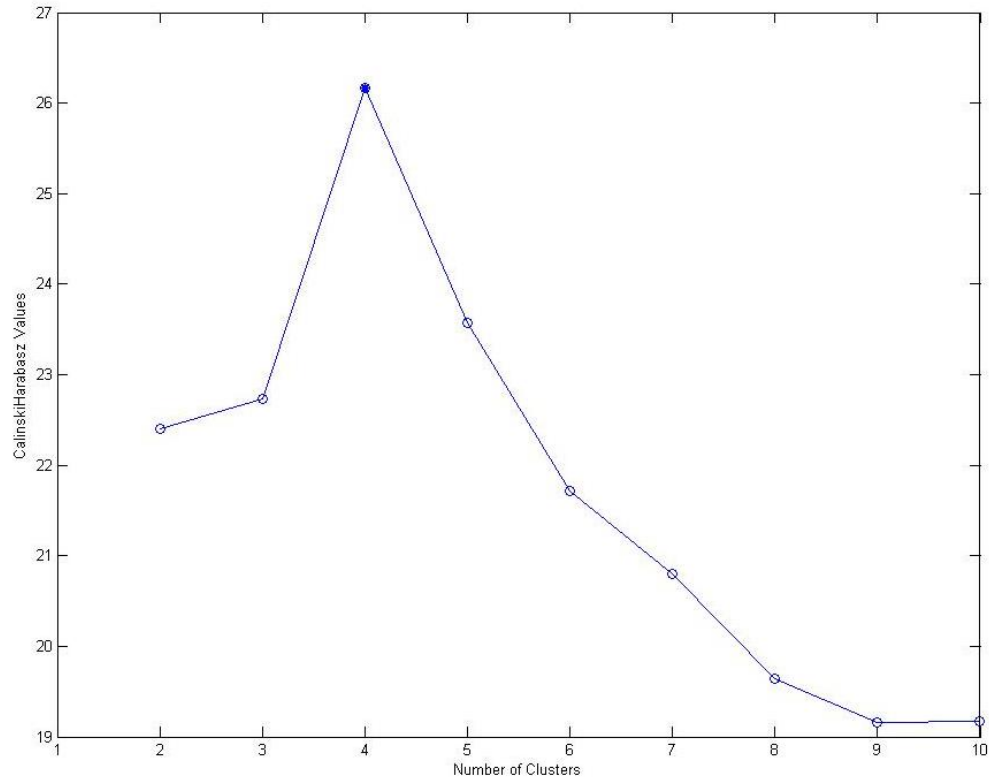


Figure 6.2: Evaluation of Optimal Number of Clusters using C-H Criterion

It is important to further validate the results obtained from the C-H criterion. Another criterion that has received significant attention for optimal cluster determination is the Gap criterion. This criterion can be applied to any clustering technique and distance measure [21]. A general graphical method, known as the “elbow” method, is used extensively where some error criterion is plotted against the number of clusters. The “elbow” occurs at the most significant drop in error. Tibshiran *et al.* [24] proposed a method to formalize the ‘elbow’ location by determining the number of clusters with the largest Gap value. The term “Gap” is used since the method focuses on comparing the distribution of the data set with a reference distribution, which is discussed below in further details. The optimal number of clusters is defined at the point where the Gap value is the largest [24]. The Gap value can be formally represented through Equation 37 [24].

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k) \quad (37)$$

$$\text{where, } W_k = \sum_{i=1}^k \frac{1}{2n_i} D_i$$

where, E_n^* denotes the expectation from a sample size under a sample size of n from the reference distribution, W_k is the pooled within cluster sum of squares around the cluster means, and D is the sum of pairwise distances for all points in cluster i .

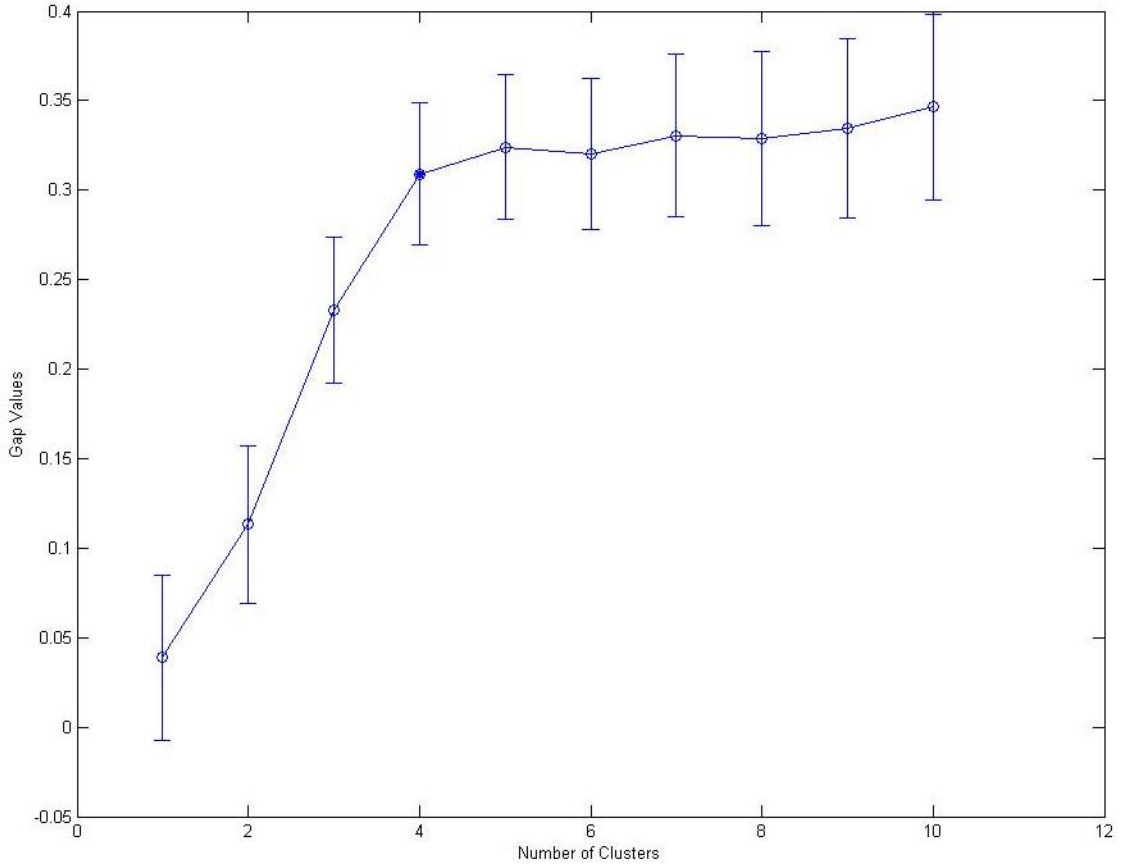


Figure 6.3: Evaluation of Optimal Number of Clusters using Gap Criterion

The expected $E_n^*\{\log(W_k)\}$ value was determined through Monte Carlo sampling from a reference distribution using MATLAB 2014a software. The reference distribution was consecutively generated using a uniform distribution over a box aligned with the principal components of the data [24]. According to Tibshiran *et al.*, this method of reference distribution takes into account the original distribution of the multivariate data set. The plot for Gap values vs. number of clusters is shown in Figure 6.3 for analysis. Based on the results obtained using Equation 37, four optimal clusters were determined for the data set as well. The numerical results for the Gap criterion are presented in Table B.1 (Appendix

B) for reference. Since both the criteria, C-H and Gap, indicated four as the optimal number of clusters for the given data set, a cluster analysis with four final partitions were conducted using the method outlined in Chapter 3.2.

The results of the final cluster analysis are best summarized using the dendrogram in Figure 6.4. The distance between the classes is plotted on the y-axis, and data points or observations are plotted on the x-axis. The four classes, along with their class members, are highlighted with the aid of different colours in Figure 6.4. Tables B.2 and B.3 (Appendix B) present information about the individual assignment of groups for each data point or observation as well as the centroids of each class. A summary of the results, along with the number of observations for each class, obtained from the cluster analysis is also presented in Tables 6.1 and 6.2.

The centroid of a cluster is essentially a vector mean of the variables present in a given cluster. From the analysis, Class 1 and Class 2 were determined to be the two largest classes. Moreover, the lower the SS_w value, the more compact the individual clusters (Table 6.1). Another important thing to consider for the analysis is the distance between any two cluster centroids (Table 6.2). A large difference between any two cluster centroids indicates that the members present in each of the classes are significantly different from each other. For instance, Class 2 and Class 3 had the largest distance between the centroids followed by Class 2 and Class 4.

Once the natural subdivisions in the data set were determined using unsupervised clustering algorithms, the next challenge was to use the obtained results to provide meaningful interpretation of the different classes. The different classes help to evaluate the driver performances based on the driver, vehicle, and environment characteristics. This problem was tackled by developing a supervised ANN network that had the capability of modelling the relationship between the input variables and each class. The following section provides detailed description of the ANN model developed for driver performance classification.

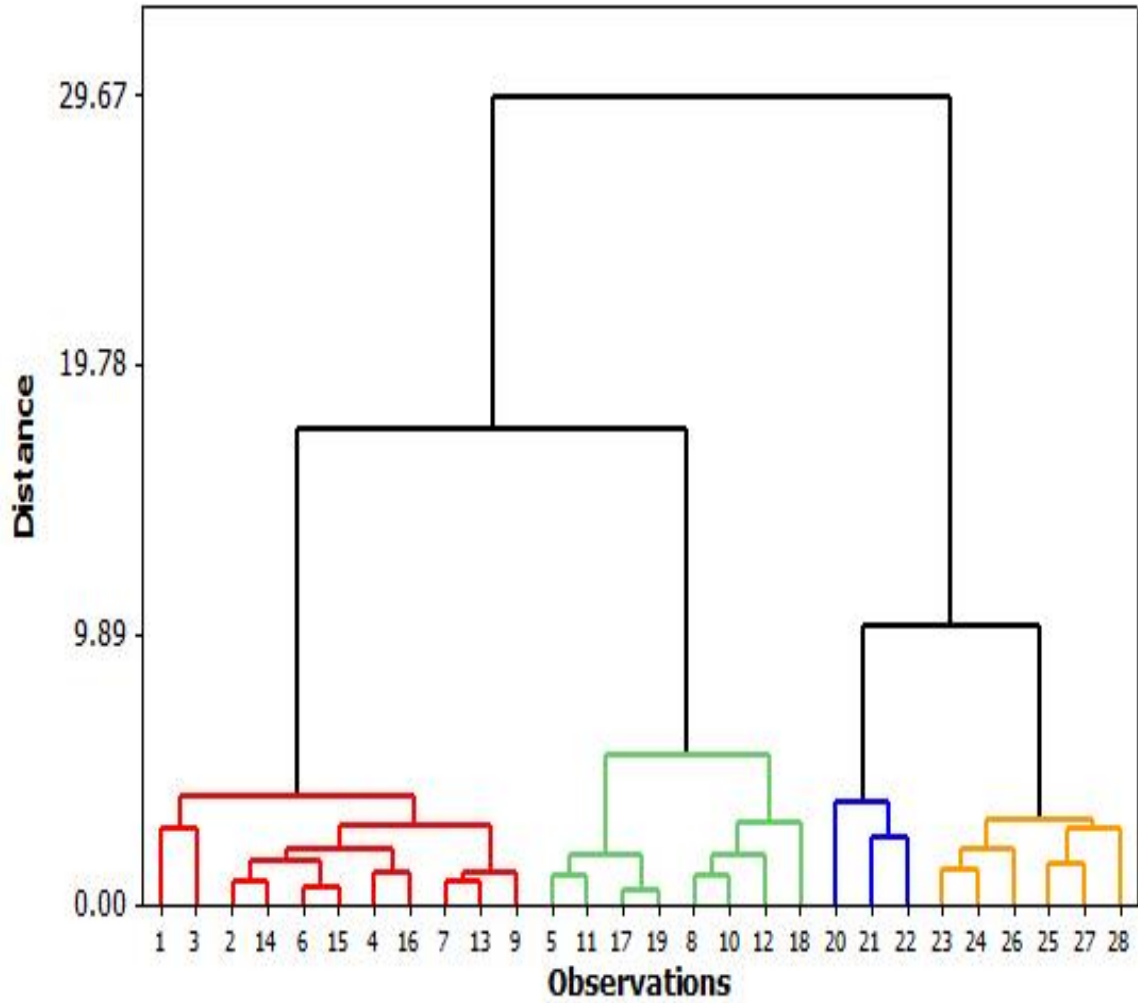


Figure 6.4: Dendrogram Showing Assignment of Data Set into Four Classes

Table 6.1: Results of Hierarchical Agglomerative Clustering with Four Classes

	Number of Observations	Within Cluster Sum of Squares (SS _w)	Average Distance from Centroid	Maximum Distance from Centroid
Class 1	11	18.66	1.20	2.34
Class 2	8	18.11	1.41	2.30
Class 3	3	9.95	1.80	2.15
Class 4	6	12.23	1.40	1.81

Table 6.2: Distance Between Cluster Centroids

	Class 1	Class 2	Class 3	Class 4
Class 1	0.00	3.15	3.78	4.55
Class 2	3.15	0.00	5.95	5.72
Class 3	3.78	5.95	0.00	3.90
Class 4	4.55	5.72	3.90	0.00

6.2 ANN Model for Classification of Driver Performance

The objective of this section is to determine the relationship between the different classes and input parameters with the aid of a supervised classification network. The first step involves transforming the different classes, obtained from unsupervised hierarchical agglomerative cluster analysis, into binary target values. To demonstrate this process, the target values for a few selected observations from the data set are presented in Table 6.3.

Table 6.3: Example of Transforming Classes to Binary Target Values

Sample	Class 1	Class 2	Class 3	Class 4
1	1	0	0	0
5	0	1	0	0
20	0	0	1	0
24	0	0	0	1

Using the binary target values, and the concepts introduced in Chapter 4, a supervised MLP feedforward backpropagation network utilizing an LM learning algorithm was designed for modelling the data set. Since neural networks are considered as “black boxes”, the objective of developing the network was to establish relationships between the input and output variables in a manner such that one can identify the class to which a particular sample of driving data belongs when only the input parameters are provided. Before a supervised ANN network can be trained using the data set, it is very important to determine a network architecture that will be able to reliably map the input parameters to the target classes. Thus, the entire modelling process can be described comprehensively in three distinct stages: network architecture, network training and validation, and network results.

6.2.1 Network Architecture

The design process for an ANN architecture is an iterative process and various network configurations (i.e. number of processing units, number of hidden layers, activation functions, learning algorithms, etc.) were implemented before selecting a final network architecture. The architecture of the network that provided the best network performance value is presented in Figure 6.5. The final designed ANN for modelling driver performance classification has a 9-12-4 architecture. As seen from Figure 6.5, there are twelve hidden neurons or processing units connecting the input layer to the output layer with the aid of weights and biases. Bias values are connected to each hidden and output processing units respectively.

As mentioned earlier, the tansigmoid activation function was chosen for both the hidden and the output processing units with an operating range of $[-1, 1]$ due to the reasons described in Chapter 4. One important thing to note is that the network was presented with standardized input variables similar to the cluster analysis in order to obtain a better performance for the classification network. A summary of the final network architecture is presented in Table 6.4 for reference.

Table 1.4: Final ANN Architecture for Driver Performance Classification

Network Architecture Description	Value
Input Parameters	9 (Standardized Continuous Values)
Target Classes	4 (Binary Values)
Hidden Layer(s)	1
Hidden Processing Units	12
Activation Function (Hidden and Output Layers)	Tansigmoid $[-1,1]$

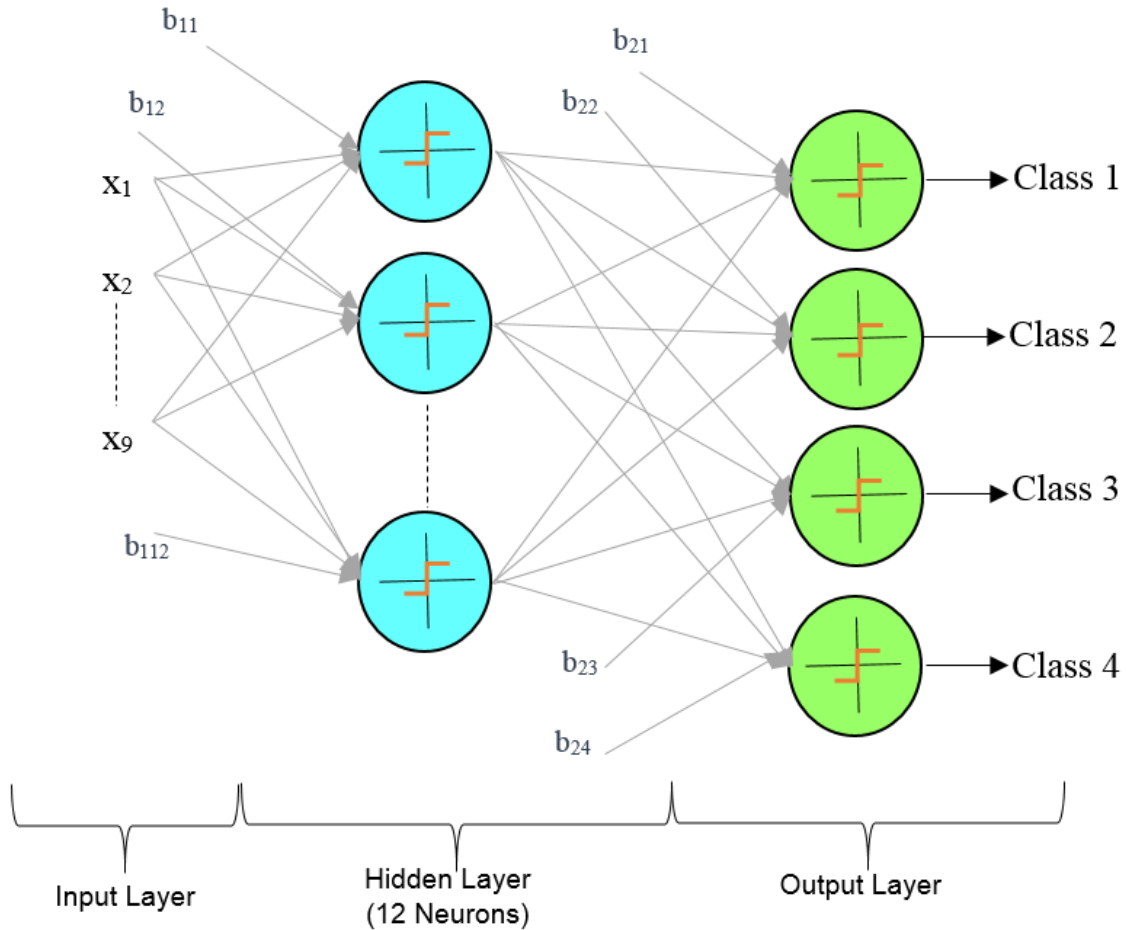


Figure 6.5: Neural Network Architecture for Classifying Driver Performance

6.2.2 Network Training and Validation

The network training phase consisted of three stages: training the network, validating the network, and testing the network. Initially, the data set was randomly divided into three different subsets in the following manner: - 70% for training, 15% for validation, and 15% testing. The training subset data was then used to train the network by computing, updating, and adjusting the corresponding weight and bias values. The LM technique was used as the training algorithm for updating the network weights due its efficient performance when compared to other algorithms (Chapter 4.4.1). Simultaneously, the validation data set was used to monitor the network performance and generalization capabilities of the network and will be discussed in detail in the following section. The purpose of the test data set was to compare the quality of the classification model once the network is trained and validated.

Once the data was divided into three subsets, random initial weights and bias values were initiated by the network to commence the network training process with the aid of known target values. The network was trained using batch learning mode where the weights and biases were updated after each epoch. MSE was used for evaluating the performance of the network. The network was trained with 70% of the 28 data sets for 11 epochs. The network training was terminated when the validation performance failed to decrease in 6 (default value for MATLAB Neural Network Toolbox) consecutive iterations in MATLAB software. An overview of the ANN training phase for classification of driver performance, along with the network performance results are presented in Table 6.5. The final network performance values (training, testing, and validation) were less than 0.1 thus indicating good overall network performance.

Table 6.5: ANN Training Parameters and Performance Results

Network Training Parameters & Results	Value
Network Training	Batch Mode
Learning Algorithm	LM
Division of Data Set	70-15-15 (Random)
Network Performance Criterion	MSE
Epochs	11
Training Performance	0.008
Validation Performance	0.021
Test Performance	0.029

6.2.3 Network Results

After completion of the network training phase, the next important step was to analyze the network results and determine the effectiveness of the network in modelling the driving data set. Figure 6.6 shows the performance (training, validation, and testing) of the network at each epoch during the training phase. The training phase began by reducing the training and validation error at each successive epoch. The best performance of the network was obtained at epoch 5 when the validation error was at its minimum. When the validation error started to increase, it indicated that the model was overfitting the training data set instead of providing a generalized result over the training and validation data sets.

Thus, the final network weight and bias values were recorded at epoch 5, where the validation error was a minimum. Moreover, the testing and validation performance curves followed a similar trend and reached a minimum error at the same epoch, as seen in Figure 6.6, which indicated that the division of the data set used was adequate for modelling purposes [25]. This method of obtaining network weights and biases at the minimum validation error to prevent overfitting is often known as the early stopping method. Although the training error continued to decrease beyond epoch 5, obtaining the network results based on the lowest training performance would have resulted in a model which would have been very specific to the training data set and might have included associated noise inherent to that specific data set.

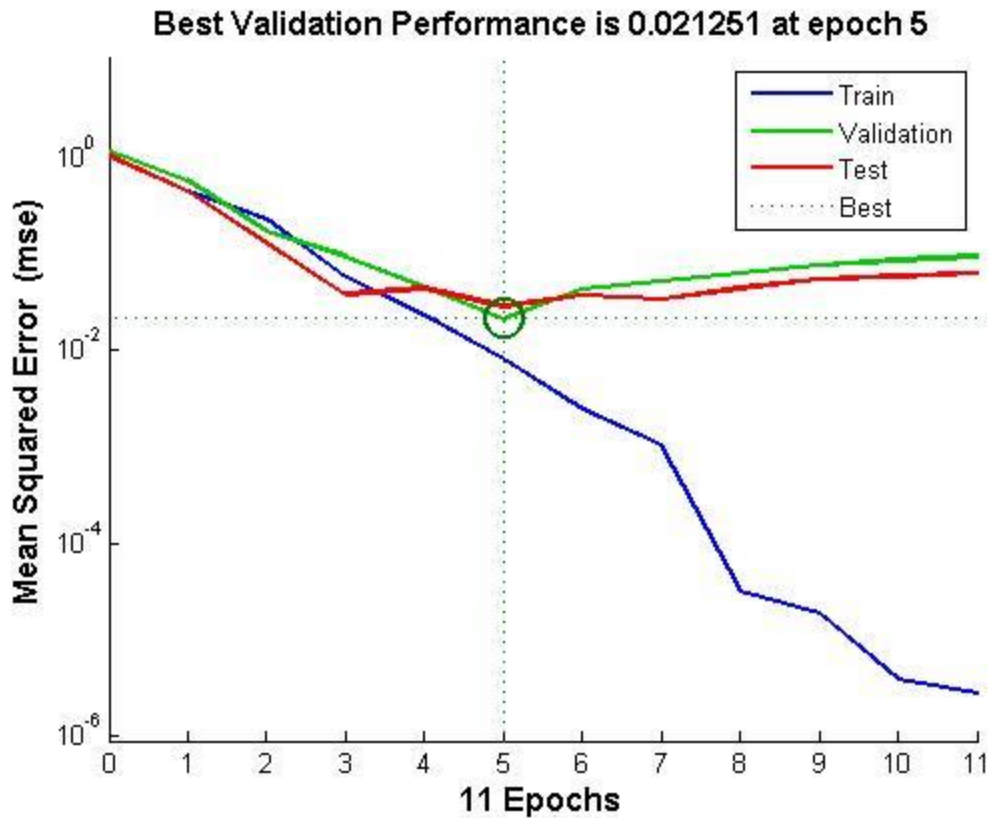


Figure 6.6: ANN Performance Curves for Training, Validation, and Testing

To further validate the network performance, the error (E_j) values between the target and the calculated network outputs (Equation 24) can be viewed on an error histogram, as shown in Figure 6.7. Furthermore, Table C.1 (Appendix C) lists all the error values obtained from the ANN network for each class for reference. An error histogram

presents a quick visualization of the error distribution presented by the network. In Figure 6.7, the training data is represented in blue, the validation data is represented in green, and the test data is represented in red, respectively. The error histogram shape follows an approximate normal distribution curve with the highest errors observed near the zero region (indicating a healthy network). This shape further helps to validate that the trained network is robust and performs in a satisfactory manner.

The next step was to see how well the trained ANN model was able to classify the parameters based on supplied target classes. Figure 6.8 presents a confusion matrix which helps to show the network’s behaviour by building a square matrix showing how the network helped predict each class based on its corresponding target value.

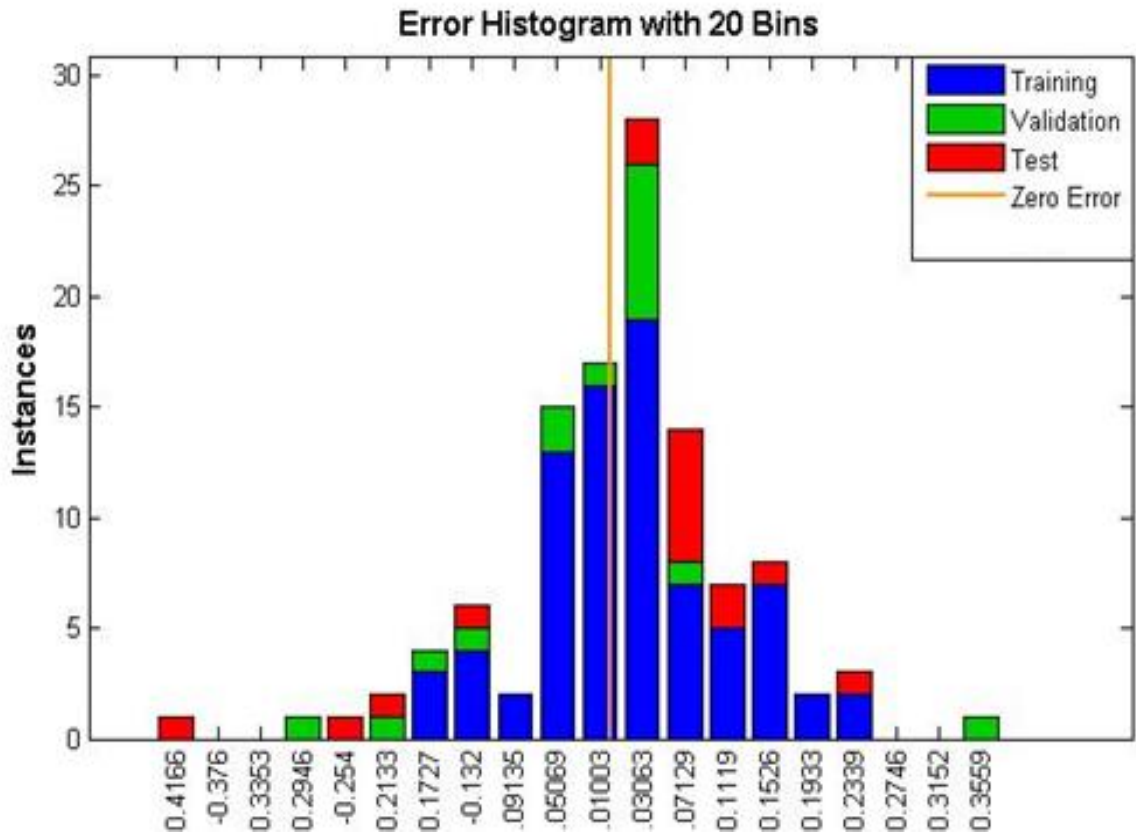


Figure 6.7: Error Histogram for Designed ANN Network

The major diagonal of each matrix indicates the percentage of observations the model was able to classify correctly. Any value observed at any other location (indicated in red) of the matrix indicates a false classification or error introduced by the ANN (Figure

6.8). It should be noted that there were no observations for 2, 3, and 4 in the validation subset. Any observation for class 2 was also absent in the test subset. As a result, the confusion matrix shows a “NAN” value where such observations were absent. The primary reason for the occurrence of missing observations is that the final data set was fairly small, consisting of only 28 observations. When the data set was divided randomly in to three subsets, the validation set and the test set had no sample observation for either class 2, 3 or 4. To avoid this issue, it is recommended that the data set size be increased to include more observations for each class. However, due to the limited test drive data available during the course of this research, such a solution could not be implemented. Apart from the missing observations, the trained ANN network was able to classify the driver performance classes accurately when compared to their target values. An overall network accuracy of 95.6% was achieved using the data set which was fairly reasonable for this modelling purpose and can be shown through the regression plot of the network presented in Figure C.1 (Appendix C).

The regression plot is a measure of the fit of the classification model, where the network output is plotted against the provided target values as determined from cluster analysis. All results presented for the network further help to demonstrate the reliability and accuracy of the developed model. Furthermore, the ANN model also helps to validate and confirm that the results obtained from the hierarchical agglomerative clustering technique provide insightful information about the presence of natural subsets or classes within the data set. If cluster analysis was not able to partition the data in to meaningful classes, the performance of the ANN model would not be satisfactory either.

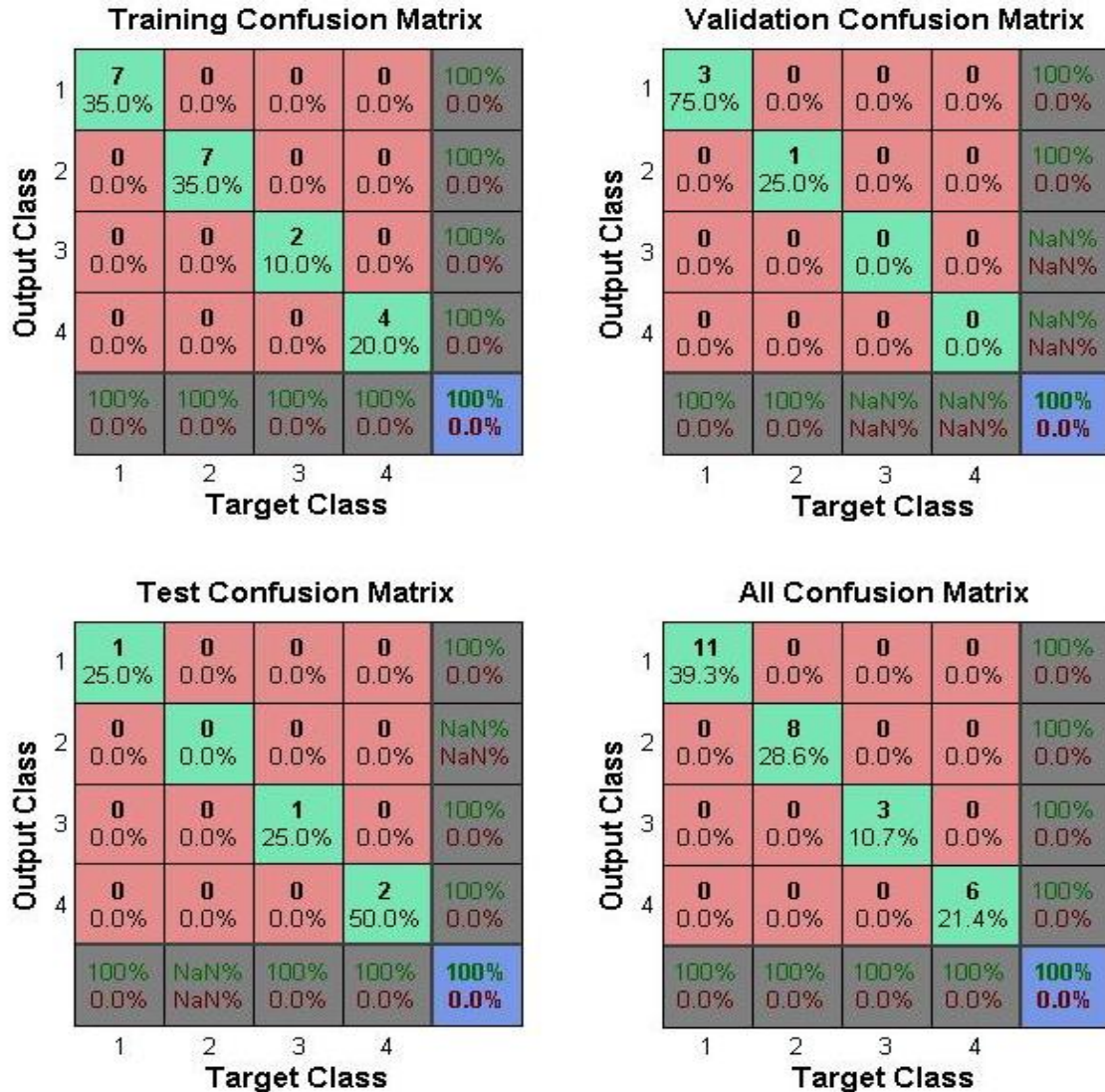


Figure 6.8: Confusion Matrix for Driver Performance Classification using ANN

6.3 Identification of Significant Variables

Both the cluster analysis and the developed ANN helped establish the different classes to evaluate the driver performance. The ANN network developed in Section 6.2 is a robust model for reliably classifying the driver performance into four distinct classes. However, none of the methods presented so far gave an insight in to the variables that had the highest influence for a given class. Neural networks have powerful prediction and classification capabilities; however, it is often challenging to interpret and rationalize the results obtained from the network. Hence, it is critical for the purpose of this research to derive inferences and identify meaningful interpretations of each of the classes identified

by the network. Since the developed classification model is non-linear in nature, it is necessary to determine the effect an individual input variable has on the entire system. This in turn will help to identify the variables that describe the different categories of driver performance observed in the data set. One way to achieve this is by analyzing the weight of the network for discovering trends or patterns in the data set. One disadvantage of this method is that MLPs often contain multiple hidden units which may make the process confusing and labour intensive.

Since very little information is known about the structure of the data set, another possible way of extracting meaningful information from the network is to compute the partial derivative of each output class with respect to each input variable presented to the network. This calculation determines the sensitivity of the output with respect to the individual variables, providing information about driver performance variability between the different classes. The larger the sensitivity of a given variable in each class, the greater the effect that variable has in determining the outcome of driver performance for a given class.

Determining the partial derivatives of outputs with respect to inputs is similar to the method presented in Chapter 4.4.1, and is similar to backpropagating error through the network. Instead of backpropagating the error, the partial derivative of the output with respect to the input can be computed using Equation 38 [17] by utilising the concept of chain rule of differentiation.

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial h} \frac{\partial h}{\partial x} \quad (38)$$

where, z is a network output, x is a network input, and h is a hidden processing unit.

The calculations can be further simplified by expressing the derivative of the activation function with respect to its output units [17]. The derivative of the hyperbolic tangent activation function can be expressed as $(1-x^2)w$. Therefore, Equation 38 can be rewritten as Equation 39 [17] for the developed ANN for driver performance classification.

$$\frac{\partial z_j}{\partial x_k} = \sum_{i=1}^{12} (1 - h_i^2) w_{i,k} (1 - z_j^2) w_{i,j} \quad (39)$$

where, h_i is the activation of processing unit j , $w_{i,k}$ is the weight from input unit k to hidden unit i , $w_{i,j}$ is the weight from hidden unit i to output unit j , and 12 is the number of hidden units used in the developed ANN.

Since the objective is to determine the sensitivity of variables for each class, one possible method is to determine the sensitivity based on class centroids identified from the hierarchical agglomerative cluster analysis performed on the data set (Table B.3, Appendix B). The class center values essentially provide a good representation of the member of a given class. Table 6.6 show the results of the partial derivatives computed for each class using Equation 39. Computation of the partial derivatives essentially provides an overview for sensitivity analysis of the driving data set.

The partial derivative values presented in Table 6.6 provide insight into which variables have the greatest sensitivity for a given output class. The weights of the ANN network utilized for the computation of the derivatives are presented in Tables C.2 and C.3 (Appendix C) for reference. The purpose to calculating sensitivity in this case is to determine the variables which might lead to a change in class membership based on a small change. A positive value indicates that a small change in the variable will move the data point or driver performance for a given test drive closer to the input class. Similarly, a negative value indicates that a small change in the variable will move the data point or driver performance away from a given class [17]. Another important thing to realize is that all the values presented in Table 6.6 have a small magnitude. This is because the centroid of each class (indicating an overall representation of each class membership) was utilized for conducting the analysis. Thus, smaller values of derivatives or gradients are noticed.

Table 6.6: Sensitivity Analysis Results for Driver Performance Class

Variable	Class 1	Class 2	Class 3	Class 4
$\frac{\partial z_i}{\partial \bar{V}}$	0.021	0.017	-0.051	0.141
$\frac{\partial z_i}{\partial \sigma_V}$	0.33	-0.003	-0.031	-0.152
$\frac{\partial z_i}{\partial V_{10}}$	0.25	-0.056	-0.047	-0.237
$\frac{\partial z_i}{\partial \sigma_a}$	0.339	0.0278	0.101	0.0124
$\frac{\partial z_i}{\partial Acc}$	0.342	0.0392	0.082	-0.003
$\frac{\partial z_i}{\partial \sigma_{Acc}}$	0.346	0.0412	0.043	0.015
$\frac{\partial z_i}{\partial Brk}$	0.263	-0.079	0.043	-0.3434
$\frac{\partial z_i}{\partial \sigma_{Brk}}$	0.263	-0.006	0.043	0.041
$\frac{\partial z_i}{\partial D_{tot}}$	0.23	-0.078	0.063	-0.018

The significant variables for each class is highlighted in boldface in Table 6.6 for reference. The following trends or observations can be summarized based on the results obtained from the sensitivity analysis:

- Class 1: For a given test drive, class 1 is most sensitive to changes in variations in overall vehicle acceleration and positive vehicle acceleration (indicative of accelerator pedal position) in a given test drive.
- Class 2: For a given test drive, class 2 is most sensitive to changes in the mean braking acceleration and total distance travelled.
- Class 3: For a given test drive, class 3 is most sensitive to changes in the variation in overall vehicle acceleration
- Class 4: For a given test drive, class 4 is most sensitive to changes in mean deceleration (indicative of brake pedal position) of any given test drive.

These results can be used to further determine the characteristics observed in each class for evaluating driver performance. The following chapter utilizes the dimensionality reduction technique to further gain insight into the different classes of driving performance.

CHAPTER 7

DRIVER PERFORMANCE CLASS DESCRIPTION

In Chapter 6, analysis has revealed that driver performance based on various driving parameters or behaviour indices can be classified into four different classes. An ANN model was developed to classify the driver performance by establishing relationships between the driving parameters and classes. Furthermore, a sensitivity analysis was performed to determine the factor most sensitive for a given class. However, no formal description has been provided for each of the driver performance categories. The objective of this chapter is to perform a factor analysis using the principal component method in an attempt to reduce the dimensionality of the input data set so that a formal description of each class can be obtained. The utilization of this method can provide a better understanding of the type of driving performance observed. Factor analysis will help to interpret the driving data with fewer variables which in turn will aid in interpreting the characteristic inherent in each driving performance category. Based on these results, the different driving classes will also be ranked in terms of the driver risk levels. The ranking will help identify the types of driving performance that contribute to risky driving scenarios with potential for damage and for injuries.

7.1 Data Dimensionality Reduction

The first step in developing a factor model is to determine the number of principal components or factors that will be required to explain the majority of the variation in the data set. As explained in Chapter 3.3.2, the basis for principal component analysis relies on the computation of eigenvalues from the variance covariance matrix. A simple scree plot of eigenvalues vs. number of components was plotted to determine an appropriate number of components for the analysis.

A quick observation of Figure 7.1 reveals a sharp drop in the eigenvalue magnitude for the first three components. An “elbow” is also observed at component three. The objective of using the principal component method is to determine a suitable number of factors (less than the number of variables in the data set) that can help explain the phenomena observed in the data set. The eigenvalues computed from the variance covariance matrix, along with the percentage of variation explained by each component,

are presented in Table 7.1. Thus, based on the results obtained from Figure 7.1 and Table 7.1, a factor model for the data set was built using three factors (components). Based on the selection of three factors for the factor model, the factor model explains 87.3% of the variation present in the data set. This is sufficient to gain an insight into the different driver performance classes.

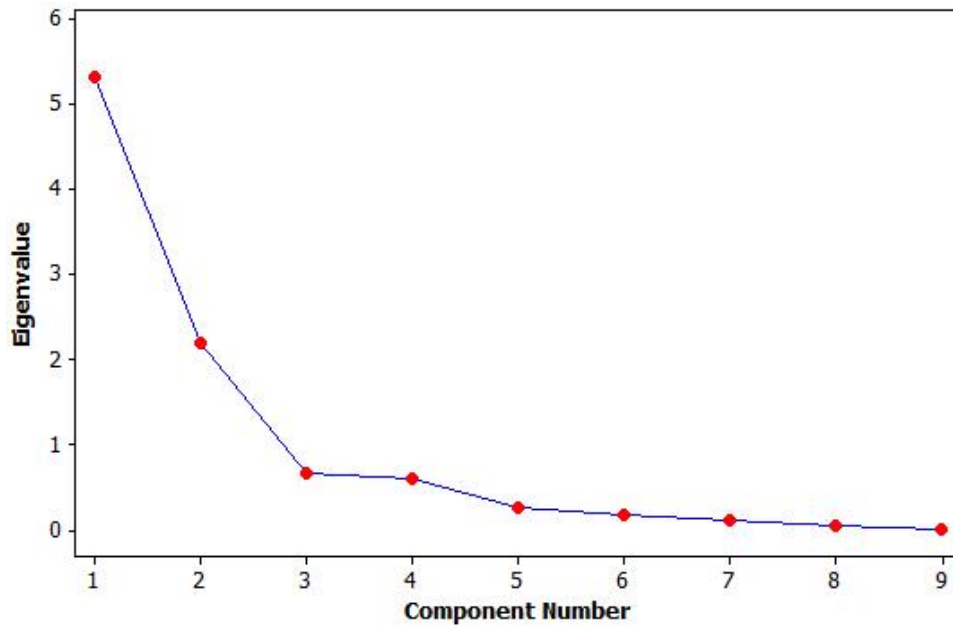


Figure 7.1: Scree Plot for Determining the Number of Factors for Factor Model

Table 7.1: Eigenvalue Analysis of the Covariance Matrix for Driving Parameters

Component	Eigenvalues	Proportion (%)	Cumulative (%)
1	5.31	56.9	56.9
2	2.18	23.4	80.3
3	0.654	0.070	87.3
4	0.597	0.064	93.7
5	0.254	0.027	96.4
6	0.181	0.019	98.4
7	0.109	0.012	99.5
8	0.039	0.004	100
9	0.003	0.000	100

7.2 Factor Model for Driver Performance Classification

Once the number of factors were determined for factor analysis, the factor loading (β_{ij}) of each input variable in the original data set were computed using Equations 20 and 21. The purpose of the factor loading is to express the new factors (determined from dimensionality reduction using principal component method) in terms of the nine input variables present in the data set. This not only helps to reduce the complexity of the data set but also aids in interpreting the different driver performance classes. The factor loading values, along with their corresponding communality values, are presented in Table 7.2.

Table 7.2: Factor Loading Values and Communalities for Factor Model

Variable	Factor 1	Factor 2	Factor 3	Communality (%)
\bar{V}	-0.131	-0.772	0.598	97.0
σ_V	0.892	-0.163	-0.073	82.8
V_{10}	0.824	-0.418	0.029	85.5
σ_a	0.947	0.267	0.135	98.5
\overline{Acc}	0.933	0.240	-0.021	92.8
σ_{Acc}	0.901	0.271	-0.013	88.5
\overline{Brk}	-0.454	0.698	0.398	85.2
σ_{Brk}	0.873	0.232	0.275	89.1
D_{tot}	0.290	-0.751	-0.124	66.4

Therefore, using the results from Table 7.2, Factor 1 can be expressed using the original nine driving parameters, as shown in Equation 40. Factor 2 and Factor 3 can be expressed in a similar manner as well. The calculated factor values corresponding to each observation or test drive is presented in Table D.1 (Appendix D).

$$\begin{aligned} \text{Factor 1} = & -0.131\bar{V} + 0.892\sigma_V + 0.824V_{10} + 0.947\sigma_a + 0.901\sigma_{Acc} \\ & - 0.454\overline{Brk} + 0.873\sigma_{Brk} + 0.290D_{tot} \end{aligned} \quad (40)$$

Since the goal is to determine a measure or criteria for interpreting each factor, the significant factor loading values ($|\beta_{ij}| > 0.5$) are presented in boldface in Table 7.2. Also, the communality value represents the percentage of variation explained by the three factors. This value serves as a good indicator of how well the model fits the driving data set. For

instance, 98.5% of the variation in vehicle acceleration (σ_a) can be explained with the aid of the three chosen factors. The communality value is analogous to the regression value for predicting the desired results. For the developed factor model, some variables have a higher communality value than others which aids in explaining the quality of the factor model in terms of each of the nine variables present in the driving data set. For instance, D_{tot} is the variable with the lowest communality value explaining only 66.4% of the variation in the original data set. Another way of determining the quality of the model is to compute the residual matrix using Equation 19. The residual matrix is based on computation of the correlation matrices and provides errors generated by the factor model. The resultant values of the residual matrix (R) is presented below.

Residual Matrix, R

$$= \begin{pmatrix} 0.029 & -0.003 & -0.026 & 0.004 & 0.018 & 0.035 & -0.032 & -0.032 & -0.036 \\ -0.003 & 0.172 & -0.016 & -0.034 & -0.040 & -0.063 & 0.023 & 0.010 & -0.048 \\ -0.026 & -0.015 & 0.145 & -0.017 & -0.053 & -0.062 & -0.035 & 0.035 & -0.118 \\ 0.004 & -0.034 & -0.017 & 0.015 & 0.022 & 0.015 & 0.002 & -0.011 & 0.029 \\ 0.018 & -0.40 & -0.053 & 0.022 & 0.072 & 0.052 & 0.042 & -0.068 & 0.087 \\ 0.035 & -0.063 & -0.062 & 0.015 & 0.052 & 0.115 & 0.016 & -0.077 & 0.067 \\ -0.032 & 0.023 & 0.023 & 0.002 & 0.042 & 0.016 & 0.148 & -0.046 & 0.191 \\ -0.032 & 0.010 & 0.010 & -0.011 & -0.068 & 0.077 & -0.046 & 0.109 & -0.052 \\ -0.036 & -0.048 & -0.048 & 0.029 & 0.087 & 0.067 & 0.191 & -0.052 & 0.336 \end{pmatrix}$$

The error magnitudes of matrix R are considerably low. Ideally, for a good factor model, the residual values should be as close to zero as possible. There are very few values in the residual matrix that have a magnitude greater than 0.1. Therefore, based on the communality and error values, it can be deduced that the three factors model helps to model the driving data set to an appropriate level of accuracy. However, it should be realized that the factor model has a significantly lower accuracy than the ANN model. The purpose of the factor model is not to classify driving performance based on inputs, but rather to provide insight into the meaningful interpretation of each driving performance class.

7.3 Interpretation of Factors

Before the driving performance for each class can be interpreted in a meaningful manner, it is important to understand what each of the three factors represent. The following inferences can be made based on the results presented in Table 7.2:

- Factor 1: Factor 1 indicates the vehicle has a higher tendency to speed over 10% of the posted speed limit with higher mean positive accelerations (indicative of vehicle accelerator pedal position). Under such circumstances, high variations of vehicle speed, vehicle acceleration, positive acceleration, and negative acceleration are also observed. These insights might give an indication about the level of risky behaviour demonstrated by the while driving in an urban or residential area.
- Factor 2: Factor 2 shows that for a given test drive, if the vehicle stops or brakes frequently while travelling in an urban or residential setting (indicated by the high mean deceleration value), the mean speed of the vehicle and the total distance covered by the vehicle in that particular test drive decreases. The observations from this factor are intuitive. If the driver is decelerating or stopping the vehicle frequently, the vehicle will not be travelling at relative high speeds or will not be travelling a larger distance within a certain period of time.
- Factor 3: Factor 3 is a factor which is primarily determined by the mean speed of the vehicle in a given test drive. Factor 3 does not provide much information with respect to other variables for interpreting the trends or underlying characteristics observed in the driving data set.

Since, through dimensionality reduction, the number of variables predicting the driver performance classification can essentially be reduced to three categories, the results of the classification can be presented in a three dimensional (3D) space, as shown Figure 7.2. A clear distinction between each class boundary is seen for each of the four classes for evaluating driver performance. Figure 7.2 further verifies the results obtained from the cluster analysis performed in Chapter 6 by ensuring that no two classes have intersecting or overlapping boundaries.

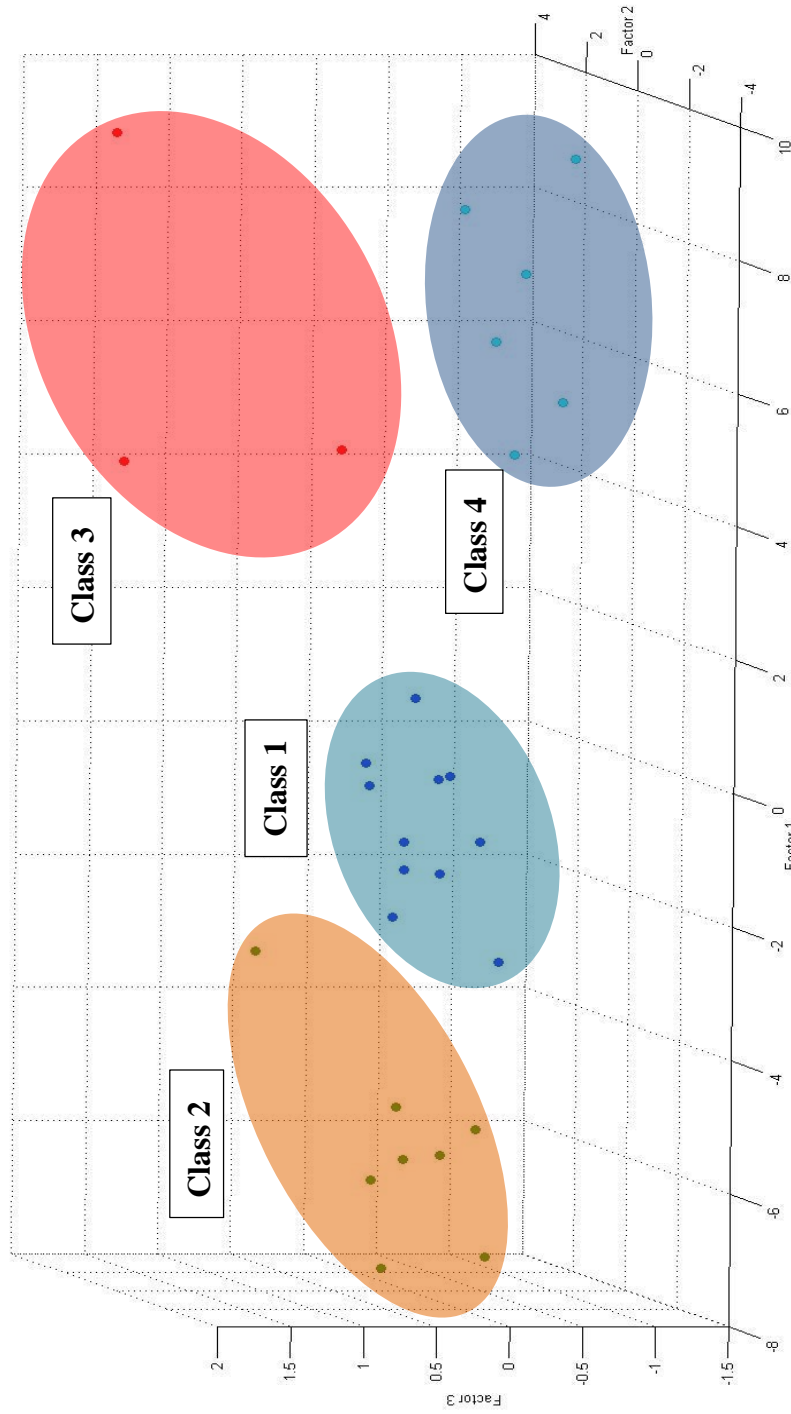


Figure 7.2: Classification of Driver Performance based on Factor Model

Once the significance of each factor is determined, it is now possible to provide some insight into the member characteristics of each class. Table 7.3 provides a comprehensive description of the classes inferred from the results from the factor model. Table 7.3 ranks the different classes based on risky driving behaviour. A value of 1 is indicative of the class which demonstrates the highest level of risk based on driving performance, as compared to the other classes.

Table 7.3: Class Description Based on Factor Model Results

		Min	Max	Description	Level of Risk
Class 1	Factor 1	-3.54	0.46	No speeding, low positive accelerations	4
	Factor 2	0.26	3.30	Frequent braking, shorter distances	
	Factor 3	-1.19	0.20	Lowest mean speed	
Class 2	Factor 1	-7.44	-2.87	No speeding, low positive accelerations	3
	Factor 2	-3.32	-0.28	Less frequent braking, longer distances	
	Factor 3	0.05	1.11	High mean speed	
Class 3	Factor 1	3.99	9.08	Speeding, high positive accelerations	1
	Factor 2	2.09	3.31	Frequent braking, shorter distances	
	Factor 3	0.13	1.70	High mean speed	
Class 4	Factor 1	4.65	9.38	High acceleration, speeding	2
	Factor 2	-2.96	-0.87	Less frequent braking, longer distances	
	Factor 3	-0.63	-0.05	Low mean speed	

To obtain better understanding of the type of performance represented by each of the four classes, the results from Table 7.3 can be summarized as follows:

- Class 1: The driving performance in Class 1 can be characterized by low vehicle accelerations, and longer travel distances. Also, the driver performs less frequent vehicle decelerations or stops during the course of the drive and the mean speed (20km/h – 35km/h) of the vehicle remains well below the posted speed limit (50

km/h) in an urban or residential driving environment. Moreover, the vehicle does not go beyond 10% of the posted speed limit of the roadway. This type of driving performance demonstrates the lowest level of risk when compared to all four classes.

- Class 2: The driving performance characteristics for the members of Class 2 are similar to the characteristics observed in Class 1, except the mean vehicle speed. The members of class 2 have a higher mean vehicle speed (35km/h – 48km/h) compared to the members in Class 1. The mean vehicle speed is very close to the posted speed limits of urban and residential roadways. Higher vehicle speeds indicates that the driver performance observed in Class 2 is more risky than the driving performance in Class 1.
- Class 3: The driving performance in Class 3 is characterized by high positive accelerations and high vehicle mean speeds. Members belonging to this class also have a tendency to drive at a speed which is 10% above the posted speed limit (for urban and residential roadways). The driving performance characteristics in Class 3 demonstrated the highest level of risk among all four classes.
- Class 4: The driving performance in Class 4 is characterized by high positive vehicle acceleration, but low mean vehicle speeds. Moreover, the driver also has a tendency to drive at a speed above 10% of the posted speed limit. In terms of risk, the members of this class have a rank of 2 among the four classes.

Thus, using the modelling techniques highlighted in Chapters 3 and 4, the driving performance was classified into four different categories and ranked based on their level of risky behaviour. The information was extracted using a simple set of variables containing information about the vehicle speed, vehicle acceleration (positive and negative), and vehicle travel distance. The following chapter presents further discussions on the results obtained and provides some concluding remarks for the work that was conducted for this research.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

Driving performance is dependent on the task demands dictated by the traffic environment. Task demands of the driver include factors such as the vehicle speed/acceleration, vehicle performance, and road structure. Exploratory statistical techniques and ANNs have been used as the backbone of the work presented in this thesis to determine and classify driver performance in different categories based on the observed level of risky behaviour. The research not only helps to outline a methodology for modelling and classifying driver performance, it also utilizes statistical tools and techniques that complement the developed model for interpreting meaningful results. Moreover, the statistical techniques also help to validate the ANN model results as well. A summary of the key findings from this research is presented in Table 8.1.

Table 8.1: Summary of Results for Driver Performance Classification

Class	Description	Significant Factor(s)
Lowest Level of Risk (Class 1)	No speeding, low positive accelerations	Variations in overall vehicle acceleration and positive vehicle acceleration
	Frequent braking, shorter distances	
	Lowest mean speed	
Less Risky (Class 2)	No speeding, low positive accelerations	Mean braking acceleration and total distance travelled
	Less frequent braking, longer distances	
	High mean speed	
Risky (Class 4)	High acceleration, speeding	Mean vehicle deceleration
	Less frequent braking, longer distances	
	Low mean speed	
Highest Level of Risk (Class 3)	Speeding, high positive accelerations	Variation in overall vehicle acceleration
	Frequent braking, shorter distances	
	High mean speed	

It is interesting to note that the vehicle acceleration/deceleration plays an important role in determining the different driver performance categories. The variation in acceleration levels in each test drive is also critical in determining the levels of risk associated with driving.

One of the key challenges faced during the development phase of the work was the data processing stage. It is very important to extract relevant information from the raw data channels and trim the data set to obtain meaningful results. The data was trimmed and processed to include naturalistic driving scenarios only under an urban setting. Moreover, the raw data had to be processed in several stages before it could be used to develop reliable models using the techniques outlined in this thesis. The hierarchical clustering algorithm was successful in partitioning the driving data into four distinct classes without any overlaps. Also, the developed ANN classified the driver performance with an overall accuracy of 96.5%. The results obtained from the analysis provide a comprehensive overview of driving performance under naturalistic driving contexts even though limited information was available pertaining to the driver and the vehicle environment. The purpose of factor analysis was to gain further understanding of the different performance classes observed in the data set rather than providing an alternative modelling technique. The results obtained from the factor analysis model complemented the results from the unsupervised clustering model and the ANN model. The factor model also helped to determine the level of risk associated with each class by reducing the dimension of the original driving data set.

Only 28 data test drives were available for conducting the analysis. It is recommended that the model is built using a larger data set to capture the various trends in driver behaviour which affect the driving performance. As a result, increased levels of risky behaviour can lead to higher chances of getting involved in traffic events which, in turn, might jeopardize traffic safety.

8.2 Future Work

Due to limitations in time and resources, the analysis was only limited to a transport truck and a passenger car. If there were more sample test drives for each vehicle, separate models could be developed for different types of vehicles and a comparison could be made between the different levels of performance and associated risk levels observed in drivers of different vehicles. The work presented in this thesis also lays the foundation for further analysis in the following areas:

- The model can be extended to include other physiological factors related to the driver such as eye tracking, heart rate variability, driver fatigue, and environmental factors (e.g. traffic information, weather information) to develop a more comprehensive and detailed overview of the driving performance.
- The situation awareness of any driver depends on the level of the driver interaction with the vehicle and the traffic environment. Hence, the driver needs to adapt to the demands of the driving task continuously. The work presented in this thesis can be extended to include the situation awareness of the driver and the associated levels of perceived risks by integrating variables which reflect the changes in the driver task demand.
- Naturalistic driving in different roadway configurations (e.g. highways, rural roads, etc.) can also be incorporated into the model once sufficient data is available for analysis.
- The presented methodology can be used as an assessment tool for fleet management services to evaluate and identify key driver characteristics that lead to risky driving behaviour. This tool can further be used to develop tailored training programs for professional drivers to effectively reduce the number of traffic collisions.

REFERENCES

- [1] T. Canada, "Canadian Motor Vehicle Traffic Collision Statistics 2010," Transport Canada, Ottawa, 2012.
- [2] D. Laurendeau, S. Beauchemin, L. Gagnon, J. Johrendt, M. Simoneau and T. C. Scialfa, "Proposal: Safety Ambulance Driver Monitoring Unit (SAMU)," 2012.
- [3] P. C. Cacciabure, *Modelling Driver Behaviour in Automotive Environments: Critical Issues in Driver Interactions with Intelligent Transport Systems*, London: Springer, 2007, p. 441.
- [4] J. A. Michon, "Critical View of Driver Behaviour Models: What do we know, What should we do?," in *Human Behaviour and Traffic Safety*, Warren, 1985.
- [5] D. T. McRuer, R. W. Allen, D. H. Weir and R. H. Klein, "New Results in Driver Steering Control Models," *Journal of Human Factors*, vol. 19, no. 4, pp. 381-397, August 1977.
- [6] L. Aarts and I. van Schagen, "Driving Speed and the Risk of Road Crashes: A Review," *Accident Analysis and Prevention*, vol. 38, no. 2, pp. 215-224, 2006.
- [7] M. R. Othman, Z. Zhong, T. Imamura and T. Myake, "Modelling Driver Operation Behaviour by Linear Prediction Analysis and Auto Associative Neural Network," in *2009 IEEE International Conference on Systems, Man and Cybernetics. SMC 2009*, San Antonio, 2009.
- [8] Z. Constantinescu, C. Marinoiu and M. Vladoiu, "Driving Style Analysis Using Data Mining Techniques," *International Journal of Computers, Communications & Control*, vol. 5, no. 5, pp. 654-663, 2010.
- [9] C. MacAdam, Z. Bareket, P. Fancher and R. Ervin, "Using Neural Networks to Identify Driving Style and Headway Control Behaviour of Drivers," in *Dynamics of Vehicles on Roads and on Tracks. 15th IAVSD Symposium*, Budapest, 1998.
- [10] R. S. Sharp, D. Casanova and P. Symonds, "A Mathematical Model for Driver Steering Control, with Design, Tuning, and Performance Results," *Journal of Vehicle Systems Dynamics*, vol. 33, no. 5, pp. 289-326, May 2000.

- [11] P. Raksincharoensak, W. Khaisongkram, M. Nagai, M. Shimosaka, T. Mori and T. Sato, "Integrated Driver Modelling Considering State Transition Feature for Individual Adaptation of Driver Assistance Systems," *Vehicle System Dynamics*, vol. 48, pp. 55-71, 2010.
- [12] W. K. Härdle and L. Simar, *Applied Multivariate Statistical Analysis*, 3 ed., London: Springer-Verlag Berlin Heidelberg, 2012, pp. 332-345.
- [13] Pennsylvania State University, "STAT 505 - Applied Multivariate Statistical Analysis," 2014. [Online]. Available: <https://onlinecourses.science.psu.edu/stat505/>. [Accessed 04 2014].
- [14] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies. 1. Hierarchical systems," *The Computer Journal*, vol. 9, no. 4, pp. 373-380, 1967.
- [15] MINITAB, *Cluster Observations*.
- [16] S. Samarsinghe, *Neural Networks for Applied Sciences and Engineering*, Boca Raton, Florida: Auerbach Publications, 2006.
- [17] K. Swingler, *Applying Neural Networks: A Practical Guide*, San Francisco, California: Morgan Kaufman Publishers, Inc, 1996.
- [18] J. C. Principe, N. R. Euliano and W. C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations*, New York, New York: John Wiley & Sons, Inc, 2000.
- [19] R. J. Clynych, "Surveying & Geospatial Engineering @ CVEN UNSW," February 2006. [Online]. Available: <http://www.gmat.unsw.edu.au/>. [Accessed 22 May 2014].
- [20] C. Veness, "Calculate Bearing, Distance, and More Between Latitude/Longitude Points," 2014. [Online]. Available: <http://www.movable-type.co.uk/>. [Accessed 2014].
- [21] M. Yan, *Methods of Determining the Number of Clusters in a Data Set*, Blacksburg, Virginia: Virginia Polytechnica Institute and State University, 2005.

- [22] G. W. Milligan and M. C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, vol. 50, no. 2, pp. 159-179, June 1985.
- [23] T. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1-27, 1974.
- [24] R. Tibshirani, G. Walther and T. Hastie, "Estimating the Number of Clusters in a Data Set via the Gap Statistic," *Journal of the Royal Statistical Society: Series B*, vol. 63, no. 2, pp. 411-423, 2001.
- [25] The Mathworks, Inc, "Documentation Center R2014a," Mathworks, 2014. [Online]. Available: <http://www.mathworks.com/help/index.html>. [Accessed June 2014].

APPENDICES

Appendix A: Driving Data Set

Table A.1: Data Set of Driving Parameters

Sample	\bar{V}	σ_V	V_{10}	σ_a	\bar{Acc}	σ_{Acc}	\bar{Brk}	σ_{Brk}	D_{tot}
1	24.86	10.08	0.00	11650.78	8919.58	8915.95	7672.13	7351.59	1.54
2	29.40	11.57	0.00	10887.61	7248.22	6700.07	8274.03	8584.52	1.70
3	16.31	7.98	0.00	9346.71	6441.97	5158.56	7326.86	7567.47	0.86
4	30.74	14.93	0.00	8867.13	6289.94	5160.47	6258.73	7218.71	1.57
5	45.67	9.63	0.00	5145.19	3398.57	3538.33	3379.67	4224.92	6.18
6	31.56	11.10	0.00	9269.12	6247.07	6250.23	7065.10	6967.35	0.98
7	34.98	12.20	0.00	9370.23	6094.25	6223.68	6517.83	7622.07	3.42
8	42.63	10.87	0.00	7339.70	4611.84	4966.08	5733.69	5474.02	0.86
9	35.72	13.90	0.03	10286.98	7144.17	6531.78	7339.45	8058.68	2.47
10	39.89	9.73	0.00	6809.31	3824.68	3513.72	4994.17	6360.86	0.93
11	44.36	10.27	0.00	6541.84	4382.83	4426.27	4535.16	5142.27	7.26
12	46.66	7.50	0.00	5678.98	3739.00	4551.67	3392.10	4297.23	2.01
13	31.92	12.61	0.01	10008.83	6759.39	7057.36	6538.58	7888.92	4.38
14	25.54	12.37	0.00	10427.40	6990.38	6786.93	7527.32	8174.14	3.01
15	29.41	12.04	0.00	9840.10	6718.37	6943.48	7038.60	7128.11	0.81
16	24.17	13.74	0.00	9130.50	6734.15	6121.35	5989.01	6969.13	0.76
17	43.38	9.97	0.01	6378.59	4507.17	4216.87	4236.38	5049.08	4.48
18	44.13	10.87	0.00	8823.98	4430.83	4920.79	5158.07	10431.01	1.87
19	46.51	9.27	0.01	6526.68	4356.01	4622.95	4349.44	5102.91	4.45
20	37.49	16.74	0.19	17363.59	10966.77	9261.85	15503.53	14134.79	2.67
21	30.10	14.59	0.01	14589.36	11408.00	8357.39	11694.23	9521.93	4.62
22	38.61	12.48	0.00	16064.99	10930.98	9167.85	14061.65	10964.41	0.87
23	34.56	17.26	0.15	13061.83	9203.09	8118.95	10022.22	9711.64	16.76
24	38.72	19.33	0.20	13118.76	9001.53	7429.21	-9768.26	10686.01	5.28
25	41.66	16.12	0.23	12882.56	9219.96	7717.97	-9633.67	9767.23	4.97
26	37.55	16.40	0.12	12408.50	9954.39	8342.84	-8528.33	8285.99	6.00
27	38.01	16.51	0.21	15064.24	9513.13	8183.60	-13469.44	12067.61	2.80
28	35.57	16.08	0.07	12578.16	8638.80	8086.46	-9362.88	9598.43	3.62
29	38.65	16.97	0.19	15016.10	11308.05	9512.06	-11672.64	9818.15	8.03

Table A.2: Standardized Values for Final Data Set

Sample	\bar{V}	σ_V	V_{10}	σ_a	\overline{Acc}	σ_{Acc}	\overline{Brk}	σ_{Brk}	D_{tot}
1	-1.47	-0.88	-0.58	0.38	0.73	1.35	0.57	-0.28	-0.78
2	-0.86	-0.38	-0.58	0.15	0.05	0.10	0.65	0.24	-0.70
3	-2.61	-1.58	-0.58	-0.33	-0.28	-0.77	0.53	-0.19	-1.11
4	-0.68	0.74	-0.58	-0.48	-0.35	-0.77	0.39	-0.34	-0.76
5	1.31	-1.03	-0.58	-1.62	-1.53	-1.68	0.01	-1.61	1.46
6	-0.57	-0.54	-0.58	-0.35	-0.36	-0.15	0.49	-0.45	-1.05
7	-0.12	-0.17	-0.52	-0.32	-0.43	-0.17	0.42	-0.17	0.13
8	0.90	-0.61	-0.58	-0.95	-1.03	-0.88	0.32	-1.08	-1.11
9	-0.02	0.40	-0.23	-0.04	0.00	0.00	0.53	0.02	-0.33
10	0.54	-0.99	-0.58	-1.11	-1.35	-1.70	0.22	-0.70	-1.07
11	1.13	-0.81	-0.58	-1.19	-1.13	-1.18	0.16	-1.22	1.97
12	1.44	-1.74	-0.58	-1.46	-1.39	-1.11	0.01	-1.58	-0.55
13	-0.52	-0.03	-0.46	-0.12	-0.15	0.30	0.42	-0.05	0.59
14	-1.38	-0.11	-0.58	0.01	-0.06	0.15	0.55	0.07	-0.07
15	-0.86	-0.22	-0.58	-0.18	-0.17	0.24	0.49	-0.38	-1.13
16	-1.56	0.34	-0.58	-0.39	-0.16	-0.23	0.35	-0.45	-1.15
17	1.00	-0.91	-0.51	-1.24	-1.07	-1.30	0.12	-1.26	0.64
18	1.10	-0.61	-0.58	-0.49	-1.11	-0.90	0.24	1.03	-0.62
19	1.42	-1.15	-0.48	-1.20	-1.14	-1.07	0.14	-1.24	0.62
20	0.22	1.35	1.88	2.15	1.57	1.54	1.59	2.61	-0.24
21	-0.77	0.63	-0.48	1.29	1.75	1.03	1.09	0.64	0.70
22	0.37	-0.08	-0.58	1.75	1.55	1.49	1.40	1.26	-1.10
24	0.38	2.21	2.01	0.84	0.76	0.51	-1.70	1.14	1.02
25	0.77	1.14	2.32	0.76	0.85	0.67	-1.68	0.75	0.87
26	0.22	1.23	0.93	0.62	1.15	1.03	-1.54	0.11	1.37
27	0.29	1.27	2.14	1.44	0.97	0.94	-2.18	1.73	-0.17
28	-0.04	1.12	0.29	0.67	0.61	0.88	-1.65	0.67	0.22
29	0.37	1.42	1.79	1.42	1.71	1.69	-1.95	0.77	2.35

Appendix B: Cluster Analysis Results

Table B.1: Results for Gap Criterion

Cluster Number	$E_n\{\log(W_k)\}$	$\log(W_k)$	Gap Value
1	4.02	3.98	0.039
2	3.77	3.66	0.113
3	3.64	3.41	0.233
4	3.53	3.22	0.309
5	3.43	3.10	0.324
6	3.33	3.01	0.320
7	3.24	2.91	0.330
8	3.15	2.83	0.329
9	3.07	2.73	0.334
10	2.98	2.63	0.346

Table B.2: Assignment of Data to Individual Classes

Sample	Class No.
1	1
2	1
3	1
4	1
5	2
6	1
7	1
8	2
9	1
10	2
11	2
12	2
13	1
14	1
15	1
16	1
17	2
18	2
19	2
20	3
21	3
22	3
24	4
25	4
26	4
27	4
28	4
29	4

Table B.3: Cluster Centroids with respect to Each Variable

Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4
\bar{V}	-0.97	1.10	-0.06	0.33
σ_V	-0.22	-0.98	0.63	1.40
V_{10}	-0.53	-0.55	0.27	1.58
σ_a	-0.15	-1.16	1.73	0.96
\overline{Acc}	-0.11	-1.22	1.62	1.01
σ_{Acc}	0.01	-1.23	1.36	0.95
\overline{Brk}	0.49	0.15	1.36	-1.78
σ_{Brk}	-0.18	-0.96	1.50	0.86
D_{tot}	-0.58	0.17	-0.21	0.94

Appendix C: ANN Results

Table C.1: Error Values Generated by the ANN Network

Sample	E_{Class1}	E_{Class2}	E_{Class3}	E_{Class4}
1	0.074	-0.197	-0.148	-0.303
2	0.040	0.155	-0.061	0.018
3	0.020	0.026	-0.010	0.001
4	0.039	0.215	-0.035	0.001
5	0.224	0.025	-0.066	-0.182
6	0.045	0.015	0.043	-0.058
7	0.060	-0.032	0.003	-0.073
8	-0.044	0.140	0.023	-0.131
9	0.044	0.171	-0.058	-0.034
10	0.017	0.135	-0.024	-0.115
11	0.122	0.038	-0.047	-0.138
12	0.376	0.023	-0.027	-0.158
13	0.030	0.105	-0.049	-0.018
14	0.021	0.219	-0.126	0.065
15	0.043	0.013	0.037	-0.052
16	0.033	0.160	-0.019	0.009
17	0.110	0.058	-0.048	-0.153
18	0.148	0.105	-0.025	-0.142
19	0.204	0.035	-0.045	-0.175
20	-0.023	0.048	0.030	0.064
21	-0.021	0.184	0.039	0.035
22	0.082	0.060	0.087	-0.437
24	0.057	0.154	-0.266	0.052
25	0.002	0.097	-0.005	0.051
26	-0.033	0.088	-0.085	0.039
27	0.118	0.124	-0.221	0.049
28	0.007	0.147	-0.033	0.040
29	-0.018	0.060	-0.028	0.061
Max	0.376	0.219	0.087	0.065
Min	-0.044	-0.197	-0.266	-0.437

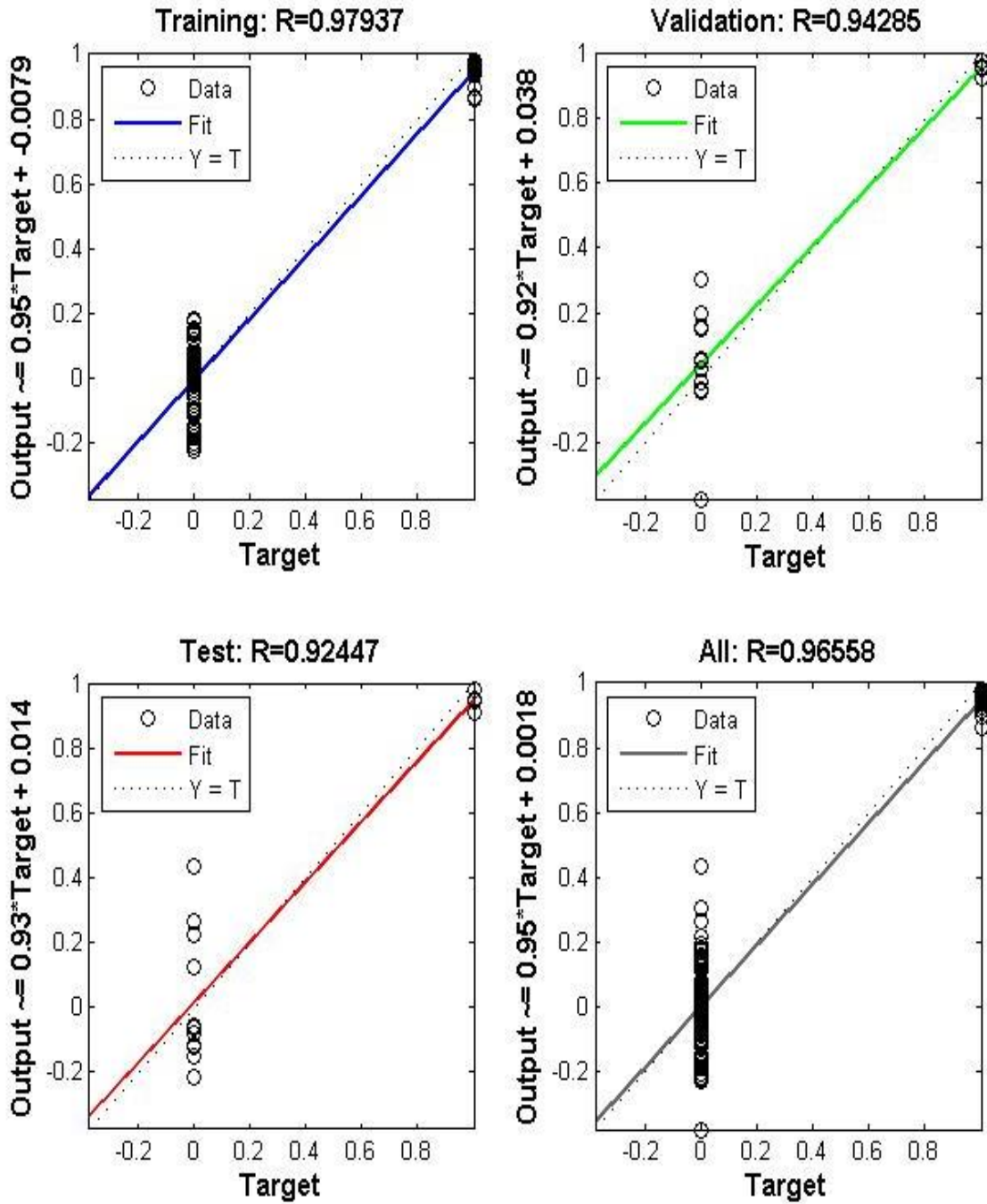


Figure C.1: Regression Plots for Developed ANN Model

Table C.2: Input Layer Weights for Driver Performance Classification ANN

Weights	\bar{V}	σ_V	V_{10}	σ_a	\overline{Acc}	σ_{Acc}	\overline{Brk}	σ_{Brk}	D_{tot}
w₁₁	-0.658	-0.653	0.559	-0.122	-0.475	0.567	0.588	0.323	1.22
w₁₂	0.337	-0.647	-0.237	0.322	0.938	0.473	0.843	-0.406	-1.232
w₁₃	1.679	0.667	0.636	-0.640	-0.519	-0.065	-1.613	0.598	0.285
w₁₄	0.286	0.892	0.481	0.891	1.192	1.301	-0.325	0.582	-0.577
w₁₅	1.412	-0.525	-0.054	-0.924	-0.793	-0.328	-0.041	0.159	0.932
w₁₆	0.374	0.308	-0.172	0.246	0.763	0.470	-1.195	0.344	-0.467
w₁₇	-0.495	-1.147	0.341	0.405	-0.449	0.578	-0.244	-0.927	0.239
w₁₈	-0.097	-0.584	-0.172	0.755	-0.612	-0.463	-0.846	-0.807	0.543
w₁₉	-0.570	0.217	-0.536	1.087	0.030	0.080	0.619	0.778	0.943
w₁₁₀	0.326	0.637	-0.954	0.397	0.495	-0.576	0.061	-0.442	-0.881
w₁₁₁	-0.843	0.554	0.310	-0.675	-0.002	-0.907	-0.059	0.791	0.191
w₁₁₂	-0.624	-0.687	1.156	-0.689	-0.205	0.193	-0.750	0.142	-0.323

Table C.3: Hidden Layer Weights for Driver Performance Classification ANN

	h₁	h₂	h₃	h₄	h₅	h₆	h₇	h₈	h₉	h₁₀	h₁₁	h₁₂
w₂₁	0.59	-0.46	-0.29	-0.63	-1.31	-0.48	0.32	-0.55	-0.65	-0.56	-0.31	-0.55
w₂₂	-0.72	0.05	0.59	-0.27	0.91	-0.10	0.58	-0.61	0.30	0.30	-0.53	-0.95
w₂₃	0.01	-0.03	0.02	0.82	0.05	-0.84	0.02	-0.56	0.63	0.62	-0.01	-0.19
w₂₄	0.01	0.01	0.09	0.36	-0.03	0.71	-0.03	0.68	-0.53	0.40	-0.07	-0.40

Appendix D: Factor Analysis Results

Table D.1: Factor Values Corresponding to Each Test Drive

Factor 1	Factor 2	Factor 3	Class
0.460313	3.077591	-0.5652	1
-0.71631	2.081729	-0.07449	1
-3.53596	3.300682	-1.19458	1
-1.88658	0.994194	-0.37078	1
-7.03761	-3.31599	0.054508	2
-2.60824	1.576643	-0.15614	1
-1.71677	0.261618	0.000238	1
-5.20243	-0.28907	0.438253	2
-0.1821	0.659736	0.202733	1
-6.30476	-0.29547	0.304688	2
-5.16679	-3.06185	0.080557	2
-7.44403	-1.55208	0.4549	2
-0.39541	0.453902	-0.26219	1
-0.52694	1.80449	-0.58462	1
-1.55354	2.019935	-0.30776	1
-1.58092	2.190253	-0.8607	1
-5.69323	-2.03963	0.144381	2
-2.86801	-0.27648	1.111984	2
-5.72824	-2.24959	0.430496	2
9.088206	2.08742	1.700602	3
4.317548	2.118412	0.129444	3
3.99149	3.31025	1.430705	3
7.602487	-2.64044	-0.27568	4
6.618668	-2.81746	-0.04986	4
5.615049	-2.11879	-0.63278	4
8.416693	-1.4449	-0.07	4
4.652582	-0.87085	-0.52869	4
9.38484	-2.96427	-0.55002	4

VITA AUCTORIS

NAME: Ishika Zonina Towfic

PLACE OF BIRTH: Dhaka, Bangladesh

YEAR OF BIRTH: 1989

EDUCATION: B.A.Sc. Mechanical Engineering with
Automotive Option (Co-op) with Distinction,
University of Windsor, Windsor, ON, 2012

M.A.Sc. Mechanical Engineering, University of
Windsor, Windsor, ON, 2014