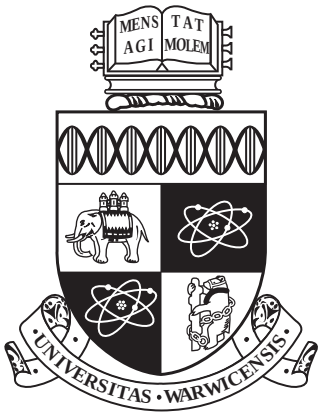


Data Mining of Vehicle Telemetry Data

by

Phillip Taylor

A thesis submitted to The University of Warwick

in partial fulfilment of the requirements

for admission to the degree of

Doctor of Philosophy

Department of Computer Science

The University of Warwick

September 2015

Abstract

Driving a safety critical task that requires a high level of attention and workload from the driver. Despite this, people often perform secondary tasks such as eating or using a mobile phone, which increase workload levels and divert cognitive and physical attention from the primary task of driving. As well as these distractions, the driver may also be overloaded for other reasons, such as dealing with an incident on the road or holding conversations in the car. One solution to this distraction problem is to limit the functionality of in-car devices while the driver is overloaded. This can take the form of withholding an incoming phone call or delaying the display of a non-urgent piece of information about the vehicle.

In order to design and build these adaptions in the car, we must first have an understanding of the driver's current level of workload. Traditionally, driver workload has been monitored using physiological sensors or camera systems in the vehicle. However, physiological systems are often intrusive and camera systems can be expensive and are unreliable in poor light conditions. It is important, therefore, to use methods that are non-intrusive, inexpensive and robust, such as sensors already installed on the car and accessible via the Controller Area Network (CAN)-bus.

This thesis presents a data mining methodology for this problem, as well as for others in domains with similar types of data, such as human activity monitoring. It focuses on the variable selection stage of the data mining process, where inputs are chosen for models to learn from and make inferences. Selecting inputs from vehicle telemetry data is challenging because there are many irrelevant variables with a high level of redundancy. Furthermore, data in this domain often contains biases because only relatively small amounts can be collected and processed, leading to some variables appearing more relevant

to the classification task than they are really.

Over the course of this thesis, a detailed variable selection framework that addresses these issues for telemetry data is developed. A novel blocked permutation method is developed and applied to mitigate biases when selecting variables from potentially biased temporal data. This approach is infeasible computationally when variable redundancies are also considered, and so a novel permutation redundancy measure with similar properties is proposed. Finally, a known redundancy structure between features in telemetry data is used to enhance the feature selection process in two ways. First the benefits of performing raw signal selection, feature extraction, and feature selection in different orders are investigated. Second, a two-stage variable selection framework is proposed and the two permutation based methods are combined. Throughout the thesis, it is shown through classification evaluations and inspection of the features that these permutation based selection methods are appropriate for use in selecting features from CAN-bus data.

Acknowledgements

I would like to first thank my supervisors, Nathan Griffiths and Abhir Bhalerao, for guiding me through my PhD and offering their enthusiasm and inspiration throughout. I also extend these thanks to Sarabjot Anand, who supervised me in my first year but left academia to seek new pastures. I am truly grateful for their support and this thesis would not have been possible without them. I am also indebted to members of Jaguar Land Rover, including Zhou Xu, Thomas Popham, Adam Gelencser, for their advice and in aiding with data collection.

Over the past eight years, the Department of Computer Science and its members have helped develop me personally, academically and professionally. I would like to extend special thanks to the postgraduate members of the Software and Systems group, who offered their enthusiasm and entertainment during lunch breaks and otherwise. I hope to continue these relationships during the post-doctoral position I have recently taken at the department, which is a fortuitous opportunity I will relish.

I would not have undertaken higher education if it were not for my parents, who have always been there for me. They have always encouraged me and told me to aim high, but will be happy that my journey through education is coming to an end. I extend my gratitude to my whole family, for the laughter we share during our Sunday dinners and for reminding me of the important things in life. I would also like to acknowledge my lifelong friends Raymond Ahmad and Stuart MacDonald, with whom I share many memories.

Finally, I reserve my kindest thanks for Katrien Steenmans, who has supported me throughout and proofread much of my work. It is unlikely that this thesis would be finished without her, and I will reciprocate when she is writing hers. I am grateful to all of the Steenmans family, who have welcomed me and whom I have enjoyed several holidays with over the last eight years.

Declarations

This research presented in this thesis was funded by the Engineering and Physical Sciences Research Council (EPSRC) and Jaguar Land Rover Cars (JLR).

Parts of this thesis have been previously published by the author in the following:

[144] Phillip Taylor, Fatima Adamu-Fika, Sarabjot Anand, Alain Dunoyer, Nathan Griffiths, and Thomas Popham. Road type classification through data mining. In *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 233–240. ACM, October 2012

In this paper some initial investigations into data mining from vehicle telemetry data were performed using the Road Classification Dataset (RCD). Different types of feature, feature selection method, and classification model were compared. The results of this paper influences the methodology used throughout this thesis.

[145] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Alain Dunoyer, Thomas Popham, and Zhou Xu. Feature selection in highly redundant signal data: A case study in vehicle telemetry data and driver monitoring. In *International Workshop on Autonomous Intelligent Systems: Multi-Agents and Data Mining*, pages 25–36, June 2013

This paper presents results on selecting features on a per-signal basis in a two-stage feature selection framework. The methods used in Sections 6.3 and 6.3.1 are based on this paper.

[146] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Derrick Watson, Zhou Xu, and Thomas Popham. Warwick-JLR Driver Monitoring Dataset (DMD): A public dataset for driver monitoring research. In *Cognitive Load and In-Vehicle Human-Machine Interaction*, pages 1–4, October 2013

In this paper the Warwick-JLR Driver Monitoring Dataset (WarwickDMD) was announced. It is further described in Section 3.3.

[147] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Thomas Popham, Zhou Xu, and Alain Dunoyer. Redundant feature selection for telemetry data. In Longbing Cao, Yifeng Zeng, Andreas Symeonidis, Vladimir Gorodetsky, Jörg Müller, and Philip Yu, editors, *Agents and Data Mining Interaction*, volume 8316 of *Lecture Notes in Computer Science*, pages 53–65. Springer Berlin Heidelberg, May 2014

This is an extension of [145], and is again the basis of the work in Sections 6.3 and 6.3.1.

[148] Phillip Taylor, Nathan Griffiths, and Abhir Bhalerao. Redundant feature selection using permutation methods. In *Automatic Machine Learning Workshop*, pages 1–8, July 2015

This paper introduced a method for computing efficiently the redundancies of features while using the permutation method. The work is presented as Chapter 5.

[149] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Derrick Watson, Zhou Xu, Adam Gelenscer, and Thomas Popham. Developing a public driver monitoring dataset. In *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, September 2015

After completion of the WarwickDMD, its release was announced in this paper along with some initial analysis of it. This analysis is also presented in Section 3.3.

In addition, the following works are under review:

- Road Classification Without Location Data: A Data Mining Approach. Submitted to Applied Artificial Intelligence.

  This paper is an extension of [144] and investigates the benefits of signal selection, feature extraction, and feature selection. The data mining

methodology used in this paper forms the basis of that used in this thesis, and the results of the experiments are presented in Section 6.2 of this thesis.

- Feature Selection from Temporally Dependent Data Using Block Permutation Methods. In preparation for submission.

  This paper introduces permutation methods for temporally dependent data such as vehicle telemetry, and the work is presented in this thesis as Chapter 4.

Abbreviations

ANOVA	Analysis Of Variance
AUC	Area Under the Receiver Operator Characteristic Curve
CAN	Controller Area Network
CoventryDMD	Coventry-JLR Driver Monitoring Dataset
DALI	Driver Activity Load Index
DFT	Discrete Fourier Transform
ECG	Electrocardiogram
ECOC	Error Correction Output Coding
EDA	Electrodermal Activity
EDR	Electrodermal Response
EEG	Electroencephalography
GPS	Global Positioning Satellites
HR	Heart Rate
HRV	Heart Rate Variability
HSIC	Hilbert-Schmidt Independence Criterion
IID	Independent and Identically Distributed
MCP	Multiple Comparison Procedure
MDL	Minimum Description Length
MI	Mutual Information

mRMR	minimal Redundancy Maximal Relevance
OARD	OPPORTUNITY Activity Recognition Dataset
OSS	One Sided Sampling
PCA	Principal Components Analysis
PCC	Pearson's Correlation Coefficient
PC	Principal Component
RBF	Radial Basis Function
RCD	Road Classification Dataset
ROC	Receiver Operator Characteristic
SCL	Skin Conductance Level
SDSD	Standard Deviation of Successive Differences
SMOTE	Synthetic Minority Oversampling TEchnique
SR	Success Rate
STD	Standard Deviation
SU	Symmetrical Uncertainty
SVM	Support Vector Machine
SWA	Steering Wheel Angle
TLX	NASA-Task Load Index
WarwickDMD	Warwick-JLR Driver Monitoring Dataset

Contents

Abstract i

Acknowledgements iii

Declarations iv

Abbreviations vii

List of Figures xv

List of Tables xvii

1 Introduction 1

1.1 Driver inattention monitoring 2

1.2 Vehicle telemetry data 2

1.3 Data mining 3

1.4 Problem statement and contributions 3

1.5 Structure of thesis 6

2 Background 9

2.1 Data mining process: The learning approach 10

2.1.1 Problem definition 10

2.1.2 Data collection 11

2.1.3 Data cleaning and exploration 13

2.1.4 Temporal feature extraction 14

2.1.5 Sampling 16

2.1.6 Discretisation 17

2.1.7 Feature selection 18

2.1.8 Algorithm engineering 20

2.2 The data mining approach: Evaluation and refinement 21
2.2.1 Structure of evaluation 21
2.2.2 Performance measures 25
2.2.3 Refinement 26
2.3 Automotive applications 26
2.3.1 Driving conditions monitoring 26
2.3.2 Driver monitoring 29
2.4 Summary 33

3 Datasets **34**
3.1 Road classification dataset 34
3.2 Coventry-JLR driver monitoring dataset 36
3.3 Warwick-JLR driver monitoring dataset 38
3.3.1 Collection protocol 39
3.3.2 Data collection 41
3.3.3 Preliminary analysis 43
3.3.4 Ground truth for classification 50
3.3.5 Data release 50
3.4 Non-vehicular datasets 52
3.5 Feature extraction 52
3.6 Summary 53

4 Temporal permutation feature relevancy **55**
4.1 Introduction 56
4.2 The permutation method 58
4.3 Permutation methods for dependent data 60
4.4 Feature ranking methods 67
4.5 Experimental setup 71
4.6 Results 73
4.6.1 Blocked-permutation test 73
4.6.2 Feature rankings 80

4.6.3 Classification 86
4.7 Conclusions 94

5 Redundant permutation feature selection 96
5.1 Introduction 97
5.2 A redundant permutation feature selector 99
5.2.1 Permutation redundancy 99
5.2.2 Redundant permutation feature selection 106
5.3 Evaluation 106
5.4 Conclusion 115

6 Feature selection from vehicle telemetry data 117
6.1 Introduction 118
6.2 Benefits of signal selection 120
6.2.1 Classification and evaluation 122
6.2.2 Results 122
6.2.3 Discussion 125
6.3 Two stage feature selection 126
6.3.1 Evaluation design 127
6.3.2 Results 128
6.4 DMD analysis 140
6.5 Conclusions 146

7 Discussion and conclusions 150
7.1 Contributions 152
7.2 Directions for future research 156
7.3 Final remarks 158

References 160

List of Figures

2.1 Diagram showing the data mining process 10
2.2 A temporal evaluation structure 23

3.1 Map of the Gaydon emissions track 37
3.2 Screen shot of the video recorded during the trials 37
3.3 Fifteen seconds of an EDA signal 43
3.4 Five seconds of an ECG signal 44
3.5 Mean error rates of participants for the secondary tasks 44
3.6 Mean responses to NASA TLX questions 45
3.7 Mean values of HR, HRV, SCL and EDR frequency over all subjects for the different periods of the trial 49

4.1 Dependency of three small consecutive blocks over time 64
4.2 Dependency of three large consecutive blocks over time 65
4.3 p-value against block size for a signal in each of the RCD, the CoventryDMD, and the WarwickDMD 74
4.4 MD_{MI} score against block size for a signal in the RCD with two further sub-samplings by factors of 10 and 100 77
4.5 MD_{MI} score against block size for a signal in the CoventryDMD with two further sub-samplings by factors of 10 and 100 78
4.6 MD_{MI} score against block size a signal in the WarwickDMD with two further sub-samplings by factors of 10 and 100 79
4.7 The relationship between the ranking strategies and ranking by MI for the RCD 81
4.8 The relationship between the ranking strategies and ranking by MI for the CoventryDMD 82

4.9 The relationship between the ranking strategies and ranking by MI for the WarwickDMD 83
4.10 MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the RCD 87
4.11 MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the CoventryDMD 88
4.12 MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the WarwickDMD 89
4.13 Classification AUC scores for the RCD with the MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} selection methods 90
4.14 Classification AUC scores for the CoventryDMD with the MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} selection methods 91
4.15 Histogram of the number of times ranks were assigned to the bias features by the MI, SU, HSIC, Z_{MI}, MD_{MI}, and MR_{MI} ranking methods for the RCD, CoventryDMD and WarwickDMD 93

5.1 Scatter plots of MI against PMD_{MI} 101
5.2 Permutation distributions of features with increasing dimensionalities computed from a common target 102
5.3 Scatter plots of MI and Z_{MI} against PMD_{MI} 104
5.4 Scatter plots of MI and Z_{MI} against PC_{MI} 105
5.5 Mean AUC scores achieved over ten evaluations when selecting between one and twenty features from the Musk 1 and TR 11 datasets 109
5.6 Computation times for the $MImRMR$, $SUmRMR$, $Z_{MI}mRMR$ and PC_{MI} methods to rank all features in simulated datasets . . 114

6.1 Processing methods for data, using PCA, MI and Feature Extraction in different orders 121
6.2 Carriageway classification AUC when selecting features using the different selection paths 123

6.3 Road-type classification AUC when selecting features using the different selection paths 124
6.4 AUC performances when between one and five features were selected in the first stage from the CoventryDMD, using Z_{MI}, $PmRMR$, Symmetrical Uncertainty (SU) and $SUmRMR$ 129
6.5 AUC performances when between one and five features were selected in the first stage from the RCD with carriageway labelling, using Z_{MI}, $PmRMR$, SU and $SUmRMR$ 130
6.6 AUC performances when between one and five features were selected in the first stage from the RCD with road labelling, using Z_{MI}, $PmRMR$, SU and $SUmRMR$ 131
6.7 AUC performances when between one and five features were selected in the first stage from the OARD, using Z_{MI}, $PmRMR$, SU and $SUmRMR$ 132
6.8 AUC performances for the CoventryDMD and RCD with carriageway labelling when features were selected with one feature per signal in the first stage 133
6.9 AUC performances of the Random Forest classifier for the RCD with road labelling and the OARD when features were selected with one feature per signal in the first stage 134
6.10 Redundancies measured in PC_{MI} for different numbers of features selected from the CoventryDMD using $SUmRMR$ and $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage 136
6.11 Redundancies measured in PC_{MI} for different numbers of features selected from the RCD with carriageway labelling using $SUmRMR$ and $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage 137

6.12 Redundancies measured in PC_{MI} for different numbers of features selected from the RCD with road labelling using $SUmRMR$ and $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage 138
6.13 Redundancies measured in PC_{MI} for different numbers of features selected from the OARD using $SUmRMR$ and $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage 139

List of Tables

3.1 List of signals recorded in the RCD. 35
3.2 Label counts for the carriageway and road ground truths 36
3.3 Secondary tasks drivers performed in the CoventryDMD 38
3.4 The protocol for the WarwickDMD experiment 40
3.5 Example of the N-back test with a block of 10 numbers 40
3.6 p-values from two way t-test and ANOVA for the physiological and selected signals of the vehicle telemetry data streams 46
3.7 Details of datasets from the UCI and Tuned IT repositories . . . 51
3.8 List of statistical and structural features extracted from each signal from the RCD, CoventryDMD, WarwickDMD and OARD . . 53

4.1 List of signals taken from the RCD, the CoventryDMD, and the WarwickDMD 71
4.2 Rank positions by MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} for illustrative signals of the RCD, the CoventryDMD and the WarwickDMD 84

5.1 Number of times features selected by $MImRMR$ outperformed those selected by $PmRMR$, and vice versa, for each classifier over all train-test iterations 110
5.2 Number of times features selected by $SUmRMR$ outperformed those selected by $PmRMR$, and vice versa, for each classifier over all train-test iterations 110
5.3 Mean AUC performances for each dataset (with the maximum number of features with $\{1, 2, 3, 4, 5\}$ splits added) and classifier for the $MImRMR$, $SUmRMR$, and $PmRMR$ selection methods when selecting 5 features 111

5.4 Total number of the original features from each dataset ranked in the top five by $MImRMR$, $SUmRMR$, and $PmRMR$ for different split types and a deform type of $\{5, 10, 20, 30, 40\}\%$ . . . 112

6.1 Redundancy for the eight selection algorithms measured by SU of top ten features selected from the CoventryDMD, RCD-carriageway, RCD-road, and OARD datasets 141

6.2 WarwickDMD features ranked by $PmRMR$ in a two-stage process with one feature selected per signal in the first stage 142

6.3 Mean AUC performances when building models for different combinations of drivers and testing on individual driver data for the distraction status (*normal* or *distracted*) and a increase in heart rate (*baseline* or *increase by 5 bpm*) 145

6.4 AUC performances for each train-test iteration with data from individual drivers to predict the distraction status (*normal* or *distracted*) and an increase in heart rate (*baseline* or *increase by 5bpm*) 147

CHAPTER 1

Introduction

Driving is a safety critical task that requires a high level of attention and workload from the driver [123, 124, 125, 168]. Despite this, people often perform secondary tasks such as eating or using a mobile phone, which increase workload levels and divert cognitive and physical attention from the primary task of driving [140]. As well as these distractions, the driver may also be overloaded for other reasons, such as with an incident on the road or holding conversations in the car.

This problem of driver distraction has several potential solutions. The first, is to remove the driver from the system with autonomous or self driving vehicles. The technology for self driving cars is still being developed, however, and there are many social, legal, and ethical issues to overcome before they are adopted widely [98, 133]. Another approach is to reduce the number of tasks a driver performs, or by simplifying the driving task [18, 150]. For example, adaptive cruise control and automatic breaking systems are designed to reduce the complexity of driving [96]. Such driver assistance systems introduce new issues that also cause inattention, however, either because the driver is under-stimulated and their attention lapses or because they trust the vehicle to perform tasks that it is incapable of [19].

Another solution to this distraction problem is monitor the workload levels of the driver [73, 161, 167], and limit the functionality of in-car devices while the driver is overloaded [114]. This can take the form of withholding an incoming phone call or delaying the display of non-urgent vehicle information. It may also be possible to warn the driver when they are inattentive, or have the vehicle intervene only when they are inattentive and urgent action is required [74].

1.1 Driver inattention monitoring

Driver monitoring can be performed in various ways, from monitoring the vehicle's external environment to directly measuring the driver's physiology. The external environment provides insight into the driver's workload through the characteristics of the road or type of terrain [131, 132]. The driver's physiology, can be used to directly assess the current workload of the driver [57, 99, 127], but physiological sensors are more intrusive. Video processing methods can also be applied to analyse the posture, head position, and gaze of the driver, but cameras systems can be expensive and unreliable in poor light conditions or when the driver wears glasses. A third approach, and one that is taken in this thesis, is to use the driving behaviour and vehicle telemetry to assess workload of a driver.

1.2 Vehicle telemetry data

Telemetry data typically consists of measurements over time, often at high sample rates. This thesis is concerned mainly with the analysis of vehicle telemetry data, although its contributions are relevant to temporal data found in other domains, including medicine, environmental monitoring and activity monitoring. In general the measurements are made by sensors in a system or environment [3]. Electrocardiogram (ECG) sensors in the medical domain, for example, record measurements of current across a patient's chest, and for earthquake detection a seismometer measures movements in the ground.

Sensors in vehicles communicate via a Controller Area Network (CAN)-bus [29, 30, 69], which is a broadcast protocol on which all nodes receive any message sent over the network. Node identifiers are sent within messages and nodes are able to ignore messages that are irrelevant to them to avoid becoming overloaded. In a modern vehicle there are over 50 sensors providing over 1000 telemetric signals, including those that measure wheel speeds, Steering Wheel Angle (SWA), suspension heights, temperatures, fuel tank status and many

other aspects. All these sensors communicate over the CAN-bus, and because a broadcast protocol is used it is possible for a data logger to connect to and record all messages they send.

1.3 Data mining

Data mining is the process whereby data is turned into patterns to describe a part of its structure [3, 61, 71, 84, 160]. Data is in abundance and is generated in almost all fields, from science to business and from media to transport. This is largely due to the ease and inexpensiveness of storing data, allowing decisions to be deferred to subsequent processing that may not yet be developed. There are several difficulties in processing such data, including the computational resources required and avoiding useless, uninteresting or spurious findings.

In the automotive industry data is produced in and analysed by a range of business divisions, including research and development, manufacturing, and after-market care [78]. In this thesis, the data mining of vehicle telemetry data is considered. Vehicle telemetry data is recorded in higher detail and for more tasks than ever before, including fault detection and diagnosis (e.g. [20, 43, 72, 175]), road type, surface and pot-hole detection (e.g. [25, 60, 83, 103, 144]), and driver monitoring for both safety and comfort concerns (e.g. [114, 130, 151, 152, 161]). We develop techniques that aim to minimise spurious findings from vehicle telemetry data in these tasks, while still ensuring that those findings are interesting and useful. Furthermore, to allow the data mining process to be performed within a reasonable time, we address the issue of computational expense in the proposed techniques.

1.4 Problem statement and contributions

This thesis aims to apply data mining techniques to vehicle telemetry data in order to assess the current workload of the driver. Specifically, the prob-

lem statement is: **Can a data mining methodology be developed that enables non-intrusive estimation of the current workload levels of a driver using telemetry data?** To have estimation of the current workload levels, models must take inputs from vehicle telemetry from only a short period prior to the current time where the estimation is being performed. The models must also be reliable enough to allow the vehicle to confidently adapt to the workload level of the driver, by either reducing its functionality or intervening in dangerous situations. In developing a data mining methodology for processing telemetry to produce such models for estimating driver workload, this thesis makes the following main contributions:

1. **Developing an unbiased relevancy measure for temporal variables based on the permutation method.**

   Selecting features from large feature sets is an example of the Multiple Comparison Procedure (MCP), which is responsible for input selection errors, over-fitting, and over-searching [65]. Permutation methods have been proposed as a solution to the MCP and are used to assign significance to a test statistic, with respect to the null hypothesis of the observation being insignificant. Permutation methods cannot be directly applied to temporal data, however, and there are several approaches to using them in ranking features. To apply permutation methods to temporal data, therefore, we use a blocking strategy in a similar manner to Kirch [75] and Adolf et al. [2], and introduce a new strategy of using random block sizes. Furthermore, to rank features we propose two non-parametric methods of normalising a correlation statistic that can be used in a feature ranking.

2. **Establishing a method for feasibly computing unbiased feature redundancies using the permutation method.**

   The naive approach for computing redundancies using the permutation method is extremely expensive computationally. Each individual permutation method for two variables consists of P correlation calculations be-

tween them. Typically, P is in the order of thousands, which means Mutual Information (MI) is computed thousands of times for each permutation method between two variables. For relevancy computations of m features, this number of correlation computations is multiplied to Pm. When redundancies between variables are considered, the computational complexity increases to $O(Pm+Pm^2)$, which is infeasible for many feature sets. We therefore propose a method for estimating the unbiased correlations by comparing permutation correlations computed during relevancy comparisons. This method is then used in feature selection frameworks such as minimal Redundancy Maximal Relevance (mRMR) [113].

3. **Using known redundancy structure in features extracted from signal data with signal selection, feature extraction, and feature selection.**

   The selection of model inputs is an important stage of the data mining process, as too many or bad inputs can cause issues in both model performance and complexity. As well as selecting from features extracted over sliding windows of telemetric signals, selection can also take place directly on the raw telemetry. Selecting from raw signals, although more computationally efficient, may harm performance of models built on the data. Extracted features can also be selected on a per-signal basis, considering the redundancies between features extracted from the same signal first. The selected features can then be combined for a second stage of selection to produce the final feature set. This possibly allows for redundancy between the features to be better removed, and should improve performance of models that use them.

4. **Advancing the process of feature selection from vehicle telemetry data for classification problems such as driver workload estimation.**

   We will provide a methodology for using telemetry data to build predictive

models for classification tasks, such as classifying the current road-type and estimating workload levels of the driver. In particular this methodology will focus on selecting features to use as inputs to such predictive models that determine parameters of the driving environment and driver. The models will then be evaluated to estimate their performance and determine the efficacy of the proposed methodology.

A final contribution, supplementary to developing a data mining methodology, is the production of publicly available datasets for driver monitoring research. The Road Classification Dataset (RCD) is collected over several journeys and is presented as an environment classification problem. The ground truth is the current road type the vehicle is on, and the workload level of the driver can be estimated from this [128, 132, 136, 172]. Town roads, for instance, entail different levels of workload than highways [131]. To estimate the current workload level more directly, the ground truth of the Warwick-JLR Driver Monitoring Dataset (WarwickDMD) is taken from physiological sensors. Physiological measures, including Heart Rate (HR) and Electrodermal Response (EDR), recorded from such sensors have been shown to change with respect to workload in past research (e.g. [13, 99, 126, 136, 167]). This means that they can be used to estimate the current workload of the driver and be converted to labels in training data for predictive models to estimate workload. A final ground truth is taken from the activity of the driver at a particular moment in time, which includes normal driving and driving with secondary tasks.

1.5 Structure of thesis

The remainder of this thesis is structured as follows:

Chapter 2 provides the necessary background for this thesis. The proposed data mining methodology, based on the general data mining process [3, 70, 160], is presented. The general data mining process describes how data should be collected, processed, and learned from to produce models capable of making pre-

dictions. With some specialisations, the methodology described can be applied in building predictive models from vehicle telemetry data in the automotive domain. Two applications of data mining of vehicle telemetry are then summarised, namely environment and driver monitoring.

Chapter 3 describes the datasets that are used throughout this thesis. Three vehicle telemetry datasets, namely RCD, Coventry-JLR Driver Monitoring Dataset (CoventryDMD) and WarwickDMD are described in detail. The RCD and WarwickDMD datasets have been made publicly available to aid research in data mining, and driver and environment monitoring. Further, statistical analysis of WarwickDMD is presented to confirm findings made by Mehler et al. [99], Reimer et al. [127]. Finally, other datasets collected in non-automotive domains and that are available via UCI[1] and Tuned IT[2] repositories are described briefly. The majority of these datasets are non-temporal and are used in Chapter 5, where temporal issues regarding the permutation method are not considered for simplicity. The vehicle telemetry datasets and the OPPORTUNITY Activity Recognition Dataset (OARD) are used for experiments in Chapter 6.

Chapter 4 investigates the application of permutation methods for feature selection in temporal domains. Permutation methods can be used to normalise for several biases found in the feature selection process, namely input selection errors, over-fitting, and over-searching. They cannot typically be applied to temporal data, however, as one assumption they require is that samples in data are exchangeable, which is not the case in data with high autocorrelation. This chapter aims to overcome this limitation through treating the data in blocks, and it is successfully used in normalising for biases in feature selection using MI. Five potential ranking statistics are suggested and applied in selecting features from the three vehicle telemetry datasets.

Chapter 5 approaches the issue of redundancy analysis with mitigated biases using the permutation method. The permutation method itself is expensive

[1] `http://archive.ics.uci.edu/ml`
[2] `http://tunedit.org/repo/Data/`

computationally and consists of thousands of individual permutations and MI correlation calculations. The permutation method must be performed m^2 times to compute redundancies between m features, which is infeasible in general. To avoid this computational requirement, a new redundancy measure is proposed based on the comparison of permutation distributions generated from a common target variable. This approach requires only m permutation methods and efficiently provides all m^2 redundancies between features. Using simulated data with features of various noise and bias levels, we show that the approach provides good estimates of permutation normalised MI. We then apply it in the mRMR framework to successfully select features from the non-temporal datasets listed in Table 3.7, having added extra features to increase the bias and redundancy of their features.

Chapter 6 combines the temporal permutation method introduced in Chapter 4 with the redundancy computation approach proposed in Chapter 5 to produce a method for selecting features from temporal data using permutation normalised correlation estimates and considering redundancy. Selecting signals prior to feature extraction is also considered as this may provide further efficiency gains in the feature selection process. Finally, a two-stage feature selection process is proposed to take advantage of known redundancy structures in features extracted from signal data. Here, features are selected first from individual signals before being combined with those from different signals for a second selection stage. This two-stage feature selection process is then applied to selecting features from all of the temporal datasets described in Chapter 3.

Chapter 7 concludes this thesis by summarising the research contributions presented, and identifying possible directions for future research.

CHAPTER 2

Background

The terms data mining, data science, pattern recognition, and machine learning are used variably in the literature and are often confused with each other, but all have the common aim of learning patterns and building models from data. This process of modelling data, or using it to make predictions, has become pervasive in the modern world and is used across all scientific disciplines. Temporal, time series or telemetry data mining has been successfully applied in medicine to monitor patients in real time and for weather and environmental prediction to predict the weather in the near future or the climate in years to come [3, 160]. Vehicle telemetry from aeroplanes [93], NASA's space program [163], and automobiles [20, 60, 78, 102, 103], has been mined for various applications, including safety improvement, fault detection, or efficiency gains [3].

Data is gathered at unprecedented rates and is often at least partially unstructured or undefined, which means analysing it is a complex and sometimes difficult task even for domain experts. The advent of connected sensor technologies [3] has amplified this as it allows collection of data streams with relative ease and often at high sample rates. In this chapter the necessary background of this thesis is covered. In Sections 2.1 and 2.2, the data mining process and methodology that is used throughout this thesis is outlined. The data mining methodology is split into the learning approach, which is discussed in Section 2.1, and evaluation and refinement, which is discussed in Section 2.2.1. Applications of data mining methodologies in the automotive industry for driver and environment monitoring are discussed in Sections 2.3.1 and 2.3.2.

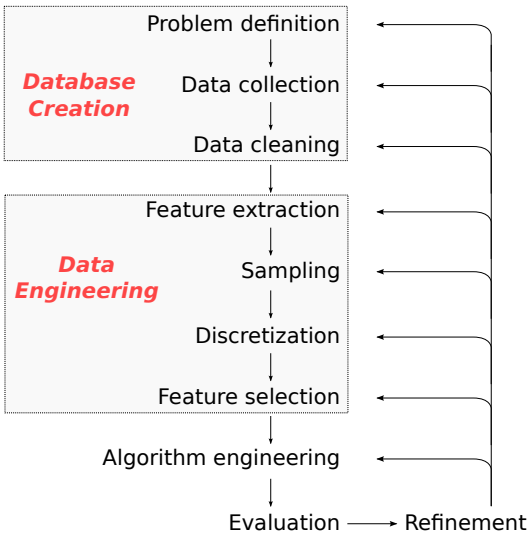


Figure 2.1: Diagram showing the data mining process.

2.1 Data mining process: The learning approach

The methodology used in this thesis is based on the general data mining process described by John [70], and is outlined in Figure 2.1. Each stage in this process is linked and it should be iterated on as new findings are made. For instance, discoveries during data cleaning and exploration will influence decisions in later stages. As well as this, results from model evaluation will affect decisions in previous stages, as the learning approach is iterated upon and refined. In this section the learning approach is discussed, which consists of the Database Creation, Data Engineering, and Algorithm Engineering components. Each stage in these components are described in the following sections.

2.1.1 Problem definition

The problem definition should communicate to the investigator what is required from the data mining process and how to know if it is successful [70]. A problem can often be formed as a question, for example "what kind of road is the vehicle currently travelling on?". In this example there is no definition of the kinds of

roads that exist, and so evaluating any predictions is impossible. An improved problem statement may then be "is the vehicle currently travelling on a single or multiple lane road?". Here, the aim of deciding the road type between a deterministic set of options is clear to the practitioner.

Even if a problem statement can be understood by the practitioner, however, it may not be complete. Analysis of the data may discover something new and unexpected about the domain, which may mean the problem definition requires some refinement. For example, a third kind of road with no lanes may be discovered, leading the domain expert and practitioner to expand the current definitions of the road types or add a third one. If the current definitions are altered, the problem statement may then read as, "is the vehicle currently travelling on a road with multiple lanes or not?".

Finally, any restrictions on the resources or inputs should be clearly defined [70]. A model that requires the vehicle to travel on the same road for a full hour before outputting a decision would not be suitable in the real world where most journeys are shorter than this. A final refinement is therefore needed, where the problem statement becomes, "is the vehicle currently travelling on a road with multiple lanes or not? The model should use inputs from vehicle telemetry recorded in the previous 2.5 seconds only.".

2.1.2 Data collection

The data collection stage is where a database describing the defined problem is created [70]. The variables present in the database should describe the problem appropriately, and the conditions under which they are recorded should be controlled carefully. For example, a problem statement such as "determine the stopping distance of the vehicle with differing load weights and an arbitrary driver on dry and good quality tarmac", would require a model with inputs such as the current travelling speed, accelerations and pedal positions. As well as recording these signals the domain expert may suggest also to record suspension measurements, as these can be used in estimating the vehicle's weight

distribution and allow the model to capture its effects on stopping distance. Data should be collected using several drivers on dry tarmac roads and with different load weight distributions, to create a database that fully describes the problem definition. Collection under conditions other than these will introduce noise into the database and be detrimental to the performance of models built on the data [50]. Of course, this is often unachievable due to limits on resources and difficulties in properly defining the problem.

Collection of vehicle telemetry data is made possible by connecting a data logger to the Controller Area Network (CAN)-bus [29, 30, 69], which is able to record all communications between the vehicles control units, such as the engine, transmission or steering control units. The CAN-bus is a protocol and medium for sensors and actuators in the vehicle to communicate with one another. Devices are easily connected to a CAN, and receive all data sent over it but process only messages relevant to themselves. The engine control unit will receive messages sent by the audio system, for example, but will not process them. When two devices communicate at the same time, the lower priority device is able to recognise this and end its communication without affecting or delaying the higher priority message. Once the higher priority broadcast has ended, the lower priority device will reattempt its communication.

CAN is an asynchronous event based message protocol where devices broadcast messages on events, which can be time based [69]. For example, *IndicatorStatus* may be communicated only when it is relevant and the indicator is being used, and others such as *VehicleSpeed* may be broadcast regularly at 5Hz. These inconsistencies mean that it difficult is to process the data log and build models on it directly, and so it is typical to re-sample the data at a common rate, e.g. between $10 - 100$Hz, to produce signals with samples of the same frequency.

Finally, if the problem is to be posed as one of classification, the ground truth used to derive the labels or targets must be assigned in a consistent and reliable way to produce a target variable [3]. Improper label assignment can

lead to noise in the learning process leading to poorer classification results or invalid conclusions during evaluation. The target variable should be assigned at the same rate as the signals, with a label for each sample in the data [3, 50].

Often, the database will be made up of several smaller datasets, either because it was split into drive cycles or because it was collected over several journeys. It is often advantageous to maintain the separate datasets and have a mechanism for combining them throughout the data mining process. This approach provides greater flexibility than if the datasets were considered as one. This enables individual journeys to be considered in learning, in order to customise models for certain circumstances, and allows training and testing data to be from separate drive cycles.

2.1.3 Data cleaning and exploration

Once a database is created, it should be inspected to ensure that all variables were recorded as expected [70, 110]. This is to say, for example, that a signal named *VehicleSpeed* represents the speed of the vehicle at the current point in time. Traditionally, this has been performed by a human analyst, inspecting each variable for any defects or unexpected characteristics. For instance, value changes from one sample to the next may be expected to be small, or two variables might be known to have a high correlation from analysis of other related data. Observations that are at odds with any expectations may have to be explained, and in some cases rectified, before any conclusions can be drawn from the data. For example, rapid deceleration and high suspension activity is unexpected in data collected during a regular commute. If the commuter reports performing an emergency stop during that journey, however, this offers a reasonable explanation for the spurious data. If this event is then seen to be outside of the problem scope, a judicious practitioner may choose to then remove this period from the database and minimise its effects on the data mining process.

Ideally the database should also be analysed for artefacts such as excessive

noise, bias, or autocorrelation, that may lead to bogus concepts appearing to be meaningful, or hiding other genuine concepts [70, 110]. This is of particular concern in vehicle telemetry data as it inherently has high autocorrelation due to its temporal nature, and data collection efforts invariably lead to datasets containing signals with biases. Signals related to the duration of a journey, such as fuel level, often appear highly related to the target variable, even though they are unrelated to the problem definition. Other signals that are affected by this, such as yaw rates, accelerations and engine oil temperatures, may also exhibit minor biases that are not obvious and are unlikely to be noticed by an analyst.

Performing this manual analysis to find issues with large databases is an extremely expensive task, and many may go unnoticed by the analyst. Typically, only a small number of key signals that are well understood are inspected and if these appear to have no issues, the same is assumed of the remainder. In this thesis, we assume that noise or bias is likely to remain in the database and so specialised techniques are developed to mitigate them later in the data mining process during the feature selection stage (Section 2.1.7).

2.1.4 Temporal feature extraction

In temporal data mining, it is advantageous to include historical information when performing classification [3, 5, 143, 161]. Without this, the current sample only contains information about the exact point that sensor measurements were made. This means that no trend or statistical information contained in signals can be used in determining the classification. We refer to this process of incorporating historical information into the current sample as *temporal feature extraction*, although in some literature it is referred to as motif extraction [3].

In this thesis the same temporal feature extraction process is applied to data from individual journeys or drive cycles. After feature extraction, they should again be maintained as separate datasets. A feature is extracted from a signal, S, by applying a function, $f(\cdot)$, to S over a sliding window of length l. At time t in the signal, the output of the function provides a temporal summary of the

signal for the window,

$$f(s_t, s_{t-1}, \ldots, s_{t-l+1}) = f(S_{t,l}), \tag{2.1}$$

where $S_{t,l}$ is the signal between times t and $t - l + 1$. If $t < l$, because it is at the beginning of the recorded signal, t samples are used in extracting the feature. This is performed for all values of t, ensuring that a signal with n samples produces a feature that contains n samples also to line up with the target variable, Y.

Features can be split into two categories, namely structural and statistical. Structural features describe the trend of the signals, whereas variations, peaks, and averages are represented by statistical features. Several different features of both types are extracted from each signal over different temporal windows to produce a set of features, $\mathbf{X} = \{X_1, X_2, \ldots, X_m\}$. Different window lengths, combined with different features, allow different types of historical information to be extracted from the signal. Features extracted over large window lengths may contain more historical information, but may update slowly to changes in circumstance and be of little value in a real time predictive model. For instance, during an emergency stop taking 2 seconds, the mean vehicle speed over a 5 seconds sliding window would be non-zero for 7 seconds after the start of the incident, even though the circumstances have changed drastically. Conversely, features extracted over shorter window lengths will update more quickly, but may have higher variance and be more susceptible to noise as they are computed from fewer values.

Instead of extracting features, it is also possible to automatically learn features, through genetic programming [44, 45, 100, 162] or by using deep learning techniques such as stacks of restricted Boltzmann machines [48, 120]. Both of these techniques have been successfully applied in learning features, and categorizing images [120], music [48, 100] or performing other classification tasks from temporal sensor data [45, 162]. Both genetic programming and deep learning

can be used to learn functions to transform data into a set of features that are more descriptive of the target variable. Genetic programming produces a function made of basic operations (and potentially of the form shown in Equation 2.1), selected for performance over generations of mutation, crossover and reproduction [44]. In deep learning an abstract representation of the data is produced by using multiple layers of models such as restricted Boltzmann machines, in which outputs of one layer are used as inputs to the next layer. The data is input in the first layer and the outputs of the final layer can be used in predictive models for classification or regression tasks [48].

2.1.5 Sampling

If the data collected has a large class imbalance, models built on it may incorrectly bias to the majority class. The Road Classification Dataset (RCD) described in Section 3.1 has almost six times more samples of single lane roads than roads with multiple lanes. An apparently successful model that predicts the road has one lane regardless of the input and would still maintain an accuracy of over 80%, although this would have little use in reality. One approach to dealing with this problem is to re-sample the data, by either removing majority class samples (under-sampling), or generating copies of the minority class samples (over-sampling) [55].

A simplistic method for under-sampling is to remove samples from the majority class at random to reduce their number and balance the class distribution. This potentially removes information from the data, however, and may cause a model to under-fit. In over-sampling, minority class samples are replicated at random to increase their number, but this may lead to over-fitting. More sophisticated methods include One Sided Sampling (OSS) [79] and Synthetic Minority Oversampling TEchnique (SMOTE) [15]. OSS aims to produce the minimal set of samples to describe the data, removing from the majority class borderline samples that are close to the decision boundary, and redundant samples that can be replaced by others [79]. In SMOTE, synthetic samples are generated

along the hyper-planes between samples of the minority class. Although this avoids over-fitting somewhat, it causes the variance of the data to increase and over-generalises the minority class [15].

Over-sampling and under-sampling techniques are impractical in multi-class situations, as the minority and majority classes are both hard to define and re-sample properly [91]. One approach to mitigating imbalance in mutliclass datasets is to use cost-sensitive learning [55], where different costs are assigned to misclassifying a sample as a particular label. Assigning higher costs to misclassifying a minority class sample than a majority class sample can force the model to adjust to the imbalance. Alternatively, Error Correction Output Coding (ECOC) can be applied, which is an ensemble classifier where the multi-class classification problem is decomposed into several binary classifications, referred to as dichotomies, and a model is built for each [8]. Re-sampling techniques can then be applied in each of the dichotomies to fix both imbalance introduced by the decomposition and any imbalance in the original dataset [91, 135].

Another consideration here is the sample size itself [70]. If there are too many samples, it may become infeasible computationally to perform an in-depth analysis of the data, which can be mitigated through sub-sampling. In non-temporal data, a stratified random sub-sample is used, retaining the original class distributions. For temporal data, taking every t^{th} sample is generally sufficient, and is equivalent to lowering the recording frequency. This will have the effect of also reducing the autocorrelation in the data, which is often beneficial for some algorithms and evaluation strategies. Sub-sampling reduces the amount of data a model learns from and leads to worse performance, which means sub-sampling should not be substantial.

2.1.6 Discretisation

Many feature selection methods and learning algorithms are only able to handle discrete data, and so any continuous features are discretised before further processing [70, 160]. The process of discretisation splits the range of values a

feature can take into blocks of contiguous values. All values of a feature within one of these blocks are then given the same discrete value. Choosing the block ranges however is non-trivial, and there are several methods of doing so, including equal range or equal frequency binning [160]. Equal range binning splits feature values into fixed discrete levels, while equal frequency binning uses discrete levels that ensure balanced distribution in the new feature. Both of these methods are unsupervised, however, and do not consider the predictive ability of the features produced.

In a supervised setting the target can be used to define the discretisation levels, in order to maximise the predictive performance of the discretised features. The widely used Minimum Description Length (MDL) method, for example, recursively splits the domain of the variable into multiple discrete levels while maximizing the information gain at each cut point [31]. Others include the class-attribute contingency coefficient [153], and class-attribute interdependence maximisation [80].

2.1.7 Feature selection

Feature sets, including those of vehicle telemetry signals, often contain numerous irrelevant and redundant features, both of which have a negative effect on the performance and complexity of models built on data [58, 76]. For example, the door lock status is unlikely to be useful in many situations and engine speed is highly redundant to the vehicle speed. Supervised feature selection aims to overcome this by selecting those features that are highly correlated to the class labels, yet uncorrelated to each other. There are three approaches to this in general: embedded methods, wrapper methods, and filter methods [46, 76]. Embedded feature selection processes act as part of the training of a machine learning algorithm, for example in Decision Trees that select features to split on to give the highest classification performance. Wrapper methods use learning algorithms to evaluate feature sets, often performing greedy searches through the feature space and using classification accuracy as a fitness function. Because

embedded methods are often specific to learning algorithms [46] and wrapper methods are computationally expensive [76], filter methods are often preferred.

Filter methods operate separately from the learning process [76]. Although the performance of features selected using a filter method are somewhat dependent on the learning algorithm used [39], they rank features by their expected performance generally [76]. A filter for feature selection can be constructed by ranking features by their individual *relevance* to the target and choosing a number of the highest ranked features. Choosing features from such a ranking does not consider *redundancy* between features, however, and is therefore likely to select several highly relevant features that each contain similar information about a target. An improvement to this would be a filter that considers feature relationships, selecting features with the highest performance when combined. Such filters include, but are not limited to, minimal Redundancy Maximal Relevance (mRMR) [113], feature clustering [9, 87] and particle swarm optimisation [16, 86], each of which considers feature relationships as well as their relationship to the target.

In any filter method for feature selection, the relevance and redundancy of features must be quantified. In some cases, where only linear relationships are of interest, it can be quantified by Pearson's Correlation Coefficient (PCC). Most domains contain non-linear relationships, which can be quantified using information based correlation estimates such as Mutual Information (MI) [160], which is discussed in detail in Section 4.4. To apply these information based approaches to numeric or continuous data the probability density functions of the variables must be estimated and integrated, and is non-trivial [81, 141]. One method for estimating entropy with continuous data is to use a Parzen window, but this requires the selection of parameters that cannot be determined easily from the data [58]. Another more simplistic approach that is adopted in this thesis is to discretise the variables [31, 58].

MI increases with the dimensionality of features, which causes rankings to bias towards features with higher numbers of distinct values. Symmetrical Un-

certainty (SU) aims to normalise for this bias by dividing the MI value by the entropy of the features. SU still prefers features with many values in some conditions, however, so the bias is not fully mitigated. Another approach to normalising against this bias is to use permutation methods [4, 37], which are explored in Chapters 4, 5 and 6.

Gretton et al. [42] introduced Hilbert-Schmidt Independence Criterion (HSIC) to quantify non-linear relationships between variables and Song et al. [137] offer proofs to show that it is unbiased. HSIC is a kernel based method to quantify the relationships primarily between continuous variables, and is often applied using the Radial Basis Function (RBF) kernel [137]. Where the forms of relationship are known, or if the variable is discrete, other kernels exist [137]. It is common also for one kernel to be used for features and another to be used for the target variable in a classification task. Applying the kernels to each feature individually is expensive computationally, and is often infeasible even with the Cholesky decomposition [7]. HSIC and the Cholesky decomposition is discussed in further detail in Section 4.4.

2.1.8 Algorithm engineering

The algorithm engineering stage encompasses the selection of the learning algorithm and its parameters. For most problems there are several suitable learning algorithms available that can be applied with little or no adaptation [70, 160]. Such common algorithms include Naïve Bayes, Decision Tree, Random Forest, Support Vector Machine (SVM) and many others [160]. Whereas these do not model temporal artefacts, and rely on information captured in extracted features, others such as Recurrent Neural Networks or Hidden Markov Models are able to capture historical trends implicitly.

Many learning algorithms also require parameters to be set. For example, the C4.5 Decision Tree [160] takes parameters that determine how aggressively the tree should be pruned and Random Forest has parameters to set the details of the bagging strategy to use, including the number of trees to use in the

ensemble. These parameters can be optimised using an exhaustive or greedy search, to find those that produce the model with highest performance, but this can lead to over-fitting as it is a multiple comparisons procedure [65]. Strategies to avoid multiple comparisons procedures or mitigate their effects are discussed in Chapter 4.

2.2 The data mining approach: Evaluation and refinement

Decisions made in the steps described in Section 2.1, prior to the evaluation and refinement stages, form a *learning approach* that defines methods for processing and learning from data to produce predictive models. Here, this learning approach is evaluated with respect to the problem definition, in order to estimate its expected performance in reality. The evaluation of a learning approach on a database can be split into two parts, namely the structure of the evaluation procedure, and computing a metric of performance for the learning approach [63]. The evaluation structure defines how a learning approach is applied on the data, and the produced model is then used to make predictions on testing samples. These predictions are then compared against the ground truth to compute metrics that describe the performance of the model and learning approach.

2.2.1 Structure of evaluation

In an evaluation, a learning approach must be applied on training data to build a model that is then used to make predictions for testing samples, which we refer to as a *train-test cycle*. A train-test cycle should be performed several times with different training and testing samples to produce a more robust performance estimate. There are two approaches to this in general, namely k-folds cross validation and random subset validation [50, 63, 160]. In k-folds cross validation the data is split into k sub-samples, of which $k-1$ are used as the training dataset in each iteration. The remaining sub-sample is used

as the testing dataset, and each test sample is used in exactly one train-test cycle. Random subset validation, sometimes referred to as bootstrap validation [160], repeats k times the train-test cycle with different random sub-samples of the same sizes for the training and testing datasets. Whereas in k-folds cross validation the size of the training dataset is determined by the number of folds, in random subset validation it can be chosen independently of the number of train-test cycles. Again, samples not used in training are used as testing data. With too few samples in the training data a learning algorithm will be unable to build a model to capture the underlying concepts in the data, and result in producing pessimistic performance estimates. If the proportion of training samples is too high when compared to testing samples, the evaluation may over-fit to the data and produce optimistic performance estimates.

In k-folds cross validation, or in each train-test cycle of random subset validation, the subsets must be randomly sampled [63]. If the data is non-temporal or has very low autocorrelation, a simple random sub-sample of the data can be used for training and the remaining samples used for testing, as these will provide distinct sets of samples. In cases with high class imbalance, a stratified sampling procedure may be considered to maintain the class distributions in the training and testing datasets. Vehicle telemetry data has high autocorrelation, however, and the values of many signals such as $VehicleSpeed$ or target variables such as road type, change rarely with time. This means that random sampling results in the training and testing datasets containing effectively the same samples, even though they are in fact distinct.

Autocorrelation in data is reduced through linear sub-sampling, where every t^{th} sample is taken [27], but the amount of sub-sampling required is difficult to determine, and the issues caused by autocorrelation may not be rectified fully. When evaluating models for temporal data in this thesis, therefore, contiguous blocks of samples from journeys or drive cycles are sampled. There are two methods for doing this, which can be used to investigate different hypotheses. First, a number of whole journeys or drive cycles can be taken to form the

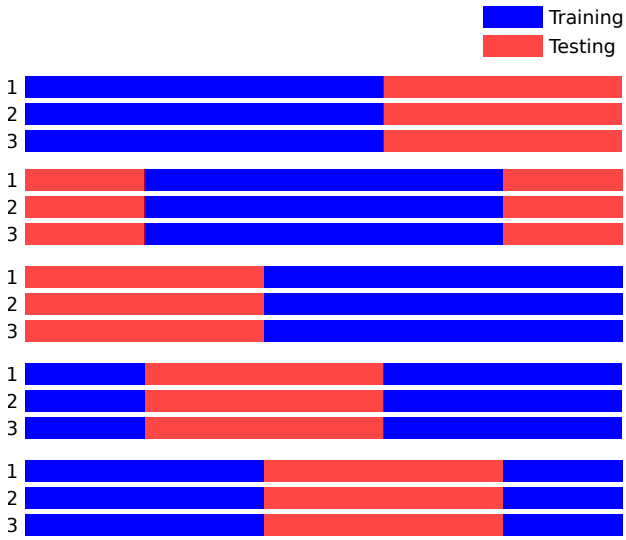


Figure 2.2: A temporal evaluation structure, where the training data is taken from the same section proportionally of each journey or drive cycle. Each set of three bars represents one train-test cycle, and each bar represent the data collected in one journey. The sections of the journeys that are used as training datasets are shown in blue, and the sections used for testing are shown in red.

training data, and samples from the remainder used as testing data, which is similar to the subject-level training adopted by Zhang et al. [173]. This investigates whether a model can be built using data collected on one journey and used on future journeys, which may be data collected with different drivers or vehicles. Second, a contiguous block of samples can be taken from the same position proportionally in each journey or drive cycle and combined to make the training data for each train-test cycle, as pictured in Figure 2.2. Here, the remaining samples from each drive cycle are then used as testing data. This approach is similar to the segment-level training used by Zhang et al. [172], but with only one contiguous training block. In each of the k iterations, the beginning of the training data is at $\frac{i}{k}$, where i is the iteration index starting with 0 to ensure an even spread over the train-test iterations. In the first iteration the beginning of the training data is at time 0 in each drive cycle. For example, the training data for the second iteration of 20 begins after $\frac{1}{20} = 5\%$ of the journey length, and the 20^{th} begins at $\frac{19}{20} = 95\%$. Although individual blocks of samples have a high linear autocorrelation, the correlation between them will be low to produce training and testing samples that do not have the same values.

Regardless of how the data is sampled, any analysis in the learning approach must be performed on the training data only [50, 63, 160]. This includes the computation of discretisation levels, selection of model inputs, and optimisation of model parameters. Any information taken from the testing data may cause optimistic performance estimates. Furthermore, although the performances on testing datasets can be used to select the best learning approach, the performance estimate produced will be optimistic as the testing data is used in its selection. To avoid this, and create an unbiased performance estimate for the best learning approach, a third validation dataset should be used. A validation dataset can be generated by removing samples from the training data and acts as a testing dataset in train-validation cycles. Train-validation cycles can again be performed several times to increase the confidence in performance estimates for selecting the best learning approach. Once the best learning approach is

selected based on its performance on validation data, it can be used to build a model on the original training data and its unbiased performance estimated using the testing dataset.

2.2.2 Performance measures

To estimate the performance of a model, the predictions it makes for testing samples are compared to their ground truth [63, 65, 160]. The performance can be reported graphically with lift charts, Receiver Operator Characteristic (ROC), or detection error tradeoff curves, or it can be summarised as statistics such as Success Rate (SR), Precision, Recall, and Area Under the Receiver Operator Characteristic Curve (AUC). The choice of measure again depends on the problem definition, domain and database, as each measure provides a different insight into the performance of a model. SR, the number of correct predictions made by a model, for example should be avoided in class imbalance domains. This is because a high SR can be achieved by a model that always predicts the majority class, but this model is useless in reality.

The issue of class imbalance is rectified by AUC, which has been adopted by many researchers [59, 63] and is used throughout this thesis. AUC is computed as the integral of ROC curves, which are sometimes referred to as threshold curves [160]. A ROC curve is computed by plotting the true-positive rate (percentage of correct positive predictions) against the false-positive rate (percentage of incorrect positive predictions) for multiple decision thresholds. A decision threshold is the threshold at which a probabilistic prediction from a model is determined to be positive or negative. A decision threshold of 0 means that all samples are labelled as negative by the model, whereas all samples are labelled as positive for a decision threshold of 1. This use of multiple decision thresholds mitigates for class imbalance in the testing data, producing a more representative performance measure of the model.

Although AUC is a good measure of performance in many situations, it should not be used to compare respective performances of different kinds of

predictive model [49]. In particular, it should also be avoided if the ROC curves of models being compared cross, as the AUC of one model may be higher even though the other may have better performance for the majority of decision thresholds. Implementations and uses of the H-measure are uncommon, however, so in this thesis performance comparisons are made with the same learning algorithm and care is taken to ensure the ROC curves do not cross.

2.2.3 Refinement

Unfortunately, the data mining process rarely works out the first time [70]. The domain expert, on seeing performance results and inspecting a model, may not be satisfied with its outcomes. The problem definition may not have been quite right, the data collected may not be representative of reality, or the expert may not be comfortable with a particular facet of the model. In any of these cases, the process should be refined and iterated upon so that the best performance and solution can be found. If the performance of a model is not acceptable, the problem definition might be changed to be less strict. If this is not possible, the expert may suggest new inputs that were previously considered as useless or too difficult to attain before. In each iteration, the practitioner should hope to get closer to what the expert really wants.

2.3 Automotive applications

The following sections discuss two applications of the data mining process in the automotive domain, namely driving conditions monitoring in Section 2.3.1 and driver monitoring in Section 2.3.2.

2.3.1 Driving conditions monitoring

Driving conditions problems relate to the outside environment, including the road terrain and quality [25, 41, 104, 126, 134], as well as traffic levels and road types [14, 60, 83, 103, 144, 157]. Terrain is defined by the materials that make

up the road [119], but also by the surface quality [25]. Roads are commonly made from asphalt, concrete, gravel, sand, and many other materials. Although the surface material is a good indicator of quality, the number of cracks and pot-holes also has an influence. After a cold winter, for example, an asphalt road without full repairs may be in poor condition, even though in general it is a high quality road surface.

Traffic levels and road type can be defined in several ways, including level of service [14, 83, 103], descriptive [53, 60, 117, 142, 144], and government classification [144]. Possibly the most used definition in research is that provided by Carlson and Austin [14], based on level of service and driving cycles. Level of service and driving cycles are qualitative measures describing observed operational conditions [83], and therefore may be subjective. Descriptive definitions are of most use as they have a direct relationship to the current situation and environment. For example, Huang et al. [60] use the labels highway, urban road (both congested and flowing), and country road. Hauptmann et al. [53] use an even more direct classification structure, based upon current car behaviour. Their five labels range from very fast, straight line driving on flat roads, to very low speeds or stop. These are used to represent further driving situations, such as highway driving, and traffic lights or parking.

Wang and Lukic [157] provide a survey for driving conditions prediction, with the focus on Hybrid Electric Vehicles. They recognise that many researchers use drive cycles for a road definition, and use only information from the vehicle speed in their models. For example, average velocity and acceleration, as well as peak accelerations and percentage of time in certain speed intervals are often used [60, 83, 103, 111]. These features are also often extracted from 150 seconds of data in order to produce good classification performances [157]. These approaches have clear limitations in determining the current driving conditions. First, steering wheel behaviour is likely to differ in different situations, providing additional predictive information. Second, if features are extracted from large amounts of temporal history, the model is likely to be slow to react to changes

in environment.

Other authors have used different features in addition to those extracted from speed cycles. Hauptmann et al. [53], for example, utilise engine speeds, accelerations, and gradient. Additionally, Qiao et al. [116] extract features from the pedal positions, temperatures and selected gear. These features, however, although they contain different information from the vehicle speed, are all related to it. Engine speed, for example, has a correlation with vehicle speed of 0.96 on data we have collected, meaning that it is adding little new information into the system. Qiao et al. [117] note that the length of the temporal window that features are extracted over is an important factor in the system's reaction time and they use a much smaller window length than in other works, of 6.25 seconds. One shortfall in their work, however, is that automatic feature selection is not performed and features are selected based on the intuition of the researchers.

To ensure that model inputs are relevant to driving conditions, Huang et al. [60] perform Analysis Of Variance (ANOVA), and cross correlation analysis to remove redundancy. They investigate 11 features in total, with only four being manually selected for classification. Murphey et al. [103] and Park et al. [111] proposed a selection procedure based on binary class separability of single features, such that if a feature is able to distinguish one class label from the others, then that feature is selected. When dealing with CAN-bus data, however, the number of signals and features can be in the order of 1000s, meaning automated approaches are necessary [145].

A final approach to the problem of driving conditions monitoring is the use of visual inputs, e.g. from cameras mounted on the vehicle or roadside, and applying image processing techniques [62, 104, 119, 134]. In their work, Tang and Breckon [142] use colour, texture and edge features from image sub-regions as inputs into a neural network, and using colour analysis, Jansen et al. [62] identify the terrain type. Also in identifying the terrain type using colour and edge features, Raj et al. [119] do not use automated learning, but develop an

algorithm using domain knowledge. Such systems are limited because they rely on non-standard sensors, generally need greater computational processing and are severely affected by poor lighting conditions, such as night-time driving. To overcome this, without the use of expensive infra-red or laser technology, Shibata et al. [134] process images taken by road side cameras that are illuminated by the headlights of passing vehicles to determine whether the road surface is wet or not.

2.3.2 Driver monitoring

Driver monitoring aims to determine parameters of the driver, which can be categorised as those of driver intentions, driver characteristics, and driver inattention [74, 114]. Predicting the intentions of a driver aims to determine their next likely action, and estimate the likelihood of the driver pushing the brake pedal, changing lane, or turning behaviour at the next intersection [6, 54, 73, 97]. Knowledge of driver intentions can be used for both increasing the efficiency of the vehicle and for improving safety systems, by priming relevant devices in the vehicle to anticipate the driver's actions [95, 96]. For example, if a lane change is imminent the indicators may automatically be turned on, or the engine may reduce its revolutions prior to a braking event. A second form of monitoring is to characterise the driver, to either personalise the vehicle to the driver or their skill level [101, 128, 173]. The type of driver can influence the vehicle settings to increase comfort or efficiency of the engine, while insurance companies have an interest in the skill level to personalise insurance.

The final form of driver monitoring, which is a focus of this thesis, is the monitoring of driver inattention. Driver inattention in general increases the risk of a crash, and so understanding the causes of inattention or determining whether a driver is attentive or not is a major safety concern [23, 74, 99]. Regan et al. [125] developed a taxonomy for driver inattention, dividing it into *diverted* (performing tasks unrelated to driving), *restricted* (fatigued or unwell), *misprioritized* (prioritizing unimportant driving tasks above critical tasks), *neglected*

(lack of due care because of familiarity to the road environment), and *cursory* (rushed or panicked driving). In a more simple categorisation, Dong et al. [23] provides two broad types of inattention, namely *distracted*, where the attention of the driver is placed in activities unrelated to driving, and *fatigued*, where the attention of the driver is reduced or and performance is impaired.

Distractions, whether the activity is related to driving or not, can in general be categorised broadly as one or more of *visual*, *auditory*, *physical* or *cognitive* [23]. For example, programming a satellite navigation system via a touch screen induces physical, cognitive and visual distractions on the driver. Other distractions include eating, listening to the radio, setting the climate control, conversing with passengers or using a mobile phone [124], and all have different impacts on attention and crash risk. The use of a mobile phone while driving, for example, impacts attention very severely and increases the risk by four times over a 'normal' level of distraction (such as listening to the radio or conversing with passengers) [122, 123]. Furthermore, there is some evidence to show that this risk does not decrease when a hands-free phone is used and the physical distraction is removed [122, 124], indicating that some forms of distraction have different impacts than others.

Fatigue also can be of different degrees, from moderate tiredness to falling asleep [23]. It is generally induced on long journeys, potentially with monotonous roads, where the will of the driver to continue waivers. Fatigue is therefore different to distraction, as attention is reduced in general rather than diverted to other tasks. It increases the time taken for a driver to react to events on the road, increases the likelihood of micro-sleeps and in some instances cause the driver to fall asleep [90]. As a result, fatigue is estimated to increase by five times the risk of a crash [90] and is a contributing factor in up to 25% of serious crashes in the UK [34].

In cases where real-time analysis of workload levels are not required, questionnaires can be used. The NASA-Task Load Index (TLX) [52] asks participants of a trial to rate on a scale their mental demand, physical demand,

temporal demand, performance effort and frustration. The TLX was originally developed to assess workload in the aviation domain, but has since been applied in much of human factors research. Because some of the TLX dimensions are less relevant to the automotive domain, or are too broad to fully represent workload, Pauzie [112] introduced the Driver Activity Load Index (DALI). The DALI also asks participants to rate their experience during a trial in six dimensions, and these are attentional effort, visual demand, auditory demand, temporal demand, interference of secondary tasks, and situational stress. Even though the DALI was developed specifically for driving tasks, the TLX is often still used in automotive research and is used in Section 3.3 for the Warwick-JLR Driver Monitoring Dataset (WarwickDMD). These workload indexes are subjective, and rely on drivers remembering accurately their experiences during a trial.

Task performances usually decrease with higher workloads, and can therefore be used to estimate workload [47]. Common performances used involve tertiary tasks, unrelated to the driving task or secondary distraction task. One example of such a task is the tactile detection response task, where a buzzer provides a stimulus that the participant must detect [169]. The idea is that under higher workloads, fewer of the stimuli will be noticed by the participant. Variations on this task include a visual, and auditory detection response tasks, where the participant responds to visual and auditory stimuli. Another task performance measure is the occlusion method [106, 139], in which a task is performed while the visual attention capacity of the participant is limited. Usually, the participant wears goggles with shutters that close intermittently and block the participant's vision. Because visual contact with the task is limited by the shutter, tasks can be evaluated for visual demand without participants having to drive or use a simulator. This removes several factors that are uncontrollable in other studies, including the behaviour of other vehicles on the road or willingness of the participant to divert attention from the driving task to the secondary task. These methods cannot be used in measuring workload in real

time, however, as the measurements require large time periods for the measures to be sensitive to cognitive workload.

Analysis of cognitive and physical workload while driving is often performed using both vehicle telemetry data and other physiological measurements, such as Heart Rate (HR), Heart Rate Variability (HRV), Skin Conductance Level (SCL) or eye blink parameters [23, 88, 99, 147]. In particular, when a driver is experiencing increased workload due to distraction or is fatigued, changes can be observed in features of the Steering Wheel Angle (SWA) [27, 56, 90, 99, 152, 161]. Other driving performance measures related to SWA, such as the deviation of the vehicle from the lane markings, have also been shown to be successful indicators of driver inattention. Likewise, the HR and SCL of a driver increases during periods of higher cognitive load [99]. When fatigued, drivers tend to blink more often and for longer, their pupils dilate and their grip on the steering wheel relaxes [23].

Many studies focus predominantly on the physiological effects of inattention, and consider relatively few performance measures or telemetric signals available via the CAN-bus [33, 57, 68, 99, 127, 128, 161]. This is likely due to the higher responsiveness of physiological measures to changes in the attention state of a driver, and in particular cognitive or mental workload [99]. Sensors to measure most physiological parameters, including Electrocardiogram (ECG), Electrodermal Activity (EDA) and Electroencephalography (EEG), are often intrusive and require large pieces of technical equipment. An accurate ECG measurement, for example, requires at least three electrodes to be placed on the driver's chest and connected to an amplifier. Eye parameters, including blink frequency and speed, pupil dilation and gaze detection, can be recorded without intrusion via a video camera mounted on the dashboard [12, 66, 68, 88]. Video feeds are often unreliable in poor light situations, however, and even infra-red cameras are often impotent if the driver wears certain types of glasses [68]. For these reasons, we do not consider them appropriate for monitoring driver inattention on a daily basis.

The use of telemetry data for driver inattention monitoring is usually considered as a secondary input to predictive models that rely heavily on physiological measures (e.g. [161]). Common model inputs include features extracted from the steering wheel, vehicle speed, and pedal positions [99, 161]. Features, such as means or Standard Deviation (STD)s, are often extracted from signals over the whole distraction or normal driving periods, which are often minutes long. For example, Mehler et al. [99], present a statistical analysis of mean values of the heart rate, skin conductance level and vehicle speed, and STD of steering wheel reversal rates and gaze dispersion. Although these results show that features of physiological and telemetric signals share a relationship with inattention, they are of little use in a real-time detection system as distraction states change in a matter of seconds. Other authors, such as Tango and Botta [143], Torkkola et al. [152], Wollmer et al. [161], present models that process these signals in smaller windows to output the distractedness of the driver.

2.4 Summary

This chapter has provided a background on data mining for this thesis and introduced the automotive applications that are used as examples in this thesis. The general data mining process described is used in all experiments throughout this thesis, with changes made to certain stages. In particular, the feature selection stage (Section 2.1.7) is advanced in Chapters 4, 5 and 6, to develop a selection method capable of selecting high performing features from telemetry data. In Chapter 5 temporal issues of the process are disregarded and the feature extraction stage (Section 2.1.4) is not considered, as non-temporal datasets described in Section 3.4 are used. The two applications, namely driving conditions monitoring and driver distraction monitoring, are central to this thesis. One dataset for road type classification (a driving conditions problem) is described in Section 3.1, and two for driver distraction monitoring are presented in Sections 3.2 and 3.3.

CHAPTER 3

Datasets

This chapter describes the datasets that are used throughout this thesis. Three vehicle telemetry datasets have been developed for this thesis, namely the Road Classification Dataset (RCD), Coventry-JLR Driver Monitoring Dataset (CoventryDMD), and Warwick-JLR Driver Monitoring Dataset (WarwickDMD). The RCD describes an environment monitoring problem and is introduced in Section 3.1. Details of the CoventryDMD and WarwickDMD, two datasets that describe different types of driver distraction and cognitive load, are provided in Sections 3.2 and 3.3 respectively. Details of datasets from domains other than automotive that are also used in this thesis are described in Section 3.4.

3.1 Road classification dataset

The RCD describes the environment monitoring problem of road classification. It was recorded using a Video VBOX Pro[1], which allows video streams from multiple cameras to be recorded and synchronised with a subset of the Controller Area Network (CAN)-bus data as well as Global Positioning Satellites (GPS) location and time data. In this case, the 17 signals listed in Table 3.1 were recorded along with GPS data, both at a constant frequency of 20Hz. The VBOX PRO interpolates signals to a constant frequency by taking their last-observed value at each time step. The dataset is available for download via `www.dcs.warwick.ac.uk/dmd/` in a comma separated variable (csv) format.

The data was collected over 16 drives across the Midlands, UK, using two cars. Each journey involved at least one driver, with a mean journey length of

[1] `http://www.vboxmotorsport.co.uk/index.php/en/products/video-loggers/video-vbox-pro`

Signal	Description
Ambient temp	Outside temperature (measured behind grill).
Brake pressure	Pressure on brake pedal.
Gear position	(Automatically) selected gear.
Longitudinal acceleration	Forward acceleration of the vehicle, measured by an accelerometer.
Lateral acceleration	Side-to-side acceleration of vehicle, measured by an accelerometer.
Suspension height (for each wheel)	Heights of suspension (Front-Right, Front-Left, Rear-Right and Rear-Left).
SWA	Angle of steering wheel.
SWA speed	Rate of change of SWA.
Vehicle speed	Vehicle speed (measured from wheel speed).
Wiper status	Speed status of the front window wipers.
Latitude	Latitude location coordinate.
Longitude	Longitude location coordinate.
GPS satellites	Number of satellites connected to GPS sensor.
Time	Time received from GPS satellite.

Table 3.1: List of signals recorded in the RCD.

51 minutes, which is comparable to the length of data used by Huang et al. [60]. Two ground truths, carriageway type and road type, were derived using the GPS data and applied by hand using Google Earth[2]. The GPS longitude and latitude coordinates were looked up in Google Earth, and a label was decided and assigned to samples. For the carriageway classification, the number of lanes is decided by looking at the satellite images provided. If there is more than one lane, the sample is *dual*, otherwise it is *single*. For road type, the road name is looked up on the map and the first letter taken. Roads names in the UK begin with *A* (for arterial or trunk roads that allow all types of traffic), *B* (for smaller local roads), or *M* (for motorways where certain types of traffic are prohibited) [22]. Roads of other classifications, including unnamed roads, are given the label *C*.

The distribution of labels for the RCD, which is heavily imbalanced, is provided in Table 3.2. The binary classification task of carriageway type has an imbalance of 5.6 : 1, which may bias classification models towards labelling samples as *Single*. Likewise, the road type classification task is imbalanced with

[2] `http://www.google.co.uk/intl/en_uk/earth/`

Label	Percent (%)
Single carriageway	85
Dual carriageway	15

(a) Carriageway labelling

Label	Percent (%)
A road	48
B road	26
C road	21
Motorway	5

(b) Road labelling

Table 3.2: Label counts for the (a) carriageway and (b) road ground truths.

almost half the samples being of classification A, and only 5% being motorways. There is also a degree of overlap between the classes, especially between A and B roads. This is because the classification system used is designed to be relative to the needs of the local area, and B roads perform the same role as A roads, only to a lesser extent [22].

3.2 Coventry-JLR driver monitoring dataset

The second vehicle telemetry dataset used in this thesis was collected during a previous study performed by Dr Graham Shelton-Rayner and Dr Helen Maddock of Coventry University, and made available to us through JLR. In this track study, participants drove a Range Rover Sport under both normal and distracted conditions. The track is located at JLR's principal engineering facility at Gaydon, Warwickshire, UK, and is pictured in Figure 3.1. It is a simulated highway of around 3.8 miles with four lanes and two main straights with two major corners at the end of each. In comparison to public roads it is quiet, as it is used solely by automotive engineers for research and development purposes. The participants are instructed to drive in the second lane at usual highway speeds of around 70mph, changing to an outer lane to overtake when necessary.

During the study, six types of data were recorded, namely: GPS and CAN data, Electrodermal Activity (EDA), Electrocardiogram (ECG), Heart Rate (HR) via a sports watch, and a video stream with forward and driver facing cameras. Because of movements in the cabin, associated with the physical

Figure 3.1: Map of the Gaydon emissions track used for the driver monitoring trials.

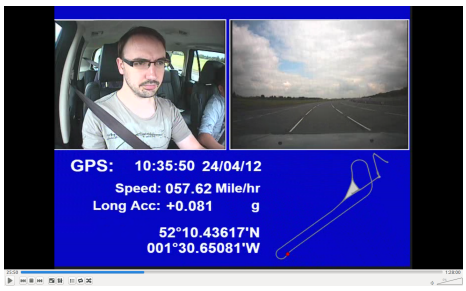


Figure 3.2: Screen shot of the video recorded during the trials, with driver and forward facing cameras, as well as GPS details overlaid.

distraction tasks performed, the EDA and ECG data were too noisy for use and were therefore discarded. Further, the data from the sports watch was not fully synchronised with data from the CAN, and therefore was also discarded. Over 1500 telemetry signals were recorded using a data logger that collects all messages sent over the CAN, which were later post-processed to be of a constant frequency of 10Hz. This was done using the CANalyzer[3] software, which is a utility for exploring CAN-bus data, and exporting it to other formats. At this sample rate, 494 of the telemetry signals contained information that may be of use in predicting the target and were retained for further processing.

To impose distraction on the driver, a series of tasks, as listed in Table 3.3, were performed at different intervals. A ground truth was applied to the data using the video streams (shown in Figure 3.2), and was synchronised via the

[3] `http://vector.com/vi_canalyzer_en.html`

Secondary Task	Description	Mean duration (s)	STD
Select Radio Station	Selection of a specified radio station from presets	70.4	48.0
Mute Radio Volume	The radio is muted or turned off	4.9	2.4
Number Recall	Recite a 9 digit number provided before the drive	83.1	46.6
Navigation	A specified destination is programmed into the in car Sat-Nav	111.5	45.6
Counting Backwards	Driver counts backwards from 200 in steps of 7 (i.e.., 200, 193, 186...)	118.3	45.4
Adjust Temperature	Cabin temperature increased and then decreased by 2°C	26.7	25.1

Table 3.3: Secondary tasks the driver was asked to perform. If there is a secondary task being performed, the data is labelled as *Distracted* for the duration, otherwise it is labelled as *Normal*. Tasks were performed in the same order for all experiments, with intervals of between 30 and 300 seconds between tasks.

GPS times also present in the CAN. For the duration of a task, the data is labelled as *Distracted*, otherwise it is labelled as *Normal*. In this study there were 8 participants, each driving for approximately 1.5 hours during which each of the 6 tasks are performed twice and each lasting for the durations listed in Table 3.3. Each participant was therefore performing each task for twice the listed times on average, and were driving with no task for an average of 2647.4 seconds. In addition to the tasks listed in Table 3.3 participants also performed two driving manoeuvres, namely abrupt acceleration and a bay park. The data from these are, however, considered to be unrelated to distraction and therefore can be viewed as noise and were removed from the dataset. This removal was done after feature extraction to avoid temporal continuity issues.

3.3 Warwick-JLR driver monitoring dataset

A second data set for driver monitoring was collected by the author and for this thesis, in similar circumstances to the CoventryDMD [146, 149]. In this study, the physical tasks that made up the majority of those in the CoventryDMD were not used, and only a cognitive task was considered. This minimised the

movement of subjects during the trials, and enabled their hands to remain on the steering wheel at all times. This mitigated the issues found with the CoventryDMD discussed in Section 3.2, and meant that the ECG and EDA data streams could be processed successfully.

3.3.1 Collection protocol

The experimental protocol we use is based on that performed by Reimer et al. [127] and Mehler et al. [99], and is outlined in Table 3.4. In their work, changes in physiology and driving style are observed while the driver is performing the N-back test [99, 127] as a secondary task to driving. The main difference in our protocol is that we perform it on a test track and the ECG electrodes are placed on the chest rather than the lower neck. Also, we use gel EDA electrodes with adhesive pads, as we have found these to be more stable and, in our experience, produce a cleaner signal.

When the participant first arrives at the trial location, electrodes are attached for both the ECG and EDA measurements. After this, the participant is taken to the vehicle and seated in the driving position. Once the seat, steering wheel, and mirrors are adjusted as appropriate, data recording is commenced. The protocol then continues with checking that the sensors are providing a clean and reliable signal, followed by practice runs of the N-back test (stages 1 and 2).

The N-back test requires the participant to repeat digits provided to them in a list with a delay. Here it is operated with three forms of increasing difficulty, with delays of 0, 1 and 2 and referred to as the 0-, 1- and 2-back tests respectively. These three difficulty levels have been shown to have an increasing impact on the participant's physiology and driving style [99, 127]. In the 0-back test, the participant is required to repeat digits back as they are said. The 1-back test requires the participant to repeat the digits with a delay of 1, and the 2-back test with a delay of 2. Each task is presented in 4 blocks of 10 digits, with a time separation between each digit of around 2 seconds. An example

	Stage	Mean duration (s)	STD
1	Habituation	1302	269
2	Baseline	280	99
3	0-back (introduction)	10	2
4	0-back	82	9
5	0-back (recovery)	256	59
6	1-back (introduction)	10	2
7	1-back	100	12
8	1-back (recovery)	300	81
9	2-back (introduction)	11	6
10	2-back	113	15
11	2-back (recovery)	294	127

Table 3.4: The protocol for the WarwickDMD experiment, employing three N-back tests of different difficulties, presented in a random order to each participant.

Stimulus	1	5	9	3	0	2	3	3	2	9	&	&
0-back	1	5	9	3	0	2	3	3	2	9		
1-back	-	1	5	9	3	0	2	3	3	2	9	
2-back	-	-	1	5	9	3	0	2	3	3	2	9

Table 3.5: Example of the N-back test with a block of 10 numbers. In place of "&" the word "and" is said by the experimenter, requiring the participant to provide a response. Where there is a "-" no response is required by the participant.

block of 10 digits is shown in Table 3.5, with expected responses for the 0-, 1- and 2-back tests. In order to continue with the experiment, the participant must show a minimum proficiency of 8 out of 10 correct responses for two consecutive blocks of each task.

The vehicle is first driven onto the track by the participant and data recording is commenced. Because this is likely to be an unfamiliar vehicle and a new environment for the participants, a habituation period of driving under normal conditions is used (stage 1). Once the habituation period is completed and the driver is comfortable on the track, a reference period under normal driving is used (stage 2). At stage 3, after this reference period, the protocol alternates between N-back tests and recovery periods of normal driving (stages 3–11). Each

participant undergoes each of the 0-, 1- and 2-back tests in a random order. Each of the N-back tests consists of 4 blocks of 10 digits, with a block separation of 5 seconds. Before each N-back task, a brief explanation and reminder of it is provided (stages 3, 6 and 9), taking around 10 seconds. The recovery periods are each of normal driving, with no secondary task. Once each task has been performed and the final recovery period has taken place, the vehicle is then taken off the track and data recording is ended.

Because all digits in the N-back tasks were repeated regardless of the shift (in contrast to Mehler et al. [99]), the 1-back task was in effect one digit longer and the 2-back task was two digits longer than the 0-back task. This is reflected in their mean durations shown in Table 3.4. Other variances in durations were due both to safety concerns, recording quality or human variations. Some events on the road such as low flying birds or overtaking vehicles, for example, caused reactions from the driver that were both out of the control of the experimenter and led to a pause in the protocol or the extension of a stage.

3.3.2 Data collection

Over 1000 signals were recorded from the vehicle's CAN-bus. Those signals which are expected to have relevance to driver workload include, steering wheel angle, pedal positions and vehicle speed. Many others are likely to be of no relevance, such as the window wiper speeds or air conditioning controls, and should be removed before attempting to predict driver workload. However, to ensure that all the relevant signals are present in the dataset, we recorded the full set of signals at a sample rate of 20Hz during the experiment. Each of these signals was written to a hard disk by a data logging system located under the passenger seat. As with the CoventryDMD, video with forward and driver facing cameras was recorded in the same format as in Figure 3.2.

Three point ECG gel electrodes were attached on the driver's chest, close enough together to minimise any noise generated through shoulder movement. The EDA electrodes were attached on the underside of the index and middle

fingertips of the participant's non-dominant hand. Surgical tape was then used to further secure them in place, minimising any movement of the sensor contacts while driving. The wires from the ECG electrodes came out of the top of the participants shirt, while the EDA wires were positioned to the side of the non-dominant hand. Note that the vehicle used has an automatic transmission and the driver does not need to use their hands for gear selection. To record this physiological data a GTEC USB biosignal amplifier (USBamp)[4] was used, which resides in the rear of the vehicle with sensor wires positioned away from any intrusion of the driver. This connects to a laptop, where the data was recorded at 256Hz using MathWorks Simulink[5].

From this data, there are five ground truths that we use to produce classification problems. These are extracted from the timings of the tasks during the experiment, the EDA signal, and the ECG signal. The timings of the tasks provides a ground truth of what the participant was doing at a given point in time, with reference to the GPS time shown in the video streams. The EDA signal provides two measurements, the Skin Conductance Level (SCL) and frequency of Electrodermal Responses (EDRs), both of which are known to increase while a participant is under high workload [10, 57, 99]. The SCL is provided by the absolute value of the EDA signal, whereas EDRs are found by spikes, as illustrated by the red dots on the EDA signal in Figure 3.3. The EDA sensor unfortunately requires configuration for each participant to ensure the value of the signal is within a measurable range, and depends on the quality of the connection between the electrodes and finger tips. This means that the absolute value of the signal cannot be directly compared across participants in the trial, as the magnitudes and range of the signal are different for each participant. Finally, two ground truths can be extracted from the time differences between R-peaks, highlighted by the red dots on the ECG signal in Figure 3.4. HR is calculated as the number of R-peaks per minute, and increases with workload in general [13]. There are several methods for computing Heart Rate Vari-

[4] `http://www.gtec.at/Products/Hardware-and-Accessories/g.USBamp-Specs-Features`
[5] `http://uk.mathworks.com/products/simulink/`

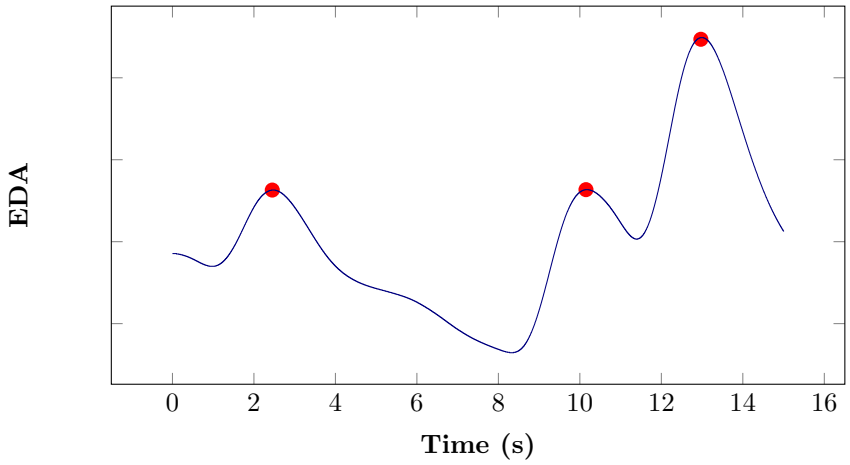


Figure 3.3: Fifteen seconds of an EDA signal recorded during driving. The dots highlight EDRs, which increase in frequency under workload. The SCL is given by the absolute value of the signal.

ability (HRV), however, including Discrete Fourier Transform (DFT) or wavelet analysis, time-domain methods and non-linear regression techniques [1]. In general it is the amount that the time delays between R-peaks vary, and here the Standard Deviation of Successive Differences (SDSD) is used as a result of findings by Mehler et al. [99], who found it to decrease with increased workload in a similar setting.

3.3.3 Preliminary analysis

In order to characterise the WarwickDMD, and to enable its use in a data mining process we performed an analysis of the raw data collected. Nominally, we performed analysis of the subjective ratings, analysis of the data streams with respect to the secondary tasks, and the production of ground truths to produce a classification problem.

Task performance and subjective ratings

The error rates for the digit recall tasks are shown in Figure 3.5. The number of incorrect responses for the 0-back test were very low on average, and there

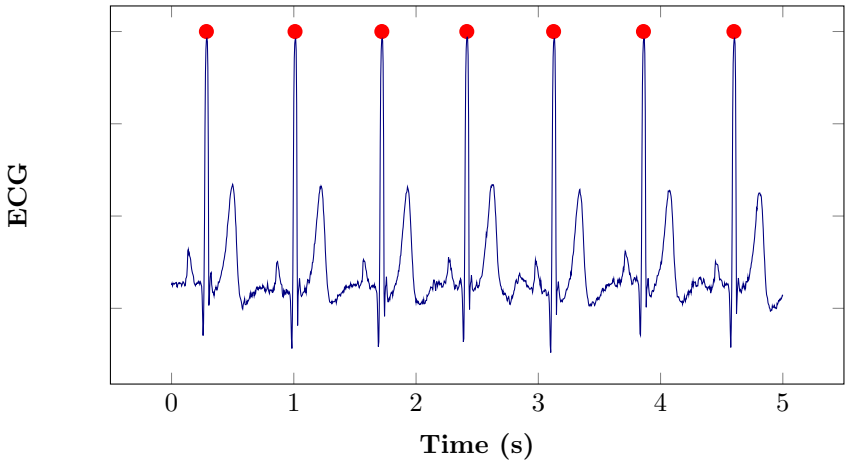


Figure 3.4: Five seconds of an ECG signal recorded during driving. The dots highlight the *R*-peaks, which can be used to compute the HR and HRV.

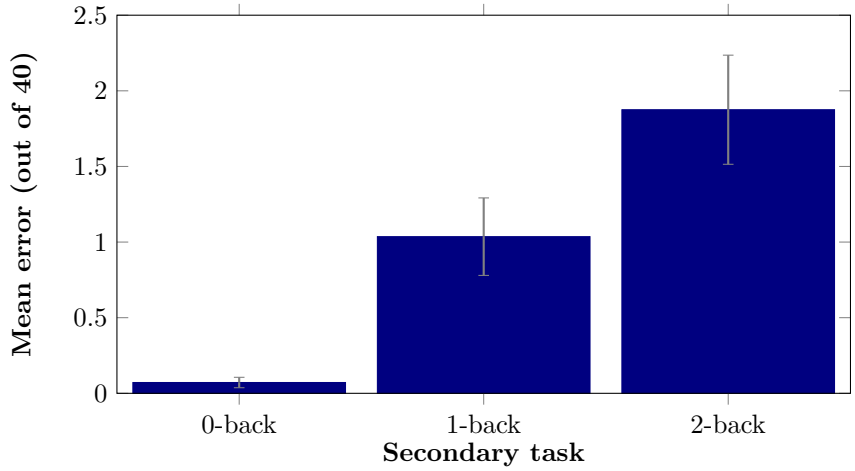


Figure 3.5: Mean error rates (out of 40 recalled digits) of participants for each of the secondary tasks. Error bars represent the standard error.

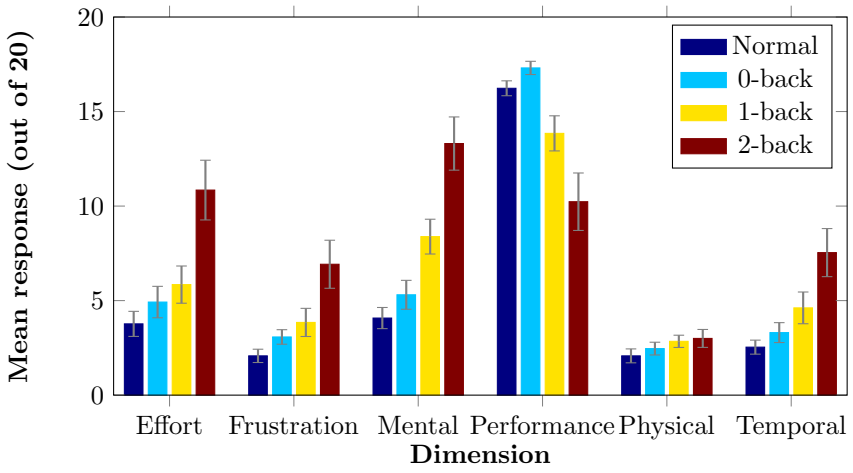


Figure 3.6: Mean responses to NASA TLX questions. Error bars represent the standard error.

were no errors for the majority of participants. In the 1-back test the number of errors was higher, and for the 2-back task there were even more incorrect responses on average. In some cases of the 2-back test the participant stopped responding to numbers of one block, and the remainder of block was counted as incorrect responses. In some other cases that were also counted as errors, the participant responded in the 2-back test as if it were the 1-back test.

When the protocol was complete, the participants were asked to fill in four NASA-Task Load Index (TLX) [52] questions – one for normal driving and for each of the N-back tasks. The TLX asks participants to rate their experiences out of 20 in 6 dimensions, namely: mental demand, physical demand, temporal demand, performance, effort, and frustration. These questions were used to confirm the tasks imposed appropriate levels of workload on the drivers, and their mean responses are shown in Figure 3.6. The TLX responses in general indicate that driving with the secondary tasks were harder, and that the difficulty increased with the delay in the digit recall tasks. The mental demand and effort dimensions, as expected, reported the largest increase in responses. The estimated performances decreased with the 1- and 2-back tasks, reported performance increased on average for the 0-back test over normal driving.

Signal	**Feature**	***p*-value**	**N vs. D**	**N vs. 0**	**N vs. 1**	**N vs. 2**	**0 vs. 1**	**0 vs. 2**	**1 vs. 2**
HR		**0.031**	**0.006**	1.000	0.422	**0.050**	1.000	1.000	1.000
HRV		0.554	0.283	1.000	1.000	0.996	1.000	1.000	1.000
SCL		**0.000**	**0.000**	1.000	**0.003**	**0.005**	0.452	0.537	1.000
EDR frequency		**0.034**	**0.004**	0.605	0.265	0.122	1.000	1.000	1.000
Adaptive Cruise Control Cancel (by brake)	STD	0.232	0.056	1.000	0.419	1.000	1.000	1.000	1.000
Brake on	STD	0.239	0.057	1.000	0.436	1.000	1.000	1.000	1.000
Engine Speed	raw	0.237	0.063	0.414	1.000	1.000	1.000	1.000	1.000
Engine Torque	raw	0.053	**0.016**	0.067	1.000	0.672	1.000	1.000	1.000
Engine Coolant Temperature	STD	0.190	**0.036**	0.362	1.000	1.000	1.000	1.000	1.000
Gear Selected (automatically)	raw	0.085	**0.012**	0.207	1.000	0.556	1.000	1.000	1.000
Steering Wheel Movement Speed	STD	**0.003**	0.066	1.000	0.087	0.055	**0.030**	**0.020**	1.000
Steering Wheel Angle	STD	**0.024**	0.471	0.555	0.423	0.968	**0.039**	0.095	1.000
Suspension Height (front-right)	STD	0.091	0.213	0.089	1.000	1.000	0.527	0.228	1.000
Throttle Position	raw	**0.044**	**0.010**	0.068	1.000	0.473	1.000	1.000	1.000
Yaw Rate	STD	**0.022**	0.532	0.422	0.679	0.715	**0.048**	0.051	1.000

Table 3.6: *p*-values from two way *t*-test and ANOVA for the physiological and selected signals of the vehicle telemetry data streams. In the heading **N** represents periods of normal driving, **0**, **1**, and **2** represents periods of the 0-, 1- and 2-back tests respectfully, and **D** is periods where any of the *N*-back tasks were being performed.

Analysis of data streams

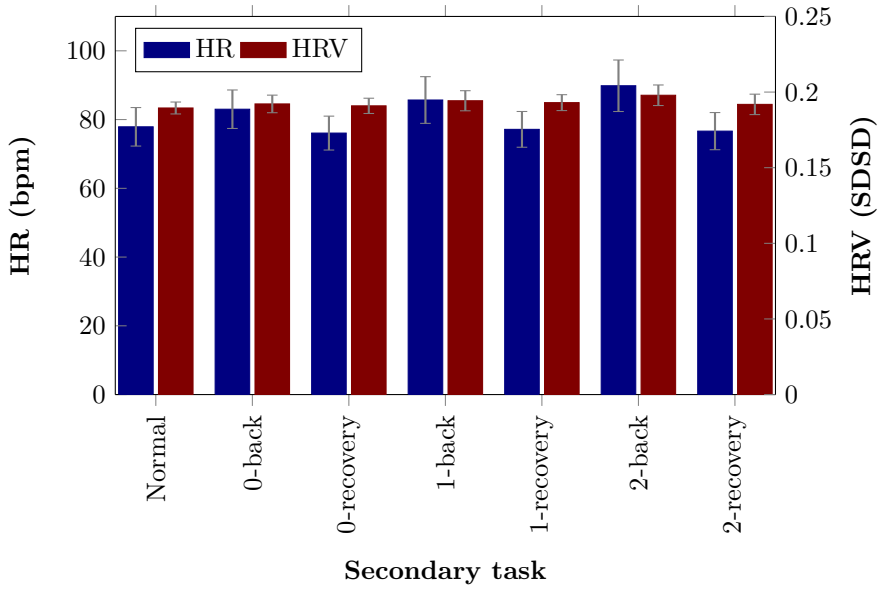
There were two data streams inspected, namely the physiological and vehicle telemetry data streams. Results of statistical analyses of both are shown in Table 3.6, comparing normal and distracted conditions in two ways to detail properties of the dataset. First, the mean of measurements over all subjects during normal (baseline or recovery) periods and distracted periods (during a secondary task) were compared using a two way t-test. Second, Analysis Of Variance (ANOVA) is used to determine if there was a significant difference in means during any of the three secondary task periods and normal driving. In follow-up to this, a four way pairwise t-test was performed and normalised by the Bonferroni correction. All results in this table produced p-values of less than 0.1 in at least one of the t-test and the ANOVA and any p-value smaller than 0.05 is highlighted in bold. The author accepts that conclusions made from this analysis are limited because it is a multiple comparisons procedure, but a two-way ANOVA, including all signals is impractical due to their number.

The physiological data consisted of the ECG and EDA signals, from which the HR, HRV, SCL and EDR frequency were extracted. The SCL during the baseline, task, and recovery periods, was normalised between 0 and 1 for each participant prior to analysis. This allows the results to be analysed more easily, but it means that the units of this measure are undefined. The two way t-test showed a significant difference in the HR, SCL, and EDR frequency with $p < 0.01$, and the ANOVA produced a significant difference between at least one of the baseline or task periods ($p < 0.05$); shown in the top section of Table 3.6. The HRV, as computed using the SDSD method, did not show a significant difference in any test. The change in HR during the 2-back task from the normal driving periods was significant ($p < 0.05$), and the change in SCL was significant for both the 1-back and 2-back tasks ($p < 0.01$). Figure 3.7 shows the mean values of the four physiological measures taken from the (a) ECG and (b) EDA signals, computed over the baseline, task, and recovery periods. The results reflect the statistical analysis and show that each of the HR, SCL, and

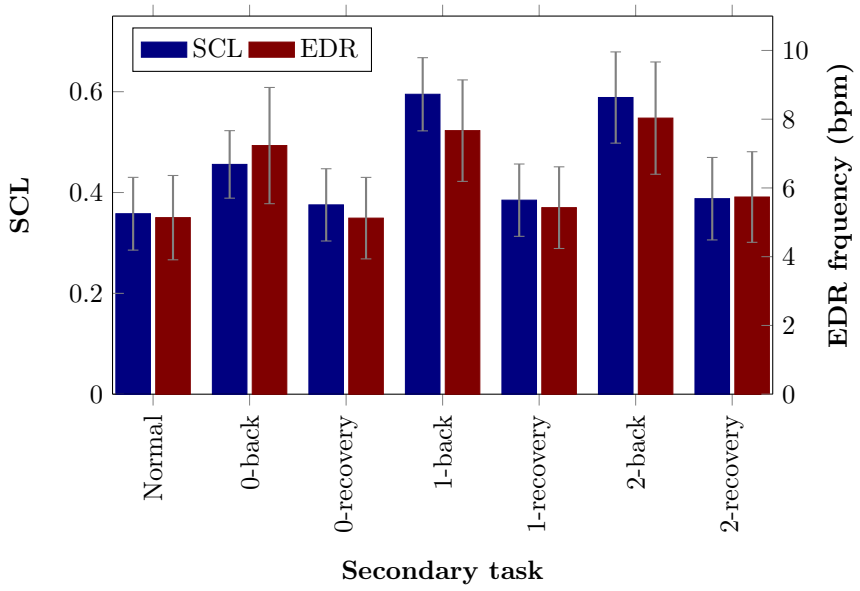
EDR frequency increased during the task periods, and decreased to the baseline levels during the recovery periods. The HRV was very similar throughout the trial.

In the lower section of Table 3.6 the results of the t-tests and ANOVA are shown for representative signals from the vehicle telemetry data. As well as the raw signal values, the Standard Deviation (STD) (STD) was computed for each signal over a one second sliding window. This produces a feature of the signals where sample values are equal to the STD of the twenty samples before and after the respective sample in the signal. Signals that were expected to have a close relationship to the driver workload were those related directly to the driving controls, such as the pedals and steering wheel. The analysis shows that the throttle position and STD of the steering wheel angle speed both have a close relationship to the driving period ($p < 0.05$ in both the two way t-test and ANOVA). The STD of the SWA however, was not as closely related to the driving period, which was unexpected. In fact, in the data the STD of the SWA decreased from the baseline during the 0-back task and increased during the 1- and 2-back tasks.

Signals with indirect relationships to the vehicle controls were expected to have weak relationships to the driving conditions. These had larger p-values in general than measures of the vehicle controls, such as with the STDs of both the suspension measurements and yaw rate. The raw values of the engine speed and target gear of the automatic gear box, however, had relationships more similar to those of the vehicle controls. Other signals that have no obvious link to the driver were of course expected to have large p-values, and for the majority this was the case. A small number, including adaptive cruise control cancel, engine coolant temperature, and others redacted from Table 3.6 as they had small p-values for the two way t-test and can only be explained by chance.



(a) ECG



(b) EDA

Figure 3.7: Mean values of (a) HR and HRV and (b) SCL and EDR frequency over all subjects for the different periods of the trial. Each recovery period is presented separately and error bars represent the standard error.

3.3.4 Ground truth for classification

Both the timings of tasks and the physiological data streams are used to produce ground truths. The task timings can be used as one ground truth to create a binary labelling to describe whether there was a secondary task being performed or not. Here, the label *normal* relates to driving under normal conditions and *distracted* signifies that a secondary task was being performed. The *distracted* label is then also split into three to signify which of the 0-, 1- or 2-back tasks was being performed, to produce a multi-label classification problem with four labels.

Each of the physiological data streams can be used to produce binary classification tasks, with a label of *normal* when the observations are close to those found during the baseline period, and *distracted* otherwise. Other levels can also be used to produce a multi-label classification problem. For example, increases of 5% or less can be assigned label A, of between 5% and 10% given label B, and of more than 10% label C.

3.3.5 Data release

The dataset is available for download via `www.dcs.warwick.ac.uk/dmd/` in a comma separated variable (csv) format, with samples in temporal order at 20Hz. Each of the class labels are provided for each sample. The physiological data is also available. This physiological data has timestamps, so that it can be associated with the CAN-bus data, but the sample rate remains at 256Hz.

Several features have been removed from the dataset to either protect intellectual property or because they are irrelevant to the problem. To avoid any human selection bias, correlation analysis with Mutual Information (MI) [160] is used; where features with a MI of zero have been removed.

The production and release of such a dataset may benefit both the driver monitoring and data mining communities. The data naturally has high autocorrelation, and several irrelevant and redundant signals, all of which affect the

Dataset	Sample size	(?)	Numeric features	Nominal features
Soybean (small)	47	(0)	0	35
Fertility	100	(0)	9	0
Promoters	106	(0)	0	58
Wine	178	(0)	13	0
Parkinsons	195	(0)	22	0
TR 23	204	(0)	5832	0
Soybean (big)	307	(41)	0	35
TR 12	313	(0)	5804	0
TR 21	336	(0)	7902	0
TR 11	414	(0)	6429	0
Congress	435	(203)	0	16
Arrhythmia	452	(384)	272	7
Musk 1	476	(0)	166	0
Metadata	528	(264)	20	0
Credit	690	(37)	6	9
Vehicle	846	(0)	18	0
Yeast	1484	(0)	8	0
Madelon	2000	(0)	500	0
Segmentation	2310	(0)	19	0
Splice	3191	(0)	0	60
Chess	3196	(0)	0	36
Optical digits	3823	(0)	0	64
Spambase	4601	(0)	57	0
OARD	869387	(639830)	242	0

Table 3.7: Details of datasets from the UCI and Tuned IT repositories used for evaluations in Chapter 5. The column **(?)** represents the number of samples containing a missing value.

performance of a classification system [76]. As well as this, some of the signals may be correlated with time, introducing biases. Overcoming these issues is not only essential to predicting driver behaviour, but they are also difficult problems for data mining in general. We provide a central dataset against which driver workload monitoring methods and temporal data mining techniques can be evaluated and compared.

3.4 Non-vehicular datasets

In addition to datasets taken from the automotive domain, the datasets listed in Table 3.7 that are available in the UCI[6] and Tuned IT[7] repositories are used. The majority of these datasets are from non-temporal domains and are used in Chapter 5 for estimating redundancy using the non-blocked permutation method. These were chosen because of their range in domains, sizes and features, as well as their use in previous feature selection literature [39, 58, 108, 170].

The OPPORTUNITY Activity Recognition Dataset (OARD) is from the human activity monitoring domain and collected from wearable, object and ambient sensors [129]. Participants performed tasks including preparing and drinking coffee while being monitored via sensors on the work surfaces, switches, objects, and on themselves. In total, 56 sensors were used to collect 242 measurements for four participants who each performed the tasks six times. Data was recorded at 33Hz to generate a dataset with 869387 samples. In this dataset there are also a large number of samples with missing values, which were caused by recording failures [129]. Almost all signals contained a missing value at some point during the recording, and so in our analysis they are treated as $NotANumber$ and retained. Finally, there are several target labels provided in the OARD, and in this thesis we use the Locomotion set which has four values: *Stand*, *Walk*, *Sit*, *Lie*.

3.5 Feature extraction

There are two approaches to feature extraction general, either for a domain expert to decide which features are expected to perform best (e.g. [94, 161]), or to use automatic feature learning (e.g. [17, 121]). Both approaches are susceptible to producing sub-optimal features. The domain expert may have biases toward certain kinds of feature, and automatically learning features assumes that the

[6] `http://archive.ics.uci.edu/ml`
[7] `http://tunedit.org/repo/Data/`

Type	Feature
Statistical	Min, Max, Mean, Standard deviation, Entropy, Fluctuation.
Structural	Raw value, First, Second and Third derivatives, First 5 and Max 5 DFT coefficient magnitudes, Max 5 DFT coefficient frequencies, Convexity, Gradient direction, Integral, and Absolute integral.

Table 3.8: List of statistical and structural features extracted from each signal from the RCD, CoventryDMD, WarwickDMD and OARD. Features are extracted over sliding temporal windows of sizes 0.5s, 1s, 2.5s and 5s.

signal data is unbiased – which is not the case with some telemetry data as discussed in Chapter 4. In either case, therefore, we believe feature selection should be performed after extraction and before using features in models.

In this thesis, each of the temporal datasets (the RCD, CoventryDMD, WarwickDMD, and OARD) undergoes the same feature extraction process, as described in Section 2.1.4. The feature extraction process involves the use of sliding windows over the signals, where each window is summarised by a statistical or structural feature [144, 145, 161]. Different information in the signal windows can be captured by different features from the signals. For instance, while STD will capture information about the signal's variability and spread, the mean will provide an average value over a period of time. Also, some signals change value faster than others, and so different window lengths should also be used. The gradient, for example, will capture value changes over longer periods if a larger window length is used. Therefore, we extract the 28 features listed in Table 3.8, each over sliding windows of sizes 0.5s, 1s, 2.5s and 5s. These features include STD, mean, minimum, maximum, as well as the gradient, and DFT components.

3.6 Summary

In this chapter the datasets used for evaluating learning approaches have been introduced. There are three automotive datasets, namely RCD (Section 3.1),

CoventryDMD (Section 3.2), and WarwickDMD (Section 3.3). A further telemetric dataset, the OARD (Section 3.4) [129] available from the UCI repository is also discussed briefly. From these temporal datasets, the same feature extraction stage is used, as presented in Section 3.5. The features extracted here are then used for evaluating feature selection approaches in Chapters 4 and 6. Finally, a set of datasets also downloaded from online repositories was introduced in Section 3.4. These are used where temporal artefacts are not considered, in Chapter 5.

CHAPTER 4

Temporal permutation feature relevancy

The feature selection stage of the data mining methodology described in Chapter 2 aims to select the features that perform best in a predictive model for a given classification task, such as road classification or driver workload estimation. Supervised feature selection, where a small number of features are chosen from a large set [58, 76], is however an example of Multiple Comparison Procedure (MCP). MCP is a major cause of input selection errors, over-fitting, and over-searching [65]. Permutation methods can be used to avoid these pathologies, but they require exchangeability of samples and are not directly applicable to temporally or spatially dependent data, or to data with high autocorrelation. To overcome this requirement, blocked-permutation methods are investigated and the resulting permutation distributions are used to normalise a Mutual Information (MI) ranking statistic. In this chapter, the validity of this blocked-permutation method is shown under assumptions of local dependency, and it is applied to supervised feature selection from vehicle telemetry data described in Chapter 3 and that has high temporal dependence and severe recording biases. Two new blocking strategies are proposed, namely: permuting the data in blocks of dynamic size and of sizes determined by the sample values. These are compared against permuting the data in blocks of static size, with and without applying a cyclic shift to the data. Finally, two novel permutation ranking statistics are compared against several existing methods, including Symmetrical Uncertainty (SU) and Hilbert-Schmidt Independence Criterion (HSIC), using rank comparison and classification performance, and are shown to successfully mitigate known biases in the data and selection process.

4.1 Introduction

Feature selection is used to produce a smaller subset of a larger feature set, containing only features that are highly correlated to class labels while being minimally redundant to one another [76]. However, selecting features from large feature sets is an example of the MCP, which is responsible for input selection errors, over-fitting, and over-searching [65]. Input selection errors occur when biases are introduced by the selection process, for example selection by MI is biased because it increases with feature dimensionality. This input selection bias is reduced in SU [160], and Song et al. [137] provide proofs to show that HSIC [42] is unbiased. SU and HSIC methods may not, however, mitigate the other forms of bias in the feature selection process. Over-fitting can occur when the data is a poor representation of the underlying distributions, causing some features to appear more relevant in the recorded data than is the case in general. Over-searching, possibly the most overlooked pathology, occurs when a large number of models or features are considered, and any high performance is a result of chance.

Feature selection pathologies all effectively occur when results from data analysis are assumed to be more significant than they really are. Significance methods such as t-tests are routinely used and assume a normal distribution [26], but this is often not the case. These methods generally assign higher significances to larger sample sizes, without consideration being taken for any bias in the data or selection process. Jensen and Cohen [65] suggest four solutions to these MCP pathologies: using a new data sample, cross-validation, Bonferroni adjustment, and randomisation tests. Sampling using new data may be costly, and the new data itself may contain the same biases as the original. Cross-validation and Bonferroni adjustment both assume that the data collection and analysis processes do not contain any biases, and will not cure over-fitting if these assumptions are incorrect. Randomisation tests, however, can be performed on existing data, while making no strong assumptions about

its distribution or the analysis process [24, 26, 40, 105]. Randomisation tests, therefore, are the most suited to detecting and avoiding the problems associated with the MCP.

The terms *randomisation test* and *permutation test* are used interchangeably and variably in the literature on hypothesis testing [24, 26, 40, 92, 105]. Ernst [26] uses the general term *"permutation methods"* to refer to such methods, as is the case in this thesis. Under any reasonable definition of permutation methods, the only strong requirement for their validity is that the samples are exchangeable. A sequence of samples, $[x_1, x_2, .., x_n]$, is exchangeable if any permutation of it has the same joint probability distribution [85],

$$Pr(x_1, x_2, \ldots, x_n) = Pr(x_1)Pr(x_2)\ldots Pr(x_n). \tag{4.1}$$

In essence, if any ordering of the samples is as likely as any other, the sequence is exchangeable [40, 105]. Indeed, any sequence of Independent and Identically Distributed (IID) observations is exchangeable, although this is not a requirement for exchangeability. For example, choosing elements from a finite set without replacement will produce an exchangeable sequence of non-IID observations [40].

Several authors regard permutation methods as the "gold standard" in hypothesis testing [24]. Bradley [11] and Fisher [32] both note that the conclusions of any significance test are only justified in that they would have also been arrived at by the permutation method. These observations, coupled with their minimal assumptions on the data, have meant that the permutation method has been widely used in applications ranging from agriculture to physics [40]. As well as this, the permutation method has also been deployed in correlation analysis and feature selection [4, 37, 118, 158, 166], in Decision Tree induction [38, 89], and in Random Forests [4, 51]. The application of permutation methods in temporal and spatial domains is limited, however, because exchangeability of the data cannot be guaranteed when autocorrelation is present in the dataset

[35, 36]. For instance if the data has inherent structure, as is the case when there is a high probability of subsequent values being similar, or a value is dependent upon its location in the sequence, samples in that data are not exchangeable. Therefore, different approaches are required to ensure the validity of the permutation method with temporal and spatial data.

This chapter addresses this issue by using a blocked-permutation approach [2, 75], where blocks are permuted rather than individual samples. We introduce the dynamic and single-value blocking strategies, both aimed at dealing fully with the issue of periodic data, and compare them against existing techniques. The dynamic blocking strategy avoids capturing periods in the data by splitting the data into blocks of random lengths for each permutation. The idea is similar to that of the cyclic shift introduced by Adolf et al. [2], but the position of all block boundaries are randomised. Dynamic blocking is conceptually more appealing and increases the number of possible permutations by a far greater amount than when using only the cyclic shift. A second strategy that we propose is single-value blocks, which is aimed purely at categorical or discrete data, and blocks contain only samples of the same value.

The remainder of this chapter is structured as follows. The operation of the permutation method is described in a general setting in Section 4.2 and the blocked permutation method is introduced in Section 4.3. Here, details of the different blocking strategies, including *dynamic* and *single-value*, are also given. Relevancy measures that can be used for feature selection with the permutation method are suggested in Section 4.4. In Section 4.5 we present empirical results of the blocked-permutation method for feature selection using the proposed ranking strategies. Finally, Section 4.7 draws conclusions on this work.

4.2 The permutation method

The permutation method is used to assign a significance to a test statistic, with respect to the null hypothesis [40]. For example, it can be used to assign a

significance to a correlation statistic, $f(X, Y)$, between two variables, X and Y, with respect to there being no relationship between the two variables. It operates by first computing the outcome of $f(X, Y)$ on the observed variables, referred to as the *observed test statistic*. Next, one of the variables is permuted in order to destroy any relationship between the variables. If Y' is a permutation of Y, the test statistic can then be recomputed as $f(X, Y')$. When $f(X, Y')$ is computed for all possible permutations of Y, its outcomes form the *permutation distribution*. It should be noted that permuting either variable provides the same results for the permutation method on two variables. In supervised feature selection from many features, for instance, it is computationally more efficient to permute the class labels rather than each feature individually when computing their relevancies.

The significance of the observed correlation is given by its location in the permutation distribution. The p-value of the observed correlation is the proportion of the permutation distribution which is at least as extreme as itself [26, 92, 108],

$$p = \frac{|\{Y' \in \Psi(Y) : f(X, Y') \geq f(X, Y)\}| + 1}{|\Psi(Y)| + 1}, \tag{4.2}$$

where $\Psi(Y)$ is the set of all possible permutations of Y, and $|\cdot|$ represents the cardinality of a set. Intuitively, the smaller the p-value the more significant the observed test statistic is with respect to the null hypothesis. A threshold can be placed on the p-value, below which the null hypothesis is rejected, for example $p < 0.05$ [51].

This is the *full permutation method*, where a complete permutation distribution is computed. However, a variable with n samples can have up to $n!$ permutations, meaning that it is often infeasible to perform the permutation method in full. In general, therefore, a Monte-Carlo permutation method is used, where a random subset of permutations are performed to estimate the p-value [26, 109]. The number of permutations, P, should be as large as possible and depends on the data. Typically, P is in the order of thousands.

Because the p-value is computed using a random sample of the permutation distribution, it is also appropriate here to place a two-sided confidence interval over it [109]. The α confidence interval of the p-value, p_α, would then be,

$$p_\alpha = p \pm z_\alpha \sqrt{\frac{p(1-p)}{P}}, \tag{4.3}$$

where z_α is the respective two sided standard-score.

4.3 Permutation methods for dependent data

As previously stated, the permutation method is not valid when the data samples are not exchangeable under the null hypothesis. This is typically the case with temporally and spatially dependent data, but also occurs when other autocorrelations are present. For example, in assessing the height of two populations, it may not make sense to swap samples between genders, because of the confounding effects of their known differences [40]. Disallowing these swaps is an example of restricted randomisation within a region [35], and is often referred to as within-block or within-population randomisation. Nichols and Holmes [105] and Zhou and Wang [174] both use the term "exchangeability-block" to refer to a temporal sequence of samples in which the samples are exchangeable. A restricted randomisation is then used, with samples being shuffled within exchangeability-blocks, but not between them.

Restricted randomisation within exchangeability-blocks will not perform well with some types of temporal or spatial data. If a variable changes value rarely with respect to time or space, for example, many of the permutations within a given block are likely to be equivalent. This is exacerbated when the data is also imbalanced, because several blocks are likely to contain data of the same value. Deng et al. [21] observed this phenomenon when performing partial permutations. In a partial permutation, some samples are not permuted, so the relationship between the variables is partially maintained. This means that test statistics on permuted data are likely to be more similar, shifting the permuta-

tion distribution towards the observed test statistic. For this reason, Ojala [107] states that each permutation must be sufficiently independent from the original data.

Ojala [107] uses a Markov randomisation procedure to preserve an underlying statistic of the data, such as the sum or variance of columns or rows, while ensuring that the permutations are sufficiently different. This type of permutation is used to determine whether or not a higher level data mining result is merely as a consequence of the simpler underlying statistic. It may be possible to adapt this to retain temporal or spatial dependencies, or autocorrelation structure, in the data to ensure exchangeability for the permutation method. However, this method and the Metropolis-Hastings algorithm used are not practical in all settings and is more computationally expensive than other randomisation techniques.

Another approach is to sub-sample the data in order to decrease the dependency of the samples. Consider a locally dependent sequence, X, where the dependency between two samples $x_i, x_{i+\tau} \in X$ and separated by τ, is $D(x_i, x_{i+\tau})$. This dependency can act both forwards and backwards and is always non-negative,

$$D(x_i, x_{i+\tau}) \geq 0, \ \forall \tau. \tag{4.4}$$

Also, because the dependencies are local, $D(\cdot)$ tends towards 0 as the distance between samples increases,

$$D(x_i, x_{i+\tau}) \to 0, \ \tau \to \infty. \tag{4.5}$$

If the dependency of the two samples is below a threshold, T_d, the samples are said to be independent, otherwise, they are said to be dependent,

$$x_i \leftrightarrow x_{i+\tau} = \begin{cases} \text{independent} & \text{if } D(x_i, x_{i+\tau}) < T_d \\ \text{dependent} & \text{otherwise.} \end{cases} \tag{4.6}$$

It follows that, there exists a τ where each sample, x_i, in a locally dependent sequence with at least τ samples, is independent to another and $D(x_i, x_{i+\tau}) < T_d$ for all positive values of T_d.

Using these observations, a locally dependent sequence can be sub-sampled so that all samples are separated by τ and are independent. After sub-sampling in this way, the sequence will be exchangeable and the permutation method can be applied to it. Using this method, however, ignores a large proportion of the data that may affect its statistics. If $\tau = 100$ and every 100^{th} sample is taken, for instance, this removes 99% of the data that would otherwise be included in correlation calculations between variables. This may lead to inaccuracies in both the observation and permutation statistics, which means any conclusions are suspect. Ideally, therefore, the sequence should be made exchangeable without removing any samples.

Lahiri [82] covers a range of block re-sampling methods for performing bootstrapping on dependent data. Bootstrapping is a related significance test which is used to assess the variability of the sampling process and determine asymptotically correct confidence intervals. Instead of permuting one variable to produce the permutation distribution, both are re-sampled with replacement to produce the bootstrap distribution. Despite their differences, however, some of the blocking techniques described can be adapted for use with permutation methods. For instance, re-sampling the data in blocks without replacement can be performed to produce a blocked-permutation of the data [159].

We define a block, $B_{i,l}$, of length l as a sequence of consecutive samples starting at i, $B_{i,l} = [x_i, x_{i+1}, \ldots, x_{i+l-1}]$, taken from observations of a variable, X. With temporal data, $B_{i,l}$ is a sequence of l temporally ordered samples observed at a fixed frequency over a period of time. In the blocked-permutation, samples within a block retain their ordering, while samples between blocks may not. Kirch [75] showed the validity of this blocked-permutation method for mean change-point analysis. The analysis is of the asymptotics of statistics based on partial and cumulative sums, generalising them to blocked data. This

specific analysis does not directly apply to MI and entropy based test statistics, however. Therefore, we provide an informal argument for the validity of the blocked permutation method for a general test statistic.

When blocks are permuted instead of samples, dependencies between samples within the same block are retained after a permutation. This means that it is not intra-block relationships between samples that are of interest, but inter-block relationships. For instance, although the dependency between the two samples in the same block may not be 0, their relative ordering remains the same throughout a blocked permutation and therefore can be ignored. With this observation, we define the dependency of two samples in a blocked sequence to be,

$$D_B(x_i, x_{i+\tau}) = \begin{cases} D(x_i, x_{i+\tau}) & \text{if } x_i \in B_{i,l}, x_{i+\tau} \in B_{j,l} : i + l \leqslant j \\ 0 & \text{otherwise,} \end{cases} \tag{4.7}$$

where $B_{i,l}$ and $B_{j,l}$ are blocks of samples that do not overlap. Because samples have a dependence only with those from different blocks, the dependence of a sample on others is determined by its distance to the block boundary. This is shown in Figure 4.1, where the plot represents the dependence of each sample with the nearest bordering sample of the neighbouring block. It highlights adjacent samples each side of a block boundary, which have the highest dependency overall. Samples that are furthest away from a block boundary, in the middle of a block, have least dependence on others. In fact, for a block $B_{i,l}$, the sample with least dependency on others is $x_{i+l/2}$ and the samples with most dependency on others are x_i and x_{i+l-1}.

With a sufficiently large block length, the samples in the middle of blocks can be made independent to others. For instance, as their distance to the block boundary increases, their dependence with the sample on the neighbouring block boundary decreases. If the block length is sufficiently large, this dependency can be made less than T_d, meaning that the sample can be considered as independent

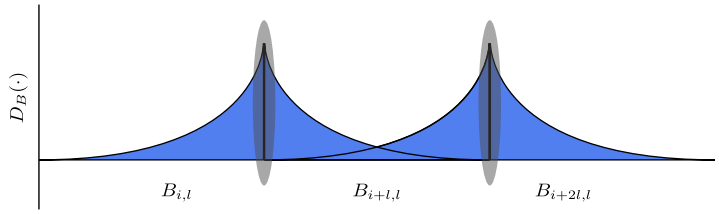


Figure 4.1: Dependency of three small consecutive blocks over time. The block boundaries have the highest level of dependency and are highlighted; the centre of the blocks have the least dependency.

to all others in the blocked sequence. As l increases further, there will be more samples with dependencies on others that are lower than this threshold, and so the number of these independent samples increases. Furthermore for all values of l that produce at least one independent sample, the number of samples that are dependent on another is the same.

Next, we define the dependency between two consecutive non-overlapping blocks as the ratio of their samples that are dependent with another and those that are not,

$$D_B(B_{i,l}, B_{i+l,l}) = \frac{|x_i \leftrightarrow x_j = dependent|}{|x_i \leftrightarrow x_j = independent|}, \forall\ x_i \in B_{i,l}, x_j \in B_{i+l,l}. \quad (4.8)$$

As l increases, the number of samples that are dependent on another in the two blocks becomes negligible when compared to the number of samples that are independent with all others. This is shown in Figure 4.2, where the block size has been increased to produce a section of independent samples, highlighted by the shaded area. As the block length and number of samples in this area increases, the blocks can be considered as more independent to each other.

With very large block lengths, such as is possible with infinite sequences, this ratio of dependent and independent samples has a limit of 0. If the sequence length is finite however, a block length close to that of the sequence would produce permutations that are similar to the observations. A trade-off is then found between having small block sizes to ensure enough different per-

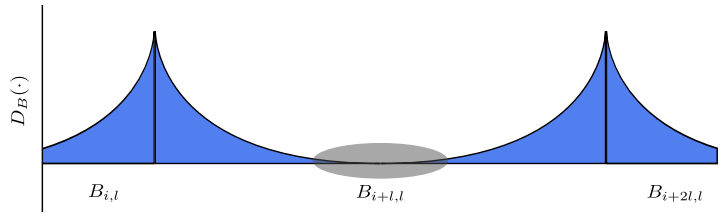


Figure 4.2: Dependency of three large consecutive blocks over time. The blocks produce an independent section of samples highlighted in the middle, introducing independence into the blocked sequence.

mutations are possible and minimising this ratio. In situations with extremely high autocorrelations or small sample sizes, where there are no suitable block lengths, approaches such as a cyclic shift may have to be used [2].

In the remainder of this section, three blocking strategies are discussed. These strategies include *static* blocks, as used by Kirch [75], and two other blocking strategies which, to our knowledge, have not previously been used with the blocked-permutation method, namely *dynamic* blocks and *single-value* blocks. Also, with the static and dynamic strategies we investigate the random cyclic shift as introduced by Adolf et al. [2]. The best strategy to be used may be dependent on properties of the data being processed, including its sample size, autocorrelation, and periodicity. Their suitability to vehicle telemetry data, namely the Road Classification Dataset (RCD), Coventry-JLR Driver Monitoring Dataset (CoventryDMD), and Warwick-JLR Driver Monitoring Dataset (WarwickDMD) is discussed in Section 4.6.

Static blocks

The *static* blocking method is derived from the non-overlapping blocking outlined by Lahiri [82]. This method is used by Kirch [75] when using the permutation method for change-point detection in signal analysis. It splits the data into k blocks of equal length, l, where $kl = n$. A blocked sequence is therefore,

$$[B_{0,l}, B_{l,l}, B_{2l,l}, \ldots, B_{(k-1)l,l}]. \tag{4.9}$$

The ordering of samples within each of the blocks is retained while the blocks themselves are permuted. If the block length does not divide exactly into the sequence length, the samples left out of the blocking can be either ignored, or treated as an extra block. In our work they are treated as an extra block.

Dynamic blocks

If the data being permuted is periodic, static block sizes may cause issues. This is because, when the block size is equal to the length of the periodic pattern in the data, each block will contain the same information. One simple method of avoiding this is to use dynamic blocks, in which the length of each block is randomised. The idea is that any periodic behaviour of the sequence is destroyed, generating a smoothed version of the permutation distribution.

Dynamic block sizes are randomised for each permutation iteration and for each block. That is to say, the blocked sequence becomes

$$[B_{0,l_1}, B_{l_1,l_2}, B_{l_1+l_2,l_3}, \ldots, B_{l_1+l_2+\ldots+l_{k-1},l_k}], \tag{4.10}$$

where each l_i is a uniform random number in the range $l_{min} : l_{max}$. An important consideration with the dynamic blocking method is that both l_{min} and l_{max} must be suitable block lengths for the data. For instance, if l_{min} does not introduce sufficient independence between blocks, or if l_{max} is too close to the sequence length, a smaller range should be chosen.

Single-value blocks

It is possible to perform Fourier analysis and choose blocks sizes informed by the periodic behaviour of the sequence. Ptitsyn et al. [115] use the permutation method in the detection of periodicity in short time series data. The permutations performed destroy any periodic patterns, by only swapping samples if they belong to a different phase of the period. A similar method could be adopted for defining block sizes, where the start and end of a block must belong to different

phases.

This method may not be applicable in all cases, especially when performing supervised feature selection with many features, where each feature would have to be analysed and permuted separately. However, if we consider the permutation of only the class labels, single-value blocks may be applicable for nominal data. Here, each block contains only samples of the same value, and a new block is defined whenever there is a change in value. However, if the change in value is very rare then block sizes will be large and permutations may again not be fully independent from one another. A maximum block size can therefore be introduced, using either the static or dynamic strategy outlined above.

Cyclic shift

Adolf et al. [2] apply a static blocked permutation method to short autocorrelated time series data for multivariate analysis. With this data the block length required for exchangeability is too large and the number of sufficiently different permutations is too small. To increase the number of distinct permutations, they apply a random cyclic shift in the data before each permutation. For instance, before the data is blocked and permuted, one of the variables is shifted by a random number of samples. Any sample that is shifted beyond the data length is moved to the start of the sequence. The result is that the position of each sample in the data is increased by the shift amount modulo the data length. Here, we apply this cyclic shift to both the static and dynamic blocking strategies.

4.4 Feature ranking methods

In supervised feature selection it is common to rank features with their relevance to the class label. One such measure of correlation for discrete data is MI,

$$MI(X,Y) = \sum_{v_1 \in vals(X)} \sum_{v_2 \in vals(Y)} p(v_1, v_2) \log_2 \frac{p(v_1, v_2)}{p(v_1)p(v_2)}, \tag{4.11}$$

where $vals(X)$ is the set of values X can take, $p(v_1, v_2)$ is the join probability distribution of X and Y, and $p(v)$ is the marginal probability distribution. However, MI tends to favour features with many values over those which have few, introducing an input selection bias [65]. This bias is reduced in SU [160] by dividing MI by the mean entropy of the two variables,

$$SU(X,Y) = 2\frac{MI(X,Y)}{H(X)+H(Y)}, \tag{4.12}$$

where,

$$H(X) = \sum_{v \in vals(X)} p(v)\log_2 p(v). \tag{4.13}$$

SU still prefers features with many values in some conditions, however, so the bias is not fully mitigated.

Song et al. [137] use HSIC to evaluate the relationship of features with the target labels, and offer proofs to show that it is unbiased. HSIC is defined as,

$$HSIC(X,Y) = \frac{1}{n(n-3)}\left[\mathrm{tr}(\mathbf{KL}) + \frac{\mathbf{1^T K 1 1^T L 1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1^T K L 1})\right] \tag{4.14}$$

where $\mathbf{1}$ is a vector of ones, and $tr(\cdot)$ is the matrix trace. $\mathbf{K}$ and $\mathbf{L}$ are kernel matrices with entries of kernel functions defined on the data and labels respectively and their diagonal values set to zero. For the data kernel matrix, $\mathbf{K}$, we use the Radial Basis Function (RBF), with entries,

$$\mathbf{K}_{ij} = k(x_i, x_j) = \begin{cases} \exp(-\frac{||x_i - x_j||^2}{\widetilde{X}}), & \text{if } i \neq j. \\ 0, & \text{otherwise,} \end{cases} \tag{4.15}$$

where each feature is standardised to have a mean of zero and a variance of one, and $\widetilde{X}$ is its median value. For the label kernel matrix, $\mathbf{L}$, we use a binary

kernel that weights classes based on their sample size,

$$\mathbf{L}_{ij} = l(y_i, y_j) = \begin{cases} n_-^{-1} n_+^{-1} y_i y_j, & \text{if } i \neq j. \\ 0, & \text{otherwise,} \end{cases} \tag{4.16}$$

where n_- and n_+ are the number of samples of negative and positive classes respectively, and $y_i, y_j \in \pm 1$ are the label values.

Song et al. [137] use forward and backward selection using the HSIC measure, to maximise overall relevancy of the selected features. In this chapter we simply rank the features by their individual HSIC scores instead, both to provide a direct comparison to other ranking methods and to reduce computational expense. Also, as $\mathbf{K}$ and $\mathbf{L}$ are symmetric positive definite matrices, a sparse approximation is produced using the incomplete Cholesky decomposition. Bach and Jordan [7] provide an algorithm for computing the matrices $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{K} \approx \mathbf{A}\mathbf{A}^\mathbf{T}$ and $\mathbf{L} \approx \mathbf{B}\mathbf{B}^\mathbf{T}$. Using this approach, we compute 100 columns of $\mathbf{A}$ and $\mathbf{B}$, which are then used to approximate HSIC in reasonable time.

One assumption of HSIC is that samples are IID, which is not the case with temporal data. To overcome this requirement, Zhang et al. [171] propose blocked variants of HSIC to handle non-IID data. With data used in this thesis, however, we found this approach to provide worse performance than HSIC without blocking.

Another approach to avoiding bias in feature selection is the permutation method. There are several methods of ranking features using MI and the permutation method, which can be separated into three groups. First, the p-value can be used directly as a ranking metric [65]. Second, the feature can be rejected if the p-value is below a threshold, and accepted otherwise [37]. Here, the ranking of the accepted features is provided by the MI value, and rejected features are not selected at all. Finally, the observed MI value can be normalised by the p-value, or some other metric defining where it lies in the permutation distribution [118, 156].

Wang et al. [156] do not compute a p-value at all. Instead they assume that the permutation distribution is normally distributed and compute the standard score of the observed MI value [118],

$$Z_{MI}(X,Y) = \frac{MI(X,Y) - \mu}{\sigma} \tag{4.17}$$

where μ and σ represent the mean and Standard Deviation (STD)s of the permutation distribution. Radivojac et al. [118] first separate features into two groups, strong and weak. Features with p-values below a threshold are said to be strong, and their p-values are unreliable for ranking. Instead they are ranked as in Equation 4.17. If the p-value is above the threshold, the features are considered to be weak and their p-values can reliably be used directly in the ranking. All weak features are given rankings below those of strong features.

In both these cases, the permutation distribution is being parametrised as if it were normal, which is often not the case. When the permutation distribution is not normal we propose to use the mean ratio between the permutation distribution and $MI(X,Y)$,

$$MR_{MI}(X,Y) = \frac{1}{|\Psi(Y)|} \sum_{Y' \in \Psi(Y)} \frac{MI(X,Y)}{MI(X,Y')}. \tag{4.18}$$

However this may generate very large values, and if any permutation of the data has a MI of 0, the function is undefined. Because of this, we also propose to use the mean difference between the permutation distribution and $MI(X,Y)$, normalised by $MI(X,Y)$,

$$MD_{MI}(X,Y) = \frac{1}{|\Psi(Y)|} \sum_{Y' \in \Psi(Y)} \frac{MI(X,Y) - MI(X,Y')}{MI(X,Y)}, \tag{4.19}$$

where if $MI(X,Y)$ is 0 then $\mathrm{MD_{MI}(X,Y)} = 0$. As in Radivojac et al. [118], these metrics can be used for strong features when a p-value is below a threshold, with the p-value being used to rank weak features below all strong features.

As with computing the p-value, it is infeasible to generate every permutation

Type	Road Classification	Driver Monitoring
Relevant	Longitudinal and Lateral accelerations, Gear position, SWA, SWA speed, Suspension measurements (for each wheel), Vehicle speed, Vehicle velocity.	Brake pressure, SWA, SWA speed, Vehicle speed, Engine speed, Pedal and Throttle positions, Absolute throttle position, Yaw rate.
Irrelevant	Ambient temperature, Brake pressure, GPS satellites, Velocity quality, Wiper status, Indicator status.	Ambient temperature, Longitudinal and Lateral accelerations, Gear position, Gear selected, Suspension measurements (for each wheel), Wiper status, Indicator status.
Bias	Latitude, Longitude, Time.	Latitude, Longitude, Time, Hour, Minute and Second counters, Minutes.

Table 4.1: List of signals taken from the RCD, the CoventryDMD, and the WarwickDMD, divided into three types. Signals for CoventryDMD and WarwickDMD are the same, except that there are fewer minute counters in WarwickDMD. *Relevant* signals are expected to have good performance for unseen data. *Irrelevant* signals are expected to have little correlation to the class labels and be of no use in solving the problem. *Bias* signals are expected to appear to have good performance in training data, while being of little use with new or unseen data.

of Y, and so in practice $\Psi(Y)$ is randomly sampled in generating these feature scores. Also, other correlation measures such as SU can be used in place of MI [160].

4.5 Experimental setup

The permutation methods are evaluated using subsets of the CoventryDMD, RCD, and WarwickDMD (described in Chapter 3). Because of their size and number of features, the subsets of signals listed in Table 4.1 are used. The signals for each dataset are separated into three kinds. The first kind is intuitively expected to be *relevant* to the class labels and be useful for the problem. The second kind are signals which are intuitively expected to be *irrelevant* to the class labels, and be of no use in solving the problem. The third kind of signals are those which contain *biases* and are expected to be highly correlated to the

data, but are of no use with new or unseen data. Furthermore, many of these listed signals and features extracted from them are redundant to each other. For example, suspension measurements are recorded for all wheels (front-right, front-left, rear-right and rear-left), which are highly correlated to one another. Choosing two features that are redundant can cause issues for models built on the data, including lower performance and an increase in complexity [46, 76], as discussed in Section 2.1.7.

The permutation method is performed with 5 blocking strategies; namely single-value, static, static with cyclic shift, dynamic, and dynamic with cyclic shift. In all blocking strategies, 5000 permutations were used for block lengths of $l = 1, 10, 20, 30, \ldots, 10000$, where $l = 1$ is equivalent to the non-blocked permutation method. The block sizes in the dynamic strategies are uniform random numbers in the range $l \pm l \times 0.25$. For single-value blocks, l is used as a maximum static block size when there are too many consecutive samples of the same value. Using this range of block lengths will highlight the point at which the block lengths introduce independence and become exchangeable under the null hypothesis.

Because these block sizes do not cover the full range for the datasets, experiments are also performed on further sub-samples of the RCD, the CoventryDMD, and the WarwickDMD. They are sub-sampled again by factors of 10 and 100 to provide data at 0.2Hz and 0.02Hz for the RCD and WarwickDMD dataset and 0.1Hz and 0.01Hz for the CoventryDMD dataset. This provides insight into the behaviour of the permutation method as the block lengths reach the sample size of the dataset. It is expected that the cyclic shift is necessary for good results in this case.

Once a suitable block length is detected, the MI, SU, HSIC, Z_{MI}, MR_{MI} and MD_{MI} ranking strategies are evaluated. These ranking methods are compared visually and used in a classification process to inspect their relative performance. For each ranking in this evaluation, the highest ranked feature extracted from each signal is used. The Random Forest and Multilayer Perceptron algo-

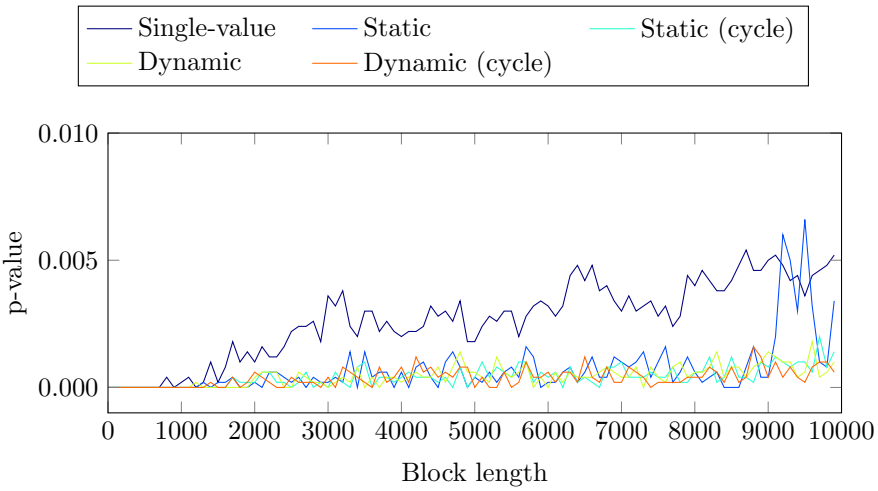
rithms, as implemented in the WEKA machine learning library [160], are used for comparison of classification performance. In order to provide an estimate of performance, 20 train-test cycles are performed in a temporal evaluation structure described in Section 2.2.1. The training dataset is used for feature selection and learning, and the testing dataset is used in estimating the performance of the model and selected features. In each train-test iteration, 40% of each journey is used as training data and the remaining 60% is used as testing data. The training datasets are made up from a section of contiguous samples starting at proportionally the same point in each journey. In the first iteration, the training data is made up from the first 40% of each journey, while in the second iteration samples of between 5% to 45% are used. The training data is shifted by 5% for each iteration, and in the final iteration the training data is made up from the last 5% and the first 35% of each journey. The predictions from all iterations are then combined to produce an overall Area Under the Receiver Operator Characteristic Curve (AUC).

4.6 Results

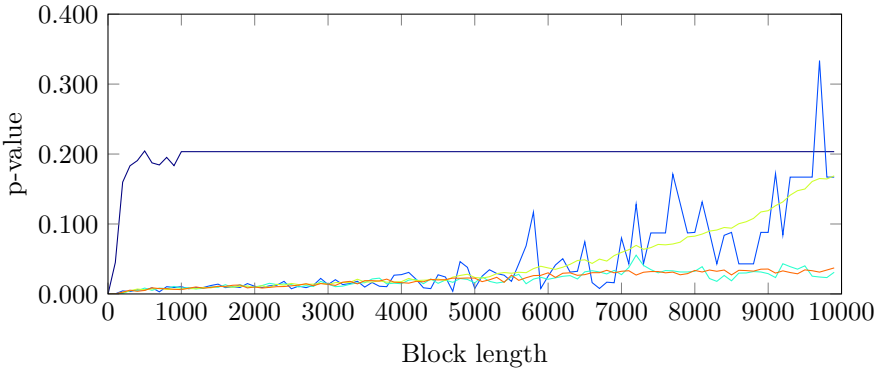
In this section we first present results for the p-value computed using the blocked-permutation method with several block lengths. Second, we show that there are biases in the selection of features by MI and SU, namely data collection bias and selection bias. We then provide evidence to show that these biases are reduced by HSIC, and removed by two of the five permutation ranking strategies considered.

4.6.1 Blocked-permutation test

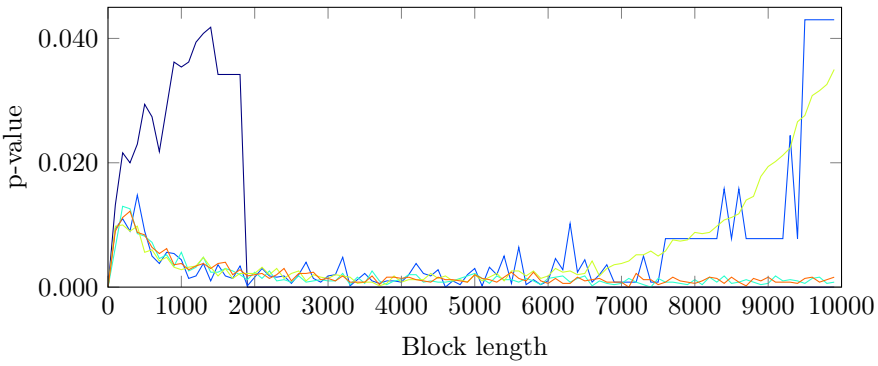
The p-values produced by each of the blocking strategies are shown in Figure 4.3 for the three datasets over block lengths of $l = 1, 100, 200, 300, \ldots, 10000$. Block sizes in increments smaller than this caused illegible plots due to the low magnitudes of the p-values and overlapping lines. For the RCD dataset, the front-right



(a) RCD: front-right suspension height



(b) CoventryDMD: minutes



(c) WarwickDMD: steering wheel angle

Figure 4.3: p-value against block size for (a) the front-right suspension height in the RCD, (b) the minutes signal in the CoventryDMD, and (c) the SWA speed in the WarwickDMD, using the static, dynamic and single-value blocking strategies. Plateaus in the p-value with a block size of about 3000 for the RCD, and 1500 for the CoventryDMD and the WarwickDMD.

suspension height is displayed as an illustrative signal, with its highest performing feature by MI. This signal was intuitively expected to perform well for this classification task, as larger roads with multiple lanes tend to be smoother than others, meaning the suspension height is less variable. In the CoventryDMD dataset the minutes signal (the number of minutes past the current hour) is shown, again with its highest performing feature by MI. This feature was expected to perform badly in general as it should not be a good indicator of driver workload, but had a very high MI value in our data. Finally, the SWA speed signal is presented for the WarwickDMD, which was again expected to perform well for this driver workload classification task. Similar trends were seen across other features and ranking statistics investigated for each of the datasets, but these are omitted for space reasons.

In all cases, the p-value for a block size of $l = 1$ was zero, or close to zero. For the RCD and CoventryDMD the p-values increased with block size, up to where the p-values reached a plateau (considered here as a region where the p-value did not change significantly in a trend over different block lengths). It should be noted that the range of p-values for the suspension signal in the RCD is very small, meaning that the changes in p-value observed in the plot in fact are small. With the WarwickDMD the plateau was still present, but a peak was observed with block sizes of less than 1000. Importantly, each of the signals in the same dataset reached the plateau (seen primarily with the dynamic blocking strategies) at around the same block size, namely around $l = 3000$ for the RCD, 1500 for the CoventryDMD, and 1500 for the WarwickDMD. This indicates that this block size produced independent and exchangeable blocks. At this point, it is clear that the p-value for the suspension height was much lower than that of the minutes signal. This shows that by the permutation method performed, the correlation of the minutes signal was much less significant than that of the suspension height or steering wheel angle signals, as is expected intuitively.

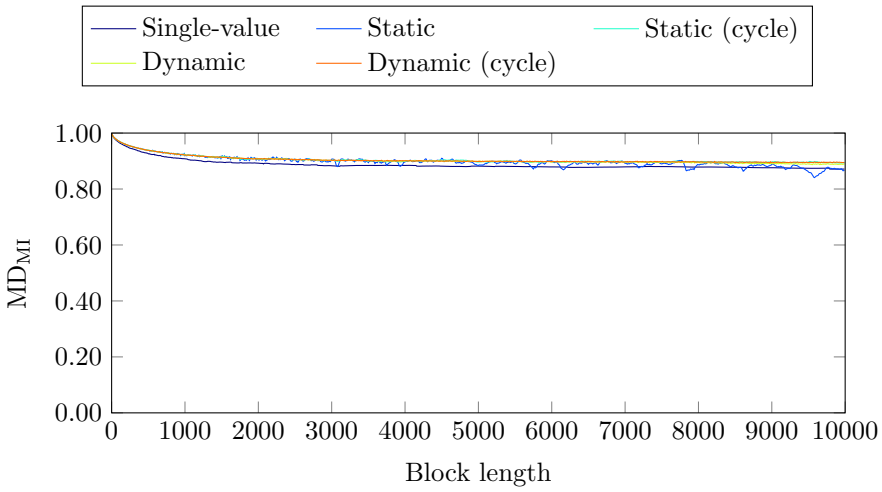
With the static and dynamic strategies without cyclic shifts, we found that the p-values increased with block sizes over 4000 in the CoventryDMD and 6000

in the WarwickDMD. This was likely because the block size was too close to the sample size, and the permutations were similar to one another [2, 107]. When a cyclic shift was applied along with these blocking strategies, the plateau was extended to all the block lengths investigated. The cyclic shift also reduced the high variability in p-values over different block sizes that was observed with the static strategy. As expected, the dynamic blocking strategy both with and without a cyclic shift was far less variable than either static blocking strategy, and appears to be a smoothed version of static blocking.

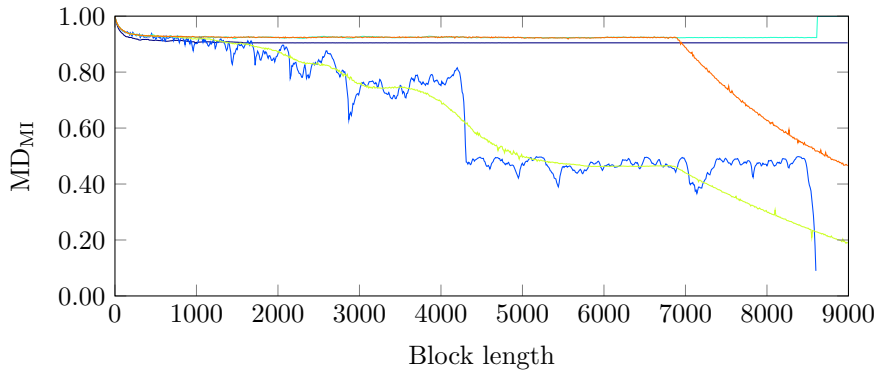
The single-value blocking strategy, which employs static blocking when there are too many consecutive samples with the same value, showed a combination of these results. The CoventryDMD dataset had a longest sequence of the same label of around 1000 samples, meaning that all block sizes larger than this produced the same p-value. With block sizes smaller than this for the WarwickDMD and all block sizes for the RCD, we found that the p-value was highly variable. The p-values produced with this blocking strategy also tended to be larger than with the other strategies, possibly because there are fewer possible permutations and they are too similar to the observed features.

We assert that a good block size for the blocked-permutation method should be chosen from those that are on the plateau. Therefore, a block size of at least 3000 for the RCD and 1500 for the CoventryDMD and WarwickDMD should be used. To confirm this, we further sub-sampled the datasets temporally by factors of 10 and 100. In the sub-sampled datasets, we assumed that the samples are less dependent on one another, and that the plateau would be produced using smaller block sizes.

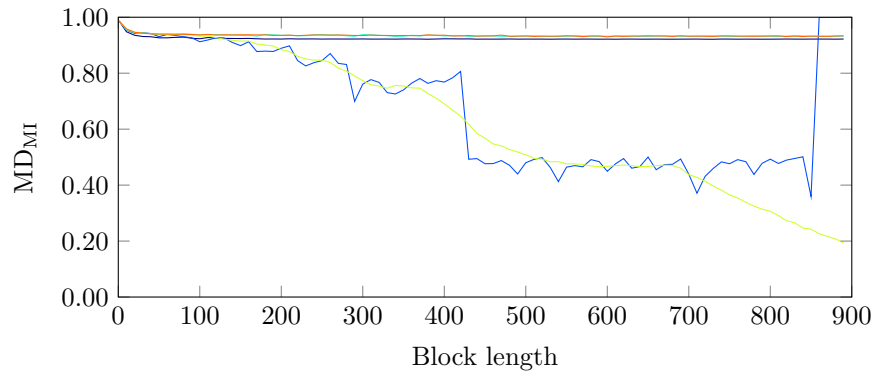
Figures 4.4, 4.5, and 4.6 show the MD_{MI} scores against block sizes of $1, 10, 20, \ldots, 10000$ for the RCD, CoventryDMD, and WarwickDMD respectfully. In each of these figures, plot (a) shows the MD_{MI} scores for the full datasets, and plots (b) and (c) show them for the two sub-sampled versions. As before, these results were the same for other signals in the datasets and ranking strategies. We can immediately notice that the plots for the two sub-sampled



(a) RCD 2Hz

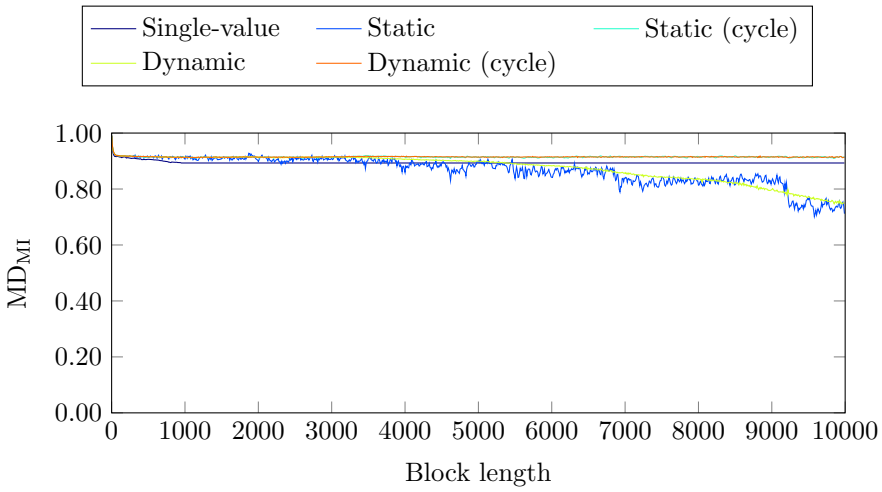


(b) RCD 0.2Hz

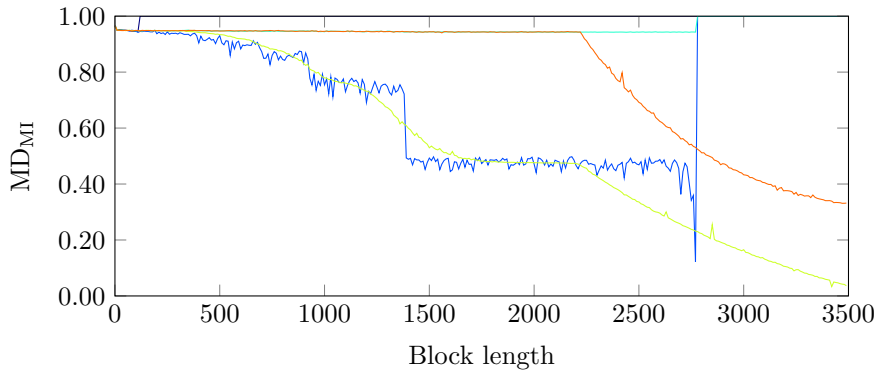


(c) RCD 0.02Hz

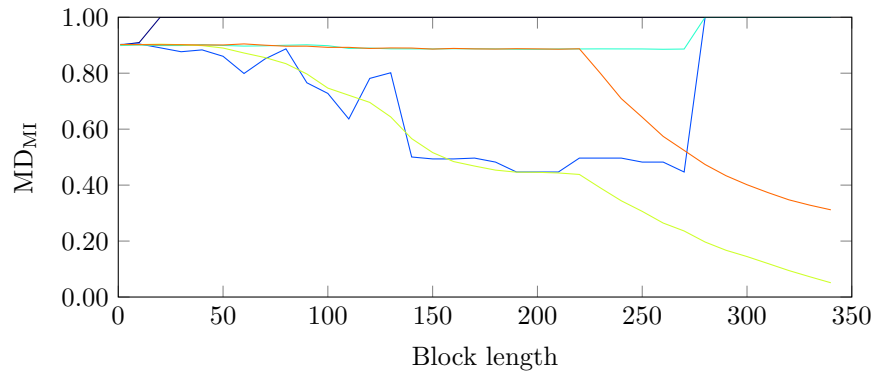
Figure 4.4: MD_{MI} score against block size for the front-right suspension height signal in the RCD with two further sub-samplings by factors of 10 and 100. The plateau of the MD_{MI} statistic was around the same block size as the p-values in Figure 4.3, with the plateaus for the sub-sampled data at 0.1 and 0.01 of the original block sizes.



(a) CoventryDMD 1Hz

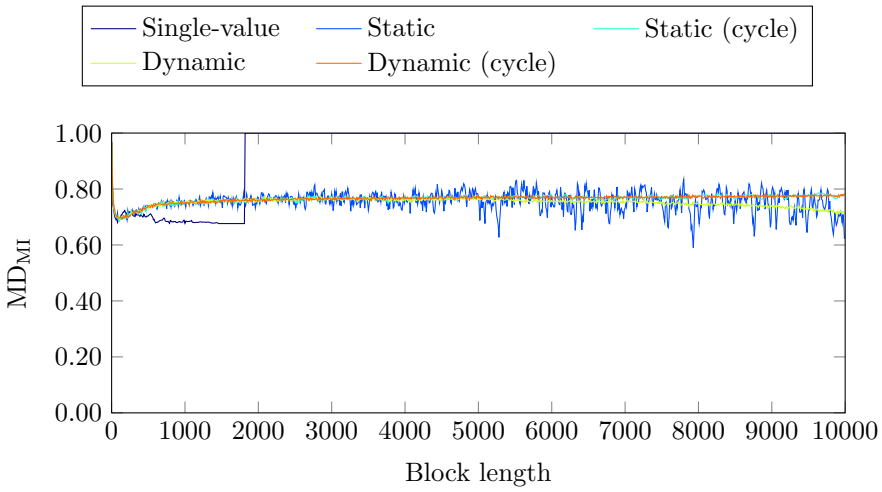


(b) CoventryDMD 0.1Hz

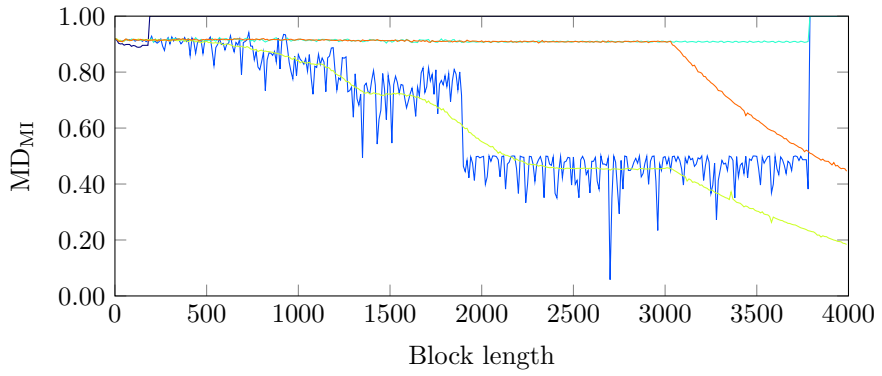


(c) CoventryDMD 0.01Hz

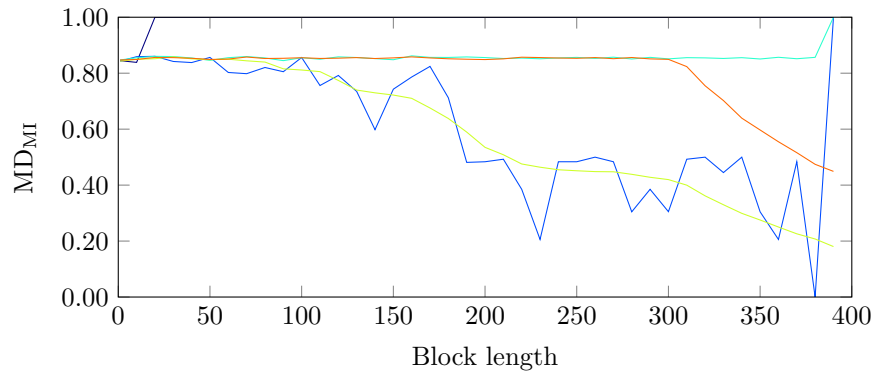
Figure 4.5: MD_{MI} score against block size for the minutes signal in the CoventryDMD with two further sub-samplings by factors of 10 and 100. The plateau of the MD_{MI} statistic was around the same block size as the p-values in Figure 4.3, with the plateaus for the sub-sampled data at 0.1 and 0.01 of the original block sizes.



(a) WarwickDMD 2Hz



(b) WarwickDMD 0.2Hz



(c) WarwickDMD 0.02Hz

Figure 4.6: MD_{MI} score against block size for the SWA speed signal in the WarwickDMD with two further sub-samplings by factors of 10 and 100. The plateau of the MD_{MI} statistic was around the same block size as the p-values in Figure 4.3, with the plateaus for the sub-sampled data at 0.1 and 0.01 of the original block sizes.

versions were of a similar form, but with the axes different by a factor of 10. The plateaus for the MD_{MI} statistic are found at similar block sizes to those found with the p-value. Furthermore, the plateaus with no cyclic shift for the sub-sampled data were found at a factor of 0.1 and 0.01 of the block size of the original datasets respectively. This signifies that the independence introduced by sub-sampling the data was also introduced with the blocked-permutation method.

When a cyclic shift was applied along with the static and dynamic blocking strategies, the plateau was again extended to all the block lengths investigated. This shows that when the data length was too small to allow a suitable block size, the cyclic shift was required. Where the data length was large enough compared to block lengths being used, the dynamic strategy was unaffected by introducing a cyclic shift.

In summary, the single-value and static blocking strategies are not suitable in this domain as they either produce bad p-values or vary too much over different block sizes. The dynamic strategy was more stable over different block sizes, but as l approaches the number of samples the number of distinct values in the permutation distribution tended to one and the p-values increased. This problem was not observed in the strategies with a cyclic shift, where the plateaus were extended across all block lengths. Out of the static and dynamic strategies with the cyclic shift, however, the dynamic strategy again varied less for different block lengths. As a result, the dynamic strategy with a cyclic shift is determined to be the most suitable of the five in this domain. For the remainder of this chapter, we use the dynamic strategy with a cyclic shift and a block length of $l = 3000$ for the RCD, and $l = 1500$ for the CoventryDMD and the WarwickDMD.

4.6.2 Feature rankings

The relationships between the ranking strategies outlined in Section 4.4, and ranking by MI are shown in Figures 4.7 (for the RCD), 4.8 (for the CoventryDMD) and 4.9 (for the WarwickDMD). Signals are ordered on the y-axis by MI (with

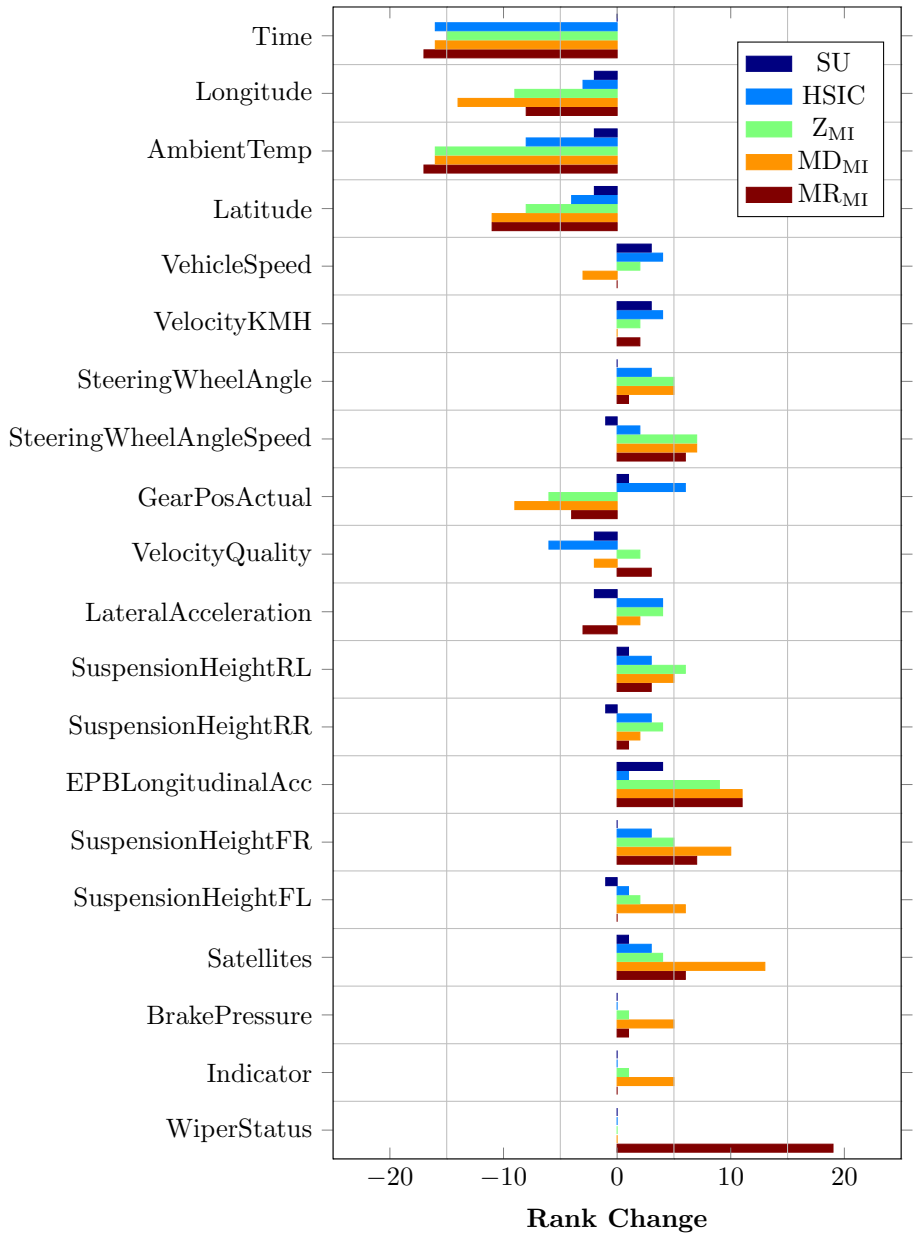


Figure 4.7: The relationship between the ranking strategies and ranking by MI for the RCD. Signals are ordered on the y-axis by MI, and each bar shows the difference between the ranks by MI and the ranks by SU, HSIC, Z_{MI}, MR_{MI} and MD_{MI}. As expected, the biased signals, such as time, longitude, and latitude, were ranked lower by the permutation statistics than by MI and SU. The bias signals were also ranked lower by HSIC, but the ranks of some signals, including longitude and latitude, were not changed significantly.

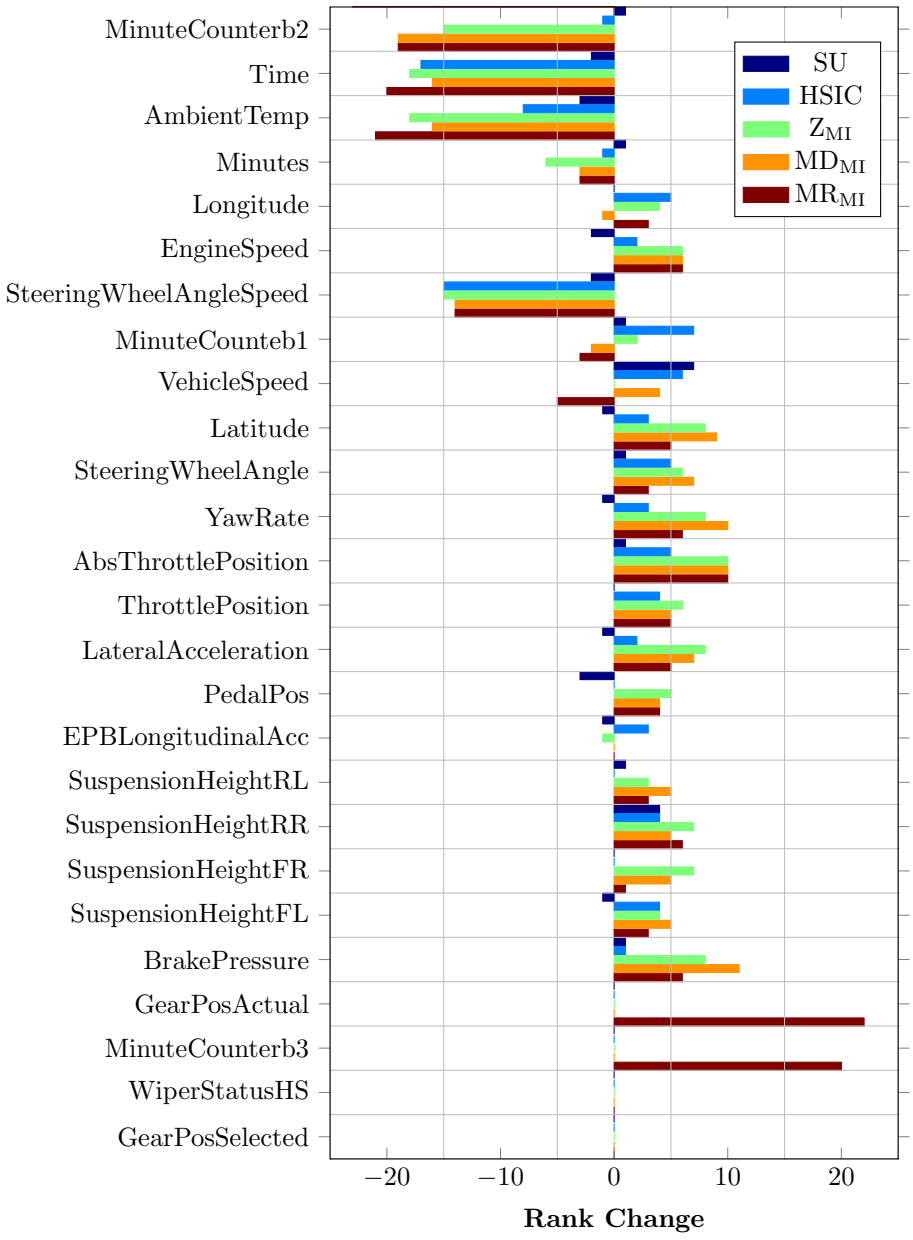


Figure 4.8: The relationship between the ranking strategies and ranking by MI for the CoventryDMD. Signals are ordered on the y-axis by MI, and each bar shows the difference between the ranks by MI and the ranks by SU, HSIC, Z_{MI}, MR_{MI} and MD_{MI}. As expected, the biased signals, such as time, longitude, and latitude, were ranked lower by the permutation statistics than by MI and SU. The bias signals were also ranked lower by HSIC, but the ranks of some signals, including longitude and latitude, were not changed significantly.

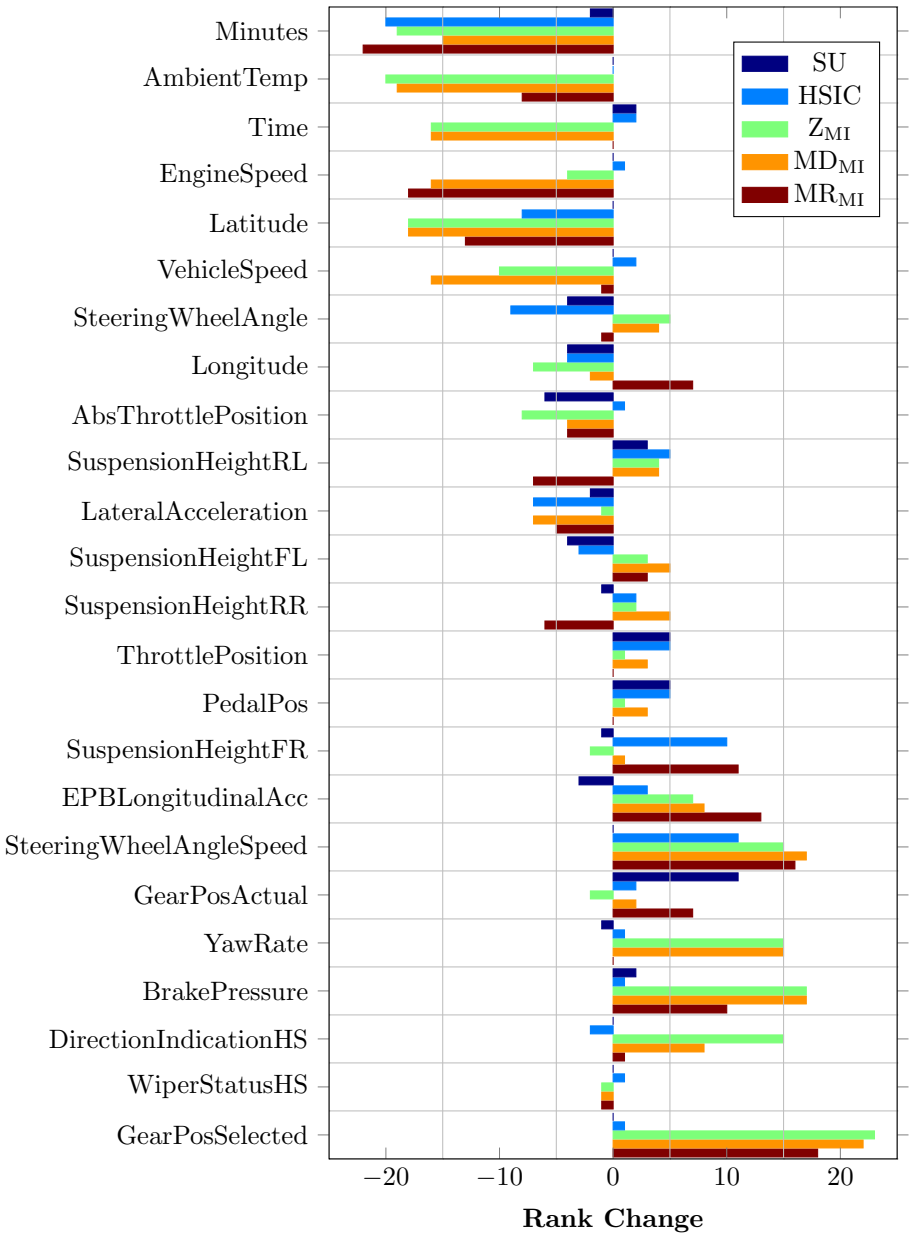


Figure 4.9: The relationship between the ranking strategies and ranking by MI for the WarwickDMD. Signals are ordered on the y-axis by MI, and each bar shows the difference between the ranks by MI and the ranks by SU, HSIC, Z_{MI}, MR_{MI} and MD_{MI}. As expected, the biased signals, such as time, longitude, and latitude, were ranked lower by the permutation statistics than by MI and SU. The bias signals were also ranked lower by HSIC, but the ranks of some signals, including longitude and latitude, were not changed significantly.

Signal	**MI**	**SU**	**HSIC**	$\mathbf{Z_{MI}}$	$\mathbf{MD_{MI}}$	$\mathbf{MR_{MI}}$
Time	1	1	17	16	17	18
Longitude	2	4	5	11	16	10
Ambient Temperature	3	5	11	19	19	20
Latitude	4	6	8	12	15	15
Vehicle Speed	5	2	1	3	8	5
SWA	7	7	4	2	2	6
Susp RearLeft	12	11	9	6	7	9
Longitudinal Acc	14	10	13	5	3	3
Brake Pressure	18	18	18	17	13	17

(a) RCD Signal Ranks

Signal	**MI**	**SU**	**HSIC**	$\mathbf{Z_{MI}}$	$\mathbf{MD_{MI}}$	$\mathbf{MR_{MI}}$
Minute Counter	1	2	13	20	23	24
Time	2	1	3	17	21	21
Ambient Temperature	3	5	20	21	19	23
Longitude	5	4	6	11	8	8
Engine Speed	6	6	1	2	7	3
SWA Speed	7	9	5	1	1	1
Vehicle Speed	9	8	2	7	11	12
Latitude	10	3	4	10	6	15
Throttle Position	14	13	9	4	4	4
Longitudinal Acc	17	20	17	12	13	13
Susp RearRight	19	18	19	16	14	16
Second Counter	24	24	24	24	24	2

(b) CoventryDMD Signal Ranks

Signal	**MI**	**SU**	**HSIC**	$\mathbf{Z_{MI}}$	$\mathbf{MD_{MI}}$	$\mathbf{MR_{MI}}$
Minutes	1	3	21	20	16	23
Ambient Temp	2	2	2	22	21	10
Engine Speed	4	4	3	8	20	22
Latitude	5	5	13	23	23	18
SWA	7	11	16	2	3	8
Pedal Pos	15	10	10	14	12	15
Gear Pos	19	8	17	21	17	12
Yaw Rate	20	21	19	5	5	20
Wiper Status	23	23	22	24	24	24

(c) WarwickDMD Signal Ranks

Table 4.2: Rank positions by MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} for illustrative signals of the (a) the RCD, (b) the CoventryDMD and (c) the WarwickDMD. Some bias signals were given lower rankings by HSIC than MI and SU, but others such as longitude and latitude, were not affected significantly. The biased signals were given lower rankings by the permutation statistics than MI and SU. Non-bias features that were ranked lowest by MI tended to also have low ranks by the permutation statistics, with the exception of MR_{MI} which seemed to prefer features with low MI scores.

features nearer the top having a higher MIs), and each bar shows the difference between the rank by MI and the rank by SU, HSIC, Z_{MI}, MR_{MI} and MD_{MI}. Features that are related to time or location were consistently ranked at the top by MI, illustrating that it is not a good ranking method for this data. Another signal, ambient temperature was also ranked highly, but it was not expected to be useful in either classification task, especially on new data. For instance, if the journey was on different roads or if the tasks were performed in a different order, these variables would be useless. This is an example of a data collection bias, as the data is not a random sample of the underlying distribution. The SU ranking also had signs of this bias and was very similar to the ranking produced by MI.

The rankings produced by HSIC were in general not significantly different to those produced using MI or SU. It is also, therefore, not a suitable ranking method for this data. The ranking of the ambient temperature signal was much lower in the RCD and the CoventryDMD, whereas it was unchanged in the WarwickDMD. The time signal in the RCD dataset was given a much lower ranking than with MI or SU, whereas its ranking was maintained in the CoventryDMD and the WarwickDMD. The minutes signal in the CoventryDMD and WarwickDMD, however, was significantly lower in the HSIC ranking than the MI ranking. Finally, the HSIC rankings of the longitude and latitude signals were slightly lower than their MI rankings for the RCD and the WarwickDMD, but latitude was given a higher ranking in the CoventryDMD dataset.

Five alternative permutation ranking statistics were considered, namely: ranking by the significance value [65], rejecting features with an MI below a significance threshold [37], Z_{MI} (Equation 4.17), MR_{MI} (Equation 4.18) and MD_{MI} (Equation 4.19). Using the p-value either directly or as a threshold was not suitable, however, because many of the p-values were either zero or close to zero. Therefore, we focused on the three rankings, Z_{MI}, MR_{MI} and MD_{MI}.

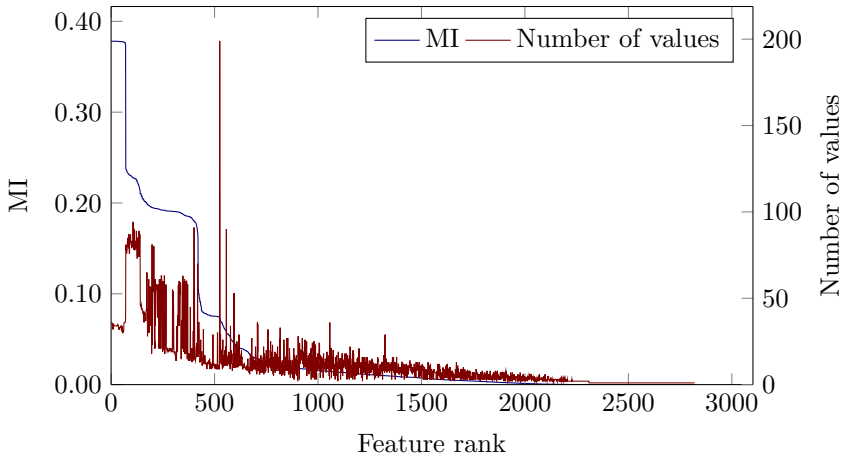
The time, location and temperature features of the CoventryDMD and RCD datasets have large negative values for each of the rankings, shown in Figures 4.7,

4.8, and 4.9. This means they were ranked much lower in the permutation rankings than in the original MI rankings. The change in ranks for other features were much smaller for the Z_{MI} and MD_{MI} rankings, indicating that they may be more suitable for application to vehicle telemetry data. The MR_{MI} ranking, seemed to prefer features that have very low correlation to the class labels by MI, such as second counter. Further to this figure, Table 4.2 displays the ranking of some illustrative signals by MI, SU, HSIC, Z_{MI}, MD_{MI}, and MR_{MI}. In this, it is clear that signals such as time or longitude, which were ranked highly by MI, were ranked very low by these permutation statistics. Also, signals that were intuitively expected to perform well, like SWA and throttle position, were much closer to the top. Again, the MR_{MI} ranking performed very poorly and seemed to select those features with low correlation to the class labels.

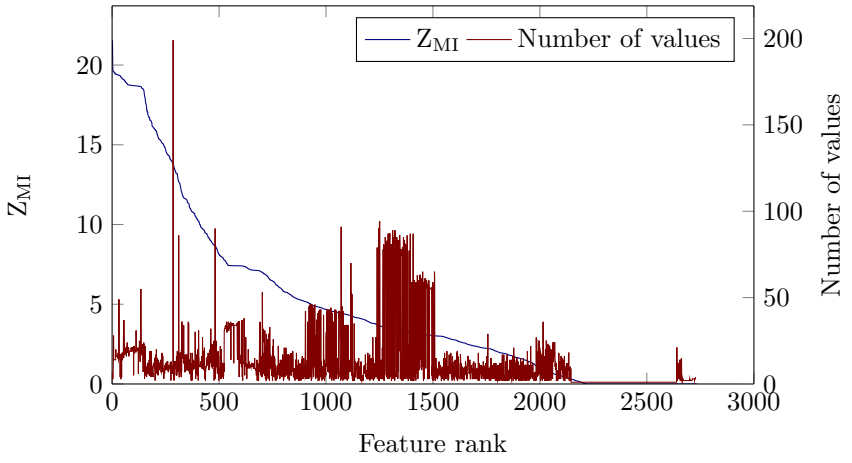
The selection bias for the RCD, the CoventryDMD, and the WarwickDMD are illustrated in Figures 4.10, 4.11 and 4.12 respectively. In each figure, plot (a) shows the MI feature scores plotted with the number of values a feature takes, and plot (b) shows the Z_{MI} feature scores. In each plot the x-axis represents the rank of the features by MI or Z_{MI}. It is clear through the correlation of the lines, that MI ranked many-valued features highly in these datasets. Z_{MI} scores show a much reduced relationship to the number of values of a feature.

4.6.3 Classification

The AUC values of the classification evaluations for each of the feature ranking strategies for the RCD and the CoventryDMD are shown in Figures 4.13 and 4.14. A dynamic blocking strategy with a cyclic shift was used for the permutation rankings, with a block size of $l = 3000$ for the RCD dataset and a block size of $l = 1500$ for the CoventryDMD and WarwickDMD. The best AUC performance with fewer than five features was achieved by the HSIC ranking for the Multilayer Perceptron and the Z_{MI} ranking for the Random Forest. The MI, SU, and MD_{MI} rankings produced the lowest AUC performances in almost all cases, which was likely a result of them ranking the bias features higher than

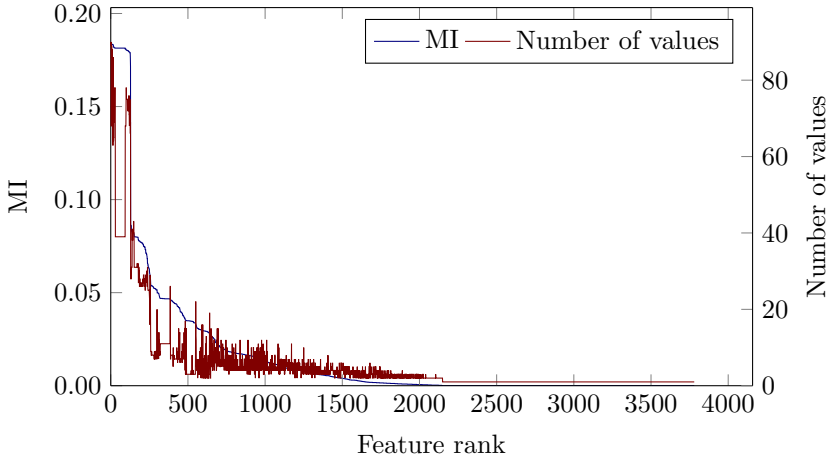


(a) RCD MI

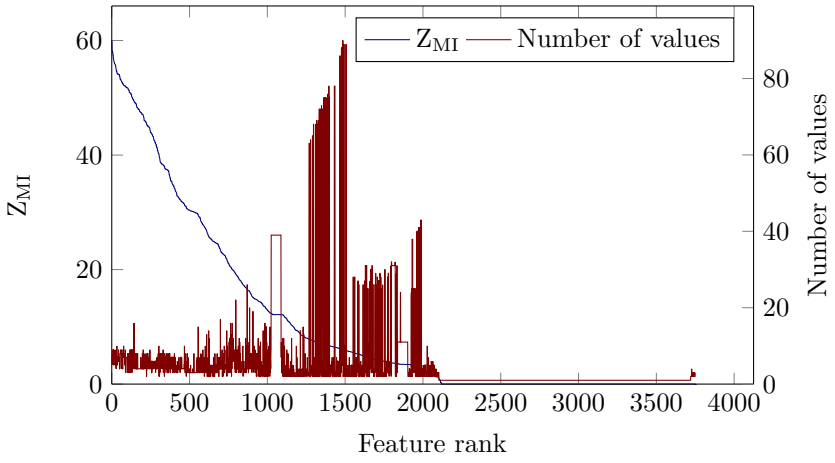


(b) RCD Z_{MI}

Figure 4.10: MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the RCD. It is clear that MI increased with the numbers of values a feature has, which is not seen with the Z_{MI} statistic.

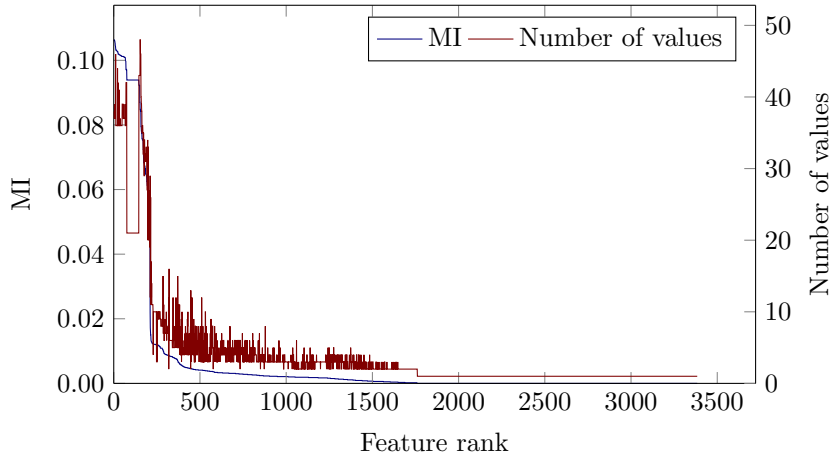


(a) CoventryDMD MI

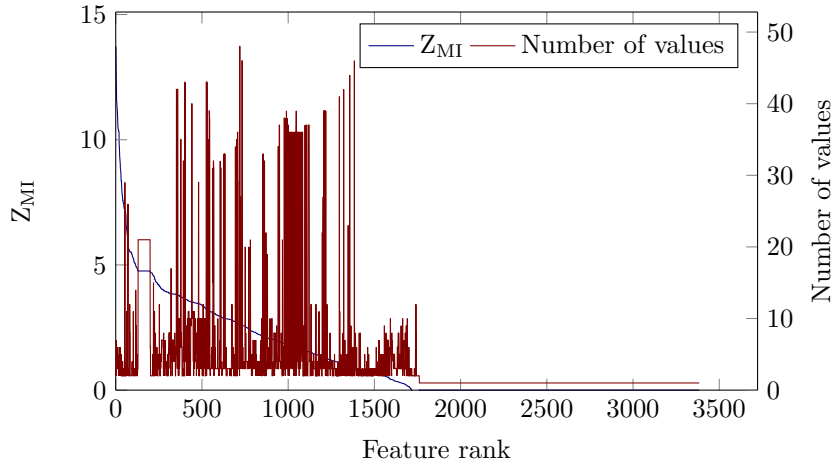


(b) CoventryDMD Z_{MI}

Figure 4.11: MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the CoventryDMD. It is clear that MI increased with the numbers of values a feature has, which is not seen with the Z_{MI} statistic.

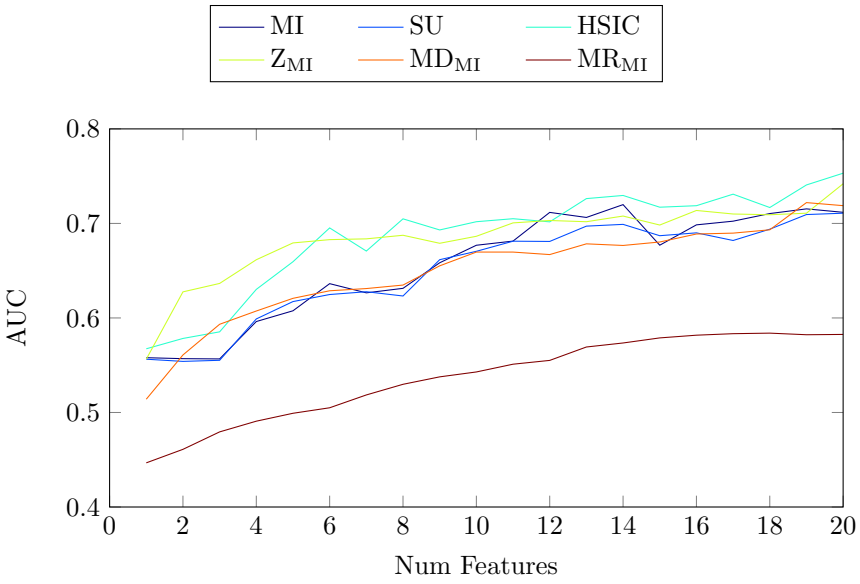


(a) WarwickDMD MI

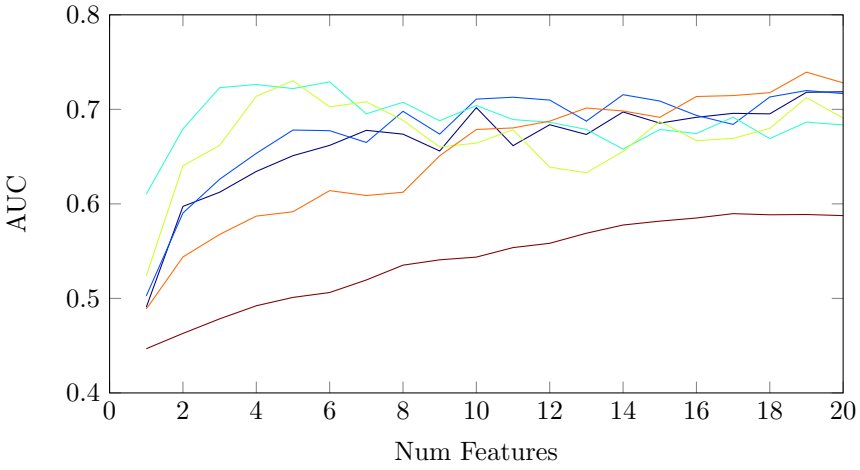


(b) WarwickDMD Z_{MI}

Figure 4.12: MI and Z_{MI}, plotted against the number of values in each feature for MI and Z_{MI} rankings of the WarwickDMD. It is clear that MI increased with the numbers of values a feature has, which is not seen with the Z_{MI} statistic.

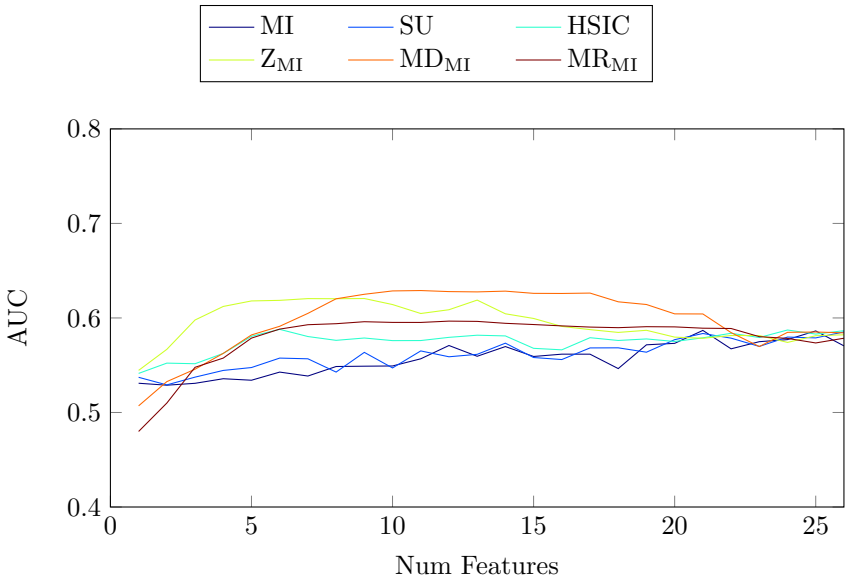


(a) RCD Random Forest

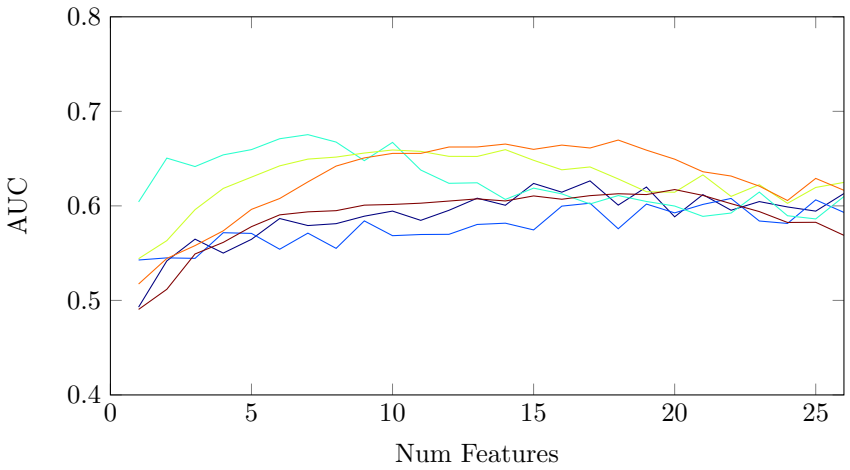


(b) RCD Multilayer Perceptron

Figure 4.13: Classification AUC scores for (a) Random Forest and (b) Multilayer Perceptron algorithms for the RCD with the MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} selection methods.



(a) CoventryDMD Random Forest



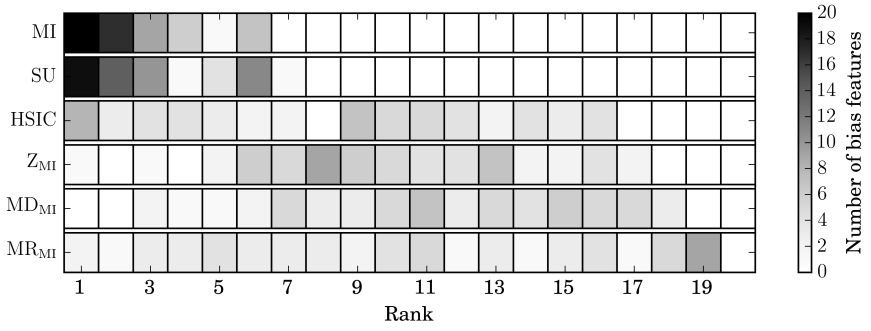
(b) CoventryDMD Multilayer Perceptron

Figure 4.14: Classification AUC scores for (a) Random Forest and (b) Multilayer Perceptron algorithms for the CoventryDMD with the MI, SU, HSIC, Z_{MI}, MD_{MI} and MR_{MI} selection methods.

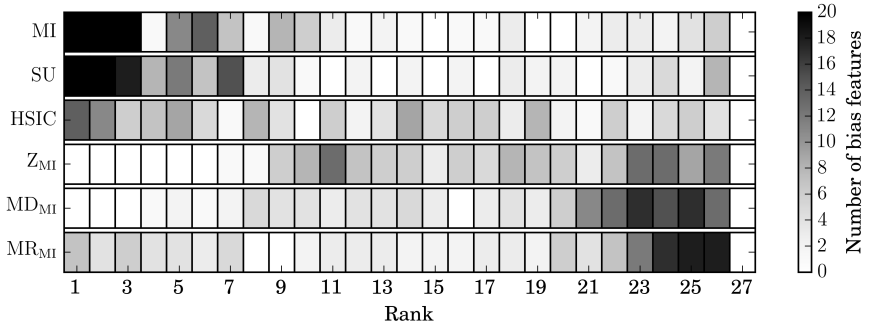
others. When selecting more than five features from the RCD dataset, the AUC performances of Z_{MI} and HSIC were the same for both classifiers. With the Multilayer Perceptron on the CoventryDMD dataset, the AUC performance decreased with more than ten features selected using the HSIC ranking. The AUC performances of the Z_{MI} and MD_{MI} rankings remained comparable up to 15 and 20 features respectively, at which point they decreased.

Classification evaluations with the WarwickDMD provided poor AUC performances of around 0.5, This is no better than choosing randomly between the *normal* and *distracted* conditions for each testing sample, and so results are not presented here. This indicates that the models built on data from several drivers were unable to successfully predict whether they were cognitively distracted or not. This is investigated further in Chapter 6, where models are built for smaller groups of drivers and individuals.

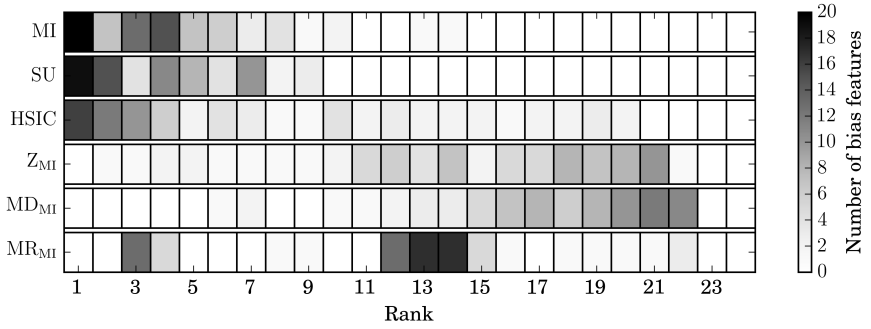
Figure 4.15 shows a histogram of the number of times ranks were assigned to the bias features by the MI, SU, HSIC, Z_{MI}, MD_{MI}, and MR_{MI} ranking methods for the (a) RCD, (b) CoventryDMD and (c) WarwickDMD. The shade of a block represents the number of bias features given a particular rank, with darker shades meaning higher numbers of bias features. MI and SU ranked bias signals highly in most iterations, which is a possible cause for their poor AUC performances. Although selecting fewer than five features using the HSIC ranking provided high AUC performance, especially with the Multilayer Perceptron, the features selected were often of bias signals. This was much less often the case for the Z_{MI} and MD_{MI} rankings, which rarely ranked bias signals in the top five. Given this, models built with features selected using the Z_{MI} and MD_{MI} rankings, rather than those selected using MI, SU, or HSIC, are more likely to be effective.



(a) RCD



(b) CoventryDMD



(c) WarwickDMD

Figure 4.15: Histogram of the number of times ranks were assigned to the bias features by the MI, SU, HSIC, Z_{MI}, MD_{MI}, and MR_{MI} ranking methods for the (a) RCD, (b) CoventryDMD and (c) WarwickDMD. The shade of a block represents the number of bias features given a particular rank, with darker shades meaning higher numbers of bias features. MI and SU assigned high rankings to bias features in all cases and for both datasets, indicated by the dark blocks in the left of their row. HSIC and MR_{MI} performed slightly better, with fewer dark blocks in the higher ranks. Z_{MI} and MD_{MI} both ranked the bias features lower than the other ranking strategies, indicated lighter grey blocks in the higher ranks and darker blocks in the lower ranks.

4.7 Conclusions

This chapter investigated a method of adapting the permutation method for use with temporally or spatially dependent data. The permutation method was first described in a general setting, before introducing the blocked-permutation method. Next, the validity of the blocked-permutation method was argued under assumptions of locally dependent sequences.

Three different blocking strategies were applied in an empirical study of the blocked-permutation method. The *static* blocking strategy [75] was extended to mitigate issues with periodicity in the data, in the form of *dynamic* and *single-value* blocking. Also, the static and dynamic strategies were investigated both with and without applying a cyclic shift to the data before each permutation [2]. The blocked-permutation method was performed for each of the blocking strategies over a wide range of block lengths on two datasets of vehicle telemetry.

In these experiments we found similar patterns for each of the RCD, the CoventryDMD and the WarwickDMD over the block sizes investigated. When the block size was small, the permutation method did not provide reasonable results. When the block size was large enough to introduce independence between blocks, we found that the permutation method provides stable significance measures. Finally, as the block size approached that of the sample size, the number of permutations which would be produced using the static and dynamic strategies decreased, and it again did not provide reasonable results. The application of the cyclic shift, however, extended the range of suitable block sizes substantially.

Two non-parametric ranking metrics were proposed for performing feature selection, namely MR_{MI} (Equation 4.18) and MD_{MI} (Equation 4.19). These were then compared, ranking by the significance value [65], rejecting features with a MI below a significance threshold [37] and normalising MI by a parameterised permutation distribution (Equation 4.17) [118, 155]. We found that using the p-value directly, or as a threshold in the ranking, was not suitable for

these datasets, as many p-values are zero or close to zero. The Z_{MI} and MD_{MI} ranking metrics produced similar feature rankings as those that would be expected by a human expert. The MR_{MI} metric failed when there were zeros in the permutation distribution, which was the case for all of the vehicle telemetry datasets. In classification experiments, we found that the performance of a classification algorithm can be affected by the bias features selected using the MI and SU rankings.

We have also shown that, because the permutation-based rankings do not select such bias features, the AUC performances in classification evaluations was higher for the RCD and the CoventryDMD. The performances of HSIC ranking was also higher, even though some non-generalisable features were selected. The HSIC ranking would therefore be expected to have poorer performance on new data, if it was collected in a different location. It may be possible to combine these approaches in a feature selection framework to increase performance further. For the WarwickDMD we found that AUC performance was no better than a random classifier when evaluated using data from all drivers. This implied that good models cannot be built for all the drivers, and so we investigate models built for subsets of drivers and individuals in Chapter 6

This chapter considered the relevancy of features and their ability to generalise to new data recorded in a different situation or at a different time or location. For a successful feature selection process, however, redundancy between features should also be considered [76] with the permutation method. In Chapter 5, efficient redundancy computation using the permutation method is investigated and applied in the minimal Redundancy Maximal Relevance (mRMR) framework [113].

CHAPTER 5

Redundant permutation feature selection

In Chapter 4 we presented permutation normalised Mutual Information (MI) for temporal data, such as vehicle telemetry, and ranked features solely by their relevance to the target variable. In general, however, redundancy between features also affects the performance of models built using them. Filters for feature selection, such as minimal Redundancy Maximal Relevance (mRMR), typically consider both relevancy and redundancy [76] via the same measure, such as MI or Symmetrical Uncertainty (SU). Permutation methods as presented in Chapter 4 are, however, computationally very expensive. Each individual permutation method, for instance, consists of thousands of permutations. Using a permutation statistic such as Z_{MI} for both relevancy and redundancy is therefore infeasible. Computing normalised MI relevancies and redundancies between m features requires $m + m^2$ permutation methods, which is prohibitive for large feature sets. In this chapter the PC_{Cor} redundancy measure is introduced, which is the Pearson correlation between permutation distributions produced from a common target during the relevancy calculations. The PC_{Cor} measure shown to approximate all m^2 redundancies while performing only m permutation methods for the relevancies, overcoming the problems with using Z_{MI} for both relevancy and redundancy.

Autocorrelation and temporal artefacts are not considered in this chapter, and vehicle telemetry data is not used. This is both for simplicity and generality, as the aim of this chapter is to propose a permutation redundancy measure that reflects the properties of Z_{MI} and is feasible to compute with even large datasets. The techniques developed in this chapter and in Chapter 4 are combined in Chapter 6, where temporal artefacts are again considered. Here, simulated data

and non-temporal datasets outlined in Section 3.4 are used. Simulated data is used to show that PC_{Cor} holds similar properties to normalised MI, and then the UCI and Tuned IT datasets are used in classification evaluations.

5.1 Introduction

Supervised feature selection aims to choose a subset of features that will provide high performance when used in a learning algorithm. As discussed in Section 2.1.7 there are several approaches to feature selection, and in this thesis filter methods are considered as they are efficient for large datasets. Filter methods in general aim to select features that are relevant to the target while being unrelated to each other. Feature clustering, for example, clusters features using their correlation as a distance measure, and the feature with highest relevancy in each cluster is selected [67]. Where the number of features required is known k-means can be applied. If the number of clusters is unknown, an iterative approach where new clusters are generated if a feature is sufficiently different to existing clusters [67], or through computing the minimum spanning tree of the redundancy graph [138], can be used. Other approaches employ genetic algorithms and use fitness functions based on the total relevancy of selected features combined with their redundancy [16].

Another approach, introduced by Koller and Sahami [77], uses the concept of Markov blankets from Bayesian networks to describe the optimal feature set. The Markov blanket of a target variable is the smallest set of features that maximally describe the target variable [16, 64]. It can be computed by iteratively eliminating the feature that least changes the probability distribution of the target, conditioned on the remaining features [77], although this is an expensive procedure computationally.

In this chapter we study the commonly used mRMR filter for feature selection, as proposed by Peng et al. [113]. In this framework, the relevancy,

$Rel(\mathbf{F}, Y)$, of a feature set $\mathbf{F}$ of size $|\mathbf{F}|$, is defined as

$$Rel(\mathbf{F}, Y) = \frac{1}{|\mathbf{F}|} \sum_{X_i \in \mathbf{F}} Cor(X_i, Y), \tag{5.1}$$

where $Cor(X_i, Y)$, is a measure of the relationship between the feature, X_i, and the target, Y. The redundancy is defined as

$$Red(\mathbf{F}) = \frac{1}{|\mathbf{F}|^2} \sum_{X_i, X_j \in \mathbf{F}} Cor(X_i, X_j). \tag{5.2}$$

The most common form of mRMR aims to select the feature set, $\mathbf{F} \subseteq \mathbf{X}$, that maximises the difference between the relevancy and redundancy of the features,

$$mRMR(\mathbf{F}, Y) = Rel(\mathbf{F}, Y) - Red(\mathbf{F}), \tag{5.3}$$

although several other variations exist [58]. Finding the optimal feature subset is infeasible, and so in practice a forward greedy search is used to iteratively select the feature that satisfies,

$$\max_{X_i \in \mathbf{X} \setminus \mathbf{F}} Rel(\{X_i\}, Y) - Red(\mathbf{F} \cup \{X_i\}), \tag{5.4}$$

where $\mathbf{F}$ is the set of currently selected features at each step.

In most applications of mRMR, $Cor(\cdot)$ is given by MI [58], and this is referred to as $MImRMR$ in this thesis. MI is biased as discussed in Chapter 4, however, and increases with the number of values a variable has, which harms the selection process. One way to reduce this bias is to normalise MI by the entropy of the variables and target, as in SU [144, 160]. Where SU is used in place of $Cor(\cdot)$ for mRMR, it is referred to as $SUmRMR$. This approach is also imperfect since it does not account for other potential biases in the data or feature selection process [65]. Another approach to mitigating these biases is to use the permutation method [65].

The remainder of this chapter is structured as follows. Two approaches

to combining a permutation method with redundant feature selection are discussed using mRMR as an example, and the permutation redundancy measure is introduced in Section 5.2. In Section 5.3, we apply this redundant feature selection method to data available from the UCI repository, with redundancy being artificially introduced to the data. Finally, we present our conclusions in Section 5.4

5.2 A redundant permutation feature selector

The simplest method of using the permutation method with mRMR is to use a permutation statistic, such as Z_{MI}, in computing relevancy (Equation 5.1) and redundancy (Equation 5.2), instead of Cor. This approach, however, requires m permutation methods to select each new feature from a set of m features. Each individual permutation methods consists of P permutations, meaning that this is prohibitive for even relatively small feature sets. In the worst case, where a full ranking is required or if the ranking algorithm requires a full redundancy analysis, is $m+m^2$ permutation methods, or $Pm+Pm^2$ permutations. Therefore, we propose a redundancy metric that is calculated directly from the permutation distributions produced in computing relevancy. Specifically, we suggest that the similarity of, or distance between, the relevancy permutation distributions be used to estimate redundancy. This approach requires exactly m permutation methods (Pm permutations) to select any number of features or to rank the full feature set.

5.2.1 Permutation redundancy

If two binary features, X_1 and X_2, are mutually redundant and $Cor(X_1, X_2) \approx 1$, then we can say that their relevancies are similar; $Cor(X_1, Y) \approx Cor(X_2, Y)$ for any target Y. A corollary of this is that dissimilar relevancies, $Cor(X_1, Y) \not\approx Cor(X_2, Y)$, imply that the features are not redundant; $Cor(X_1, X_2) \not\approx 1$. Unfortunately similar relevancies, $Cor(X_1, Y) \approx Cor(X_2, Y)$, do not guarantee

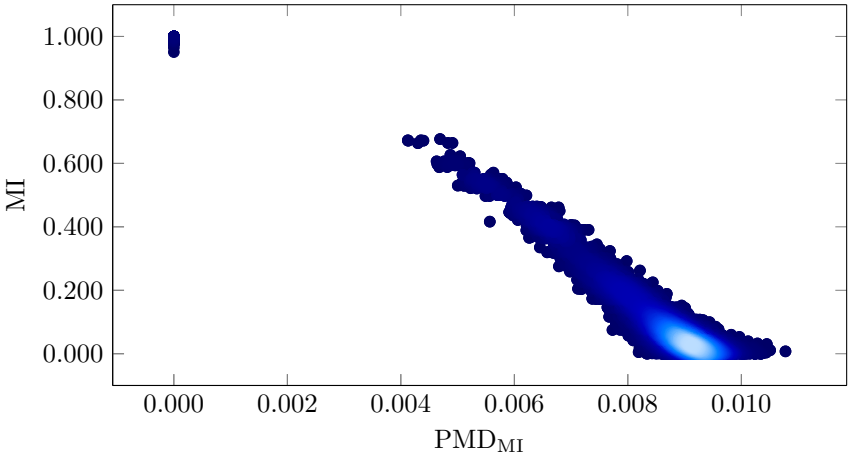
that the features are redundant, and there may be unrelated features with similar relevancies. Knowledge of relevancies does, however, provide some insight into the feature redundancy relationship. For instance, if the two relevancies, $Cor(X_1, Y)$ and $Cor(X_2, Y)$, are similar then the features are more likely to be redundant than if the relevancies are very different. Furthermore, if it is known that the features have a similar relevancies with many different targets, the likelihood of their redundancy is increased. This is the basis of the proposed permutation redundancy measure.

The permutation redundancy measure is computed by performing the permutation method for several features simultaneously, permuting only the target at each iteration. For a given permutation, Y', the permutation correlations, $Cor(X_i, Y')$ are recorded for all features $X_i \in \mathbf{F}$. Imagine that for all permutations of Y, $\Psi(Y)$, the permutation correlations for features X_1 and X_2 are similar, i.e. $Cor(X_1, Y') \approx Cor(X_2, Y') \ \forall \ Y' \in \Psi(Y)$. In this case it is reasonable to conclude that X_1 and X_2 are related and redundant features. If the features were not related, some proportion of the permutation correlations are expected to be dissimilar. As when computing the p-value of the observed statistic, more confidence can be assigned to the similarity of the features as more permutation statistics are computed.

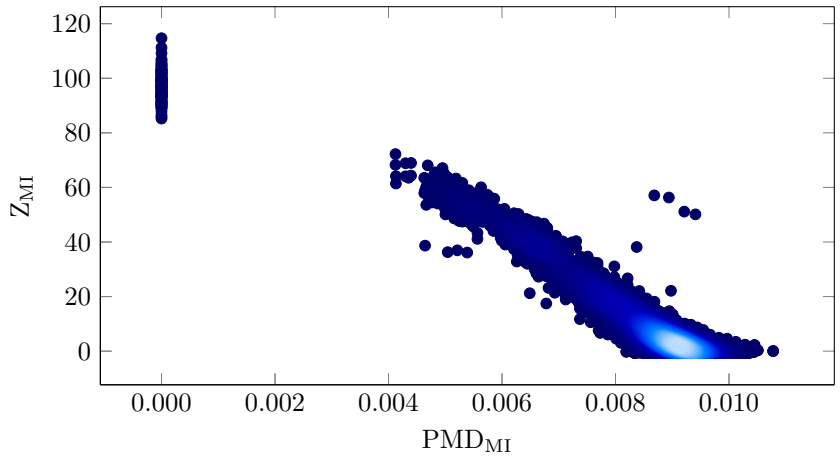
One method for quantifying permutation redundancy is the mean difference between permutation correlations,

$$PMD_{Cor}(X_1, X_2, Y) = \frac{1}{|\Psi(Y)|} \sum_{Y' \in \Psi(Y)} ||Cor(X_1, Y') - Cor(X_2, Y')||. \quad (5.5)$$

This measure captures directly the difference in $Cor(\cdot)$ values for features over different permutations of the target. Figure 5.1 shows two scatter plots of PMD_{MI} (where MI is used instead of $Cor(\cdot)$ in Equation 5.5) against (a) MI and (b) Z_{MI} for a simulated binary dataset. The data is simulated by generating a uniform binary string of 100 independent samples which is taken to be the target, Y. A total of 125 features are then generated by copying this target and



(a) MI x PMD_{MI}



(b) Z_{Cor} x PMD_{MI}

Figure 5.1: Scatter plots of (a) MI and (b) Z_{MI} (y-axis) against PMD_{MI} (x-axis). Lighter points indicate higher density regions. The Pearson correlation all the measures is -0.980.

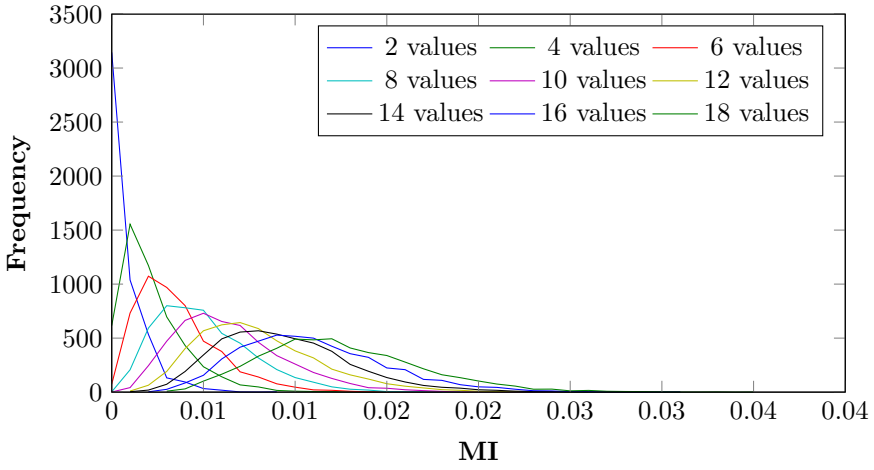


Figure 5.2: Permutation distributions of features with increasing dimensionalities computed from a common target.

changing a percentage of the sample values randomly. The features are separated into five sets of 25, each of which has a different percentage of the sample values altered. Specifically, the percentages of changed samples are 5%, 10%, 20%, 30%, 40%; producing features varying in levels of relevancy and redundancy. Each point in the scatter plots are the redundancies computed between two of the features, and higher density regions are represented by lighter points. In all the permutation methods, $P = 1000$ permutations were used. These plots show that PMD_{MI} has a close linear relationship with MI and Z_{MI}, and the Pearson correlation is -0.980 for both.

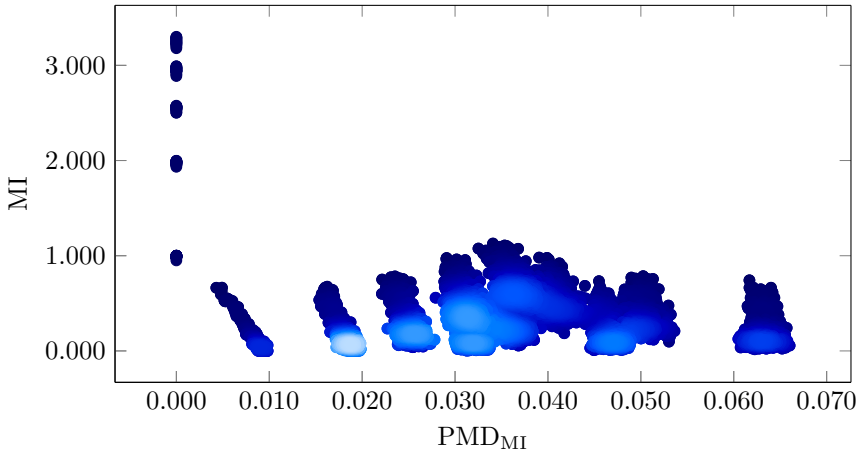
The PMD_{MI} redundancy measure, however, suffers from a similar bias to that of MI with non-binary variables. This is because permutation distributions generated from variables with more values tend to contain higher MI values and are not directly comparable to those generated from variables with fewer values. To illustrate this, Figure 5.2 shows five permutation distributions of nine simulated features computed from a common binary balanced target. Each of the features can be used to predict perfectly the values of the target, but their dimensionalities vary from two to eighteen. For the features with higher numbers of values, the permutation distributions contain more distinct and larger MI

values. Because of this, a measure that is able to compare distributions of different ranges is appropriate. One such measure is the Pearson's Correlation Coefficient (PCC) between permutation distributions,

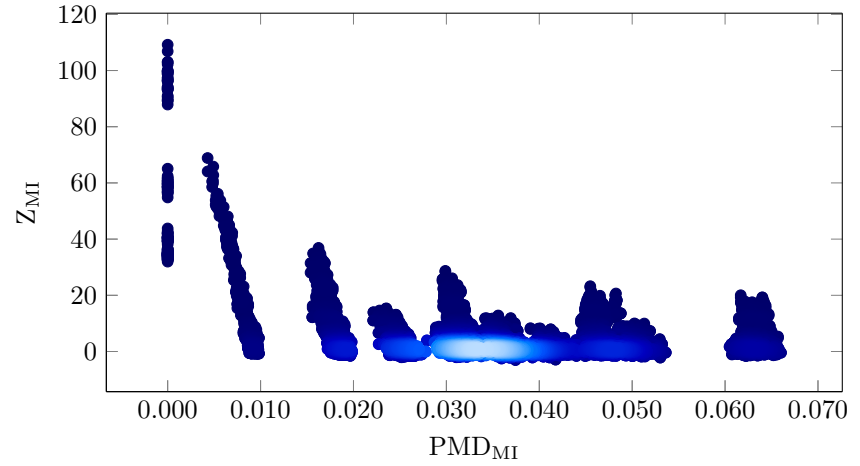
$$PC_{Cor}(X_1, X_2, Y) = PCC(Cor(X_1, Y'), Cor(X_2, Y') : \forall Y' \in \Psi(Y)). \quad (5.6)$$

Simulated data is again used to investigate the relationships between PMD_{MI} and PC_{MI} (where MI is again used in place of $Cor(\cdot)$), and MI and Z_{MI}. The dataset is first generated in the same way as before, and the target variable remains a uniform random binary string of 100 independent samples. Next, the cardinality of several features is increased in order to increase their entropy and bias their MI with other features. Each set of 25 features with the same number of value changes is split once more into 5 subsets. In the first subset, the features are kept the same and remain binary. In the second, each of the feature values are divided uniformly at random into two, creating features of cardinality 4. The third subset has each of the feature values divided into three, while the fourth and fifth subsets have features of cardinality 8 and 10 respectively. This creates a simulated dataset with 5 features for each value change and value split combination, totalling 125 features.

The scatter plots Figure 5.3 show that for the PMD_{MI} measure there is little correlation with either (a) MI or (b) Z_{MI} when the features have varying numbers of values. The plots in Figure 5.4 show that, in this case, PC_{MI} is again not related to (a) MI, but is highly related to (b) Z_{MI}. This provides evidence that the PC_{MI} redundancy measure does not exhibit the bias in MI, which is rectified by Z_{MI}. Therefore, using this measure may be beneficial to redundant feature selection with the permutation method, as it can be used as a surrogate for permutation normalised MI so that m^2 permutation methods do not have to be performed.

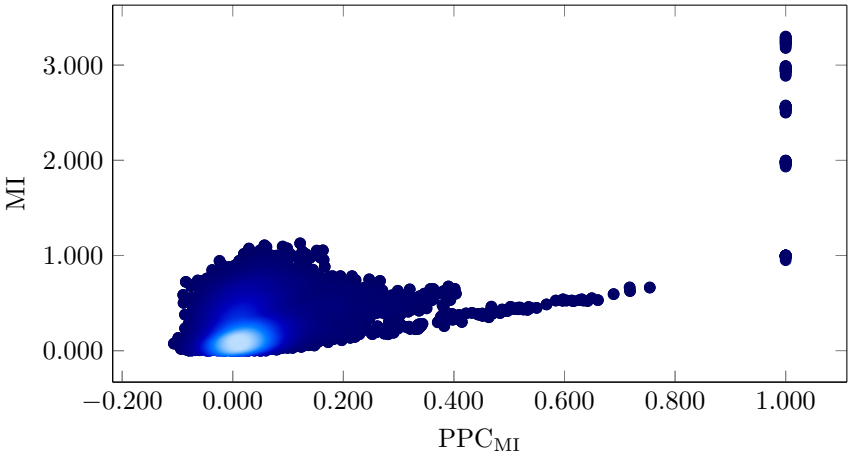


(a) MI x PMD_{MI} (−0.078)

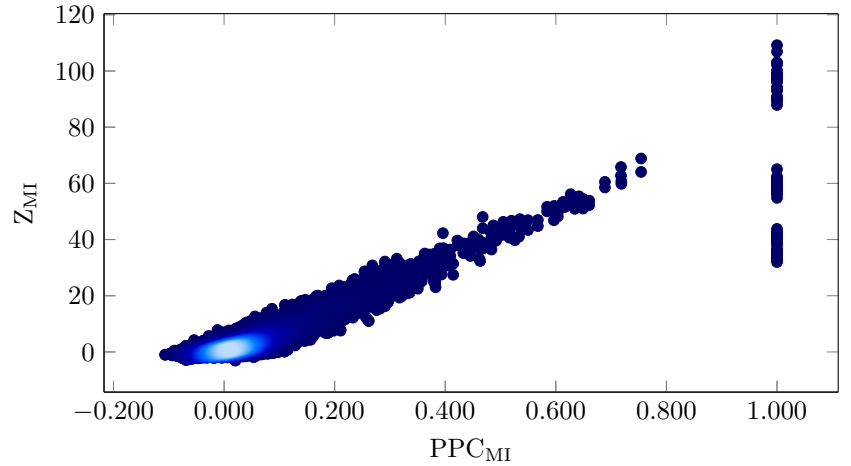


(b) Z_{MI} x PMD_{MI} (−0.295)

Figure 5.3: Scatter plots of (a) MI and (b) Z_{MI} (y-axis) against PMD_{MI} (x-axis). Lighter points indicate higher density regions. The Pearson correlations of the redundancy measures are shown in braces after their subtitles.



(a) MI x PC_{MI} (0.567)



(b) Z_{MI} x PC_{MI} (0.893)

Figure 5.4: Scatter plots of (a) MI and (b) Z_{MI} (y-axis) against PC_{MI} (x-axis). Lighter red points indicate higher density regions. The Pearson correlations of the redundancy measures are shown in braces after their subtitles.

5.2.2 Redundant permutation feature selection

One final consideration when using permutation redundancy metrics is how to combine them in a redundancy feature selection framework such as mRMR. One method of doing this is to use Z_{MI} as a measure of feature relevancy (in place of $Cor(\cdot)$ in Equation 5.1), and PC_{MI} as a measure of redundancy (in place of $Cor(\cdot)$ in Equation 5.2). Features that maximise the difference between relevancy and redundancy can then be selected iteratively in a forward greedy search, as in Equation 5.4. This approach, however, is far from ideal, because the ranges of Z_{MI} and PC_{MI} are very different. In fact, PC_{MI} has a range much smaller than Z_{MI}, which causes this feature selection process to assign more importance to relevancy than redundancy. To deal with this, the relevancies and redundancies to be considered in each selection step are both normalised between 0 and 1. This means that the most relevant feature that is not yet selected will have a relevancy score of 1. Likewise, the least redundant unselected feature will have a redundancy score of 0. This is as if relevancy and redundancy are being considered of equal importance when selecting each feature, and we refer to this method as $PmRMR$. Another approach that can be used in conjunction with this is to weight the redundancy term in order to counteract the bias [28, 154]. This requires the weighting parameter to be chosen and optimised, which is beyond the scope of this thesis.

5.3 Evaluation

To evaluate the $MImRMR$, $SUmRMR$ and $PmRMR$ feature selection methods, we used the non-temporal datasets listed in Table 3.7. These datasets were chosen because of their range in size and features, as well as their use in previous feature selection literature [58]. All samples with missing values were first removed from the dataset, before numeric or real valued features were discretised using the minimum descriptive length method [31]. At this point, features with only one discrete value were discarded as they contain no infor-

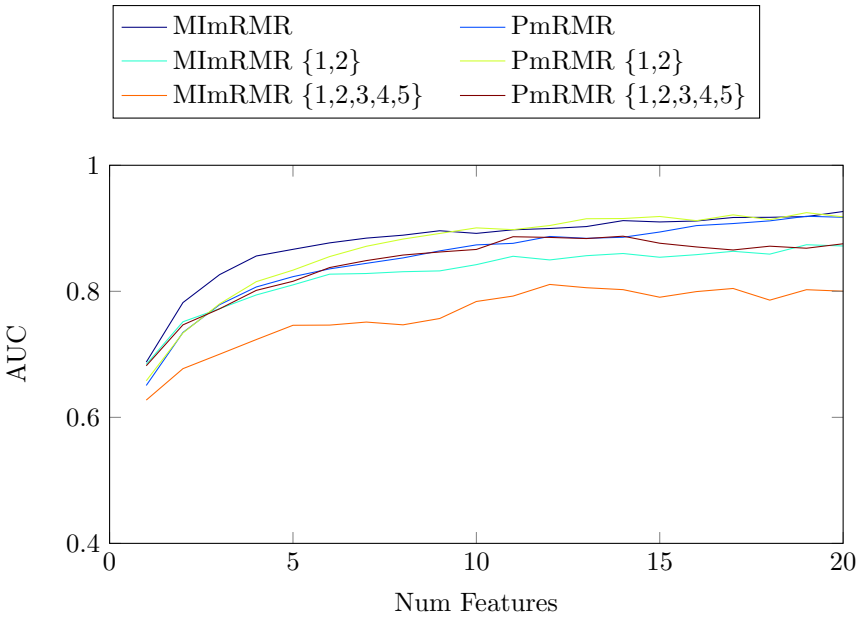
mation. This discretisation is so features can be generated from existing ones, while changing their sample values to worsen their predictive performance and increasing their dimensionality to bias MI. Each of these datasets had redundant features injected through a generation process similar to earlier simulations in this chapter. In this case, features in a dataset were copied several times, with a percentage of their values changed in order to worsen their predictive abilities. Next, the values of the copied features were split in order to increase their entropy and make them appear as better predictors. All of the original features were retained in the datasets, while the target to be predicted was not copied or modified. Ideally a feature selector should choose the original features over the copies, as copies have added noise which makes them worse predictors, and increased entropy which may lead to over-fitting.

From each dataset, 5 new datasets were generated by copying original features and increasing their dimensionalities. In all cases, before increasing the dimensionality of a feature, 5% of the values were changed to worsen their predictive abilities. The target variable was not copied or altered in any of the new datasets. The 5 datasets, referred to as $\{1\}$, $\{1, 2\}$, $\{1, 2, 3\}$, $\{1, 2, 3, 4\}$, and $\{1, 2, 3, 4, 5\}$, had different numbers of features added to the original ones with different numbers of splits in their values. Dataset $\{1\}$ had double the number of features as the original, and the values of each added feature were split once to double its dimensionality. All of the features present in $\{1\}$ were also in $\{1, 2\}$, with one extra copy of the original features having two splits in their values to triple their dimensionalities. In each of the subsequent datasets an extra copy was added on top of the previous, with one extra split in values applied. In the third, fourth and fifth datasets therefore, there were four, five, and six times as many features as in the original dataset, with dimensionalities multiplied by four, five and six respectively.

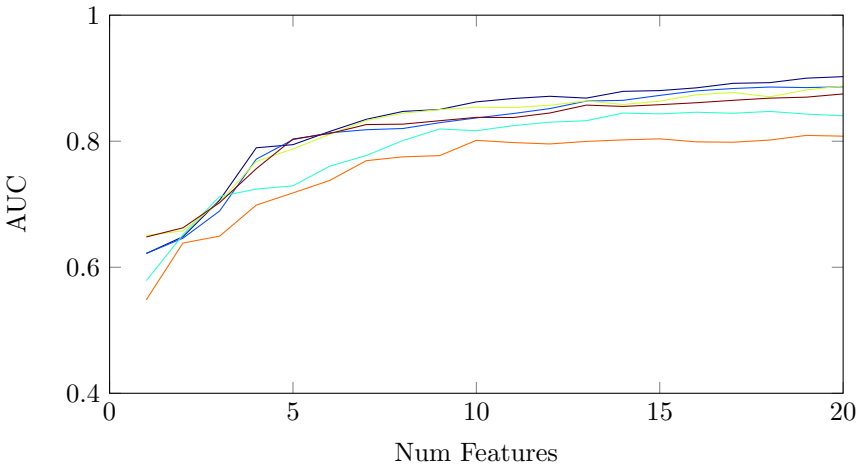
Datasets with more injected features with higher numbers of value splits were expected to be harder to select good features from. The least difficult case was expected to be selecting features from the original datasets, and the

most difficult was expected to be the cases where 5 extra features per original feature were injected. To investigate these hypotheses for each dataset, a random subset validation procedure with ten train-test iterations was performed. In each repeat, 50% of the samples were taken uniformly at random to be the training data, from which features were ranked using forward selection with $MImRMR$, $SUmRMR$ and $PmRMR$. To consider relevancy and redundancy of equal importance and for a fair comparison, the relevancies and redundancies all cases were normalised between 0 and 1, before choosing each feature. Twenty classifiers were then built with increasing numbers of features (between one and twenty) taken from the top of these rankings. The classification algorithms used were Naïve Bayes, Decision Tree, Random Forest, and Support Vector Machine (SVM), which are all available in the WEKA library [160]. The remaining 50% of the samples in each iteration were used as testing data to produce a performance measure in the form of a weighted Area Under the Receiver Operator Characteristic Curve (AUC). Finally, because features were ranked using the same training samples for both ranking methods, the AUC performances produced during each testing iteration can be compared directly.

For illustration, the mean AUC performances over the ten iterations of the TR 21 and Musk 1 datasets, using the Naïve Bayes (NB), SVM, Decision Tree (DT) and Random Forest (RF) classifiers respectively are shown in Figure 5.5. The plots are representative of using other classifiers with different datasets, and show that AUC decreases as more features with higher dimensionalities are present. It also shows that performance decreases less when features are selected using $PmRMR$ than with $SUmRMR$. In some datasets, where AUC performances of above 0.95 were found with fewer than 5 features, performances were unaffected generally with all four classifiers. Such datasets included Chess, Congress, Soybean (S), Spambase, Splice and Wine. This result was observed more often with the Decision Tree classifier, which is prone to over-fitting. In other experiments we also added features with different amounts of sample value changes, but did not find this to significantly affect the performance of any of



(a) Random Forest, Musk 1



(b) SVM, TR 21

Figure 5.5: Mean AUC scores achieved over ten evaluations when selecting between one and twenty features from (a) Musk 1 and (b) TR 11 datasets and using Random Forest and SVM respectively. AUC performance is lower with copied features injected, and $PmRMR$ outperformed $MImRMR$ in general.

	Original		{1}		{2}		{3}		{4}		{5}	
Classifier	**MI**	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**
NB	**1861**	968	1564	**1691**	1355	**2069**	1367	**2109**	1172	**2292**	1280	**2205**
SVM	**1653**	1042	1508	**1757**	1325	**2082**	1274	**2202**	1161	**2325**	1333	**2245**
DT	**1304**	1198	1162	**1687**	1039	**1928**	892	**2192**	852	**2249**	1066	**2075**
RF	**1792**	1245	1321	**2132**	1075	**2542**	984	**2713**	978	**2735**	1018	**2717**
Total	**6610**	4453	5555	**7267**	4794	**8621**	4517	**9216**	4163	**9601**	4697	**9242**

Table 5.1: Number of times features selected by $MImRMR$ (MI) outperformed those selected by $PmRMR$ (P), and vice versa, for each classifier over all train-test iterations. The result for the best selector in each case is highlighted in bold.

	Original		{1}		{2}		{3}		{4}		{5}	
	MI	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**	**MI**	**P**
NB	**1767**	987	**1854**	1160	**1777**	1358	**1866**	1295	**1863**	1297	**1974**	1236
SVM	**1519**	1111	**1673**	1411	**1620**	1524	**1710**	1450	**1655**	1490	**1847**	1405
DT	**1302**	1095	**1317**	1232	1339	**1419**	1294	**1431**	1297	**1541**	**1487**	1390
RF	**1684**	1317	1649	**1657**	**1705**	1681	**1850**	1563	**1900**	1549	**1943**	1538
Total	**6272**	4510	**6493**	5460	**6441**	5982	**6720**	5739	**6715**	5877	**7251**	5569

Table 5.2: Number of times features selected by $SUmRMR$ (SU) outperformed those selected by $PmRMR$ (P), and vice versa, for each classifier over all train-test iterations. The result for the best selector in each case is highlighted in bold.

the feature ranking methods.

Table 5.1 shows the number of times $MImRMR$ outperformed features selected by $PmRMR$, and vice versa, for each classifier over the 4000 train-test iterations. Cases where the same AUC was achieved by both selection algorithms were not counted in these results. As more copied features are injected into the datasets with more value splits, $PmRMR$ tends to outperform $MImRMR$ more often. The number of times $SUmRMR$ outperformed $PmRMR$ and vice versa is shown in Table 5.2. These results show that $SUmRMR$ outperformed $PmRMR$ much more often than $MImRMR$, meaning that it performed best in the majority of cases. In some cases, such as when the Random Forest classifier is used with the TR 11, TR 12 and TR 21 datasets, $PmRMR$ had higher AUC performances than both $MImRMR$ and $SUmRMR$ significantly more often than where $PmRMR$ performed worse. These datasets have the highest numbers of features, and so are likely to contain a large amount of redundancy.

	Naïve Bayes			SVM			Decision Tree			Random Forest		
Dataset	**MI**	**SU**	**P**	**MI**	**SU**	**P**	**MI**	**SU**	**P**	**MI**	**SU**	**P**
Arrhythmia	0.77	0.75	**0.82**	0.73	0.71	**0.76**	0.69	0.71	**0.72**	0.63	0.61	**0.64**
Congress	0.99	0.99	0.98	0.96	0.97	0.97	0.97	0.97	0.97	0.98	0.98	0.98
Credit	0.91	**0.93**	0.92	0.86	0.87	0.87	0.87	0.88	0.88	0.87	0.90	0.90
Madelon	0.68	0.69	0.69	0.63	0.64	0.64	0.64	0.70	**0.71**	0.61	0.72	**0.73**
Musk 1.00	0.83	**0.85**	0.83	0.73	**0.76**	0.73	0.66	**0.81**	0.78	0.72	**0.85**	0.83
Optdigits	0.93	0.94	0.94	0.92	0.93	0.93	0.73	**0.82**	0.81	0.83	0.88	0.88
Parkinsons	0.93	0.93	0.93	0.80	0.80	0.79	0.81	0.81	0.79	0.89	**0.91**	0.89
Promoters	0.94	**0.96**	0.95	0.87	0.90	0.87	0.79	0.79	0.78	0.82	**0.92**	0.91
Soybean (L)	0.93	0.93	0.93	0.93	0.93	0.93	0.92	0.92	0.92	0.82	0.83	**0.93**
Soybean (S)	1.00	1.00	1.00	**1.00**	0.99	0.98	**0.99**	0.96	0.98	0.94	**0.99**	0.98
Spambase	0.94	**0.95**	0.94	0.88	0.88	0.86	0.91	0.91	0.89	0.93	**0.95**	0.93
Splice	0.98	0.98	0.98	0.96	0.96	0.95	0.96	0.96	0.96	0.97	0.97	0.97
TR 11	0.94	0.95	0.95	0.89	0.90	**0.91**	0.83	0.90	0.90	0.87	0.91	**0.94**
TR 12	0.90	**0.93**	0.92	0.86	**0.91**	0.90	0.82	**0.90**	0.89	0.85	**0.92**	0.91
TR 21	0.86	**0.89**	0.88	0.74	**0.79**	0.78	0.65	**0.83**	0.81	0.79	0.87	0.87
TR 23	0.97	0.97	**0.98**	0.90	0.90	0.90	0.95	0.95	0.95	0.90	0.93	**0.95**
Vehicles	0.83	0.84	0.84	0.80	0.80	0.80	0.84	**0.86**	0.85	0.81	0.84	**0.85**
Wine	1.00	1.00	0.99	0.98	0.98	0.96	0.97	0.97	0.95	0.98	**1.00**	0.98
Yeast	0.76	0.79	**0.81**	0.73	0.75	**0.77**	0.70	0.73	**0.76**	0.68	0.72	**0.76**

Table 5.3: Mean AUC performances for each dataset (with the maximum number of features with $\{1, 2, 3, 4, 5\}$ splits added) and classifier for the $MImRMR$ (MI), $SUmRMR$ (SU), and $PmRMR$ (P) selection methods when selecting 5 features. The highest unique AUC over the three selection algorithms in each case is highlighted in bold.

Furthermore, as we observed in Chapter 4, and observe in Chapter 6, ranking features using SU does not remove some biases present in vehicle telemetry data.

The AUC performances of all the datasets for the $MImRMR$, $SUmRMR$ and $PmRMR$ rankings methods are shown in Table 5.3. In all cases, the top five features were used to build a model on the training data for the Naïve Bayes, SVM, Decision Tree and Random Forest learning algorithms. The highest AUC for each classifier and dataset is highlighted in bold. $MImRMR$ performed highest in only one case while $SUmRMR$ and $PmRMR$ had the unique highest performance in 21 and 16 cases respectively. In those cases where $PmRMR$ outperformed $SUmRMR$, however, the difference between the mean AUC performances was higher than in the converse case. For example, for the Soybean (L) dataset with the Random Forest classifier, the AUC performance of $PmRMR$ was 0.1 higher than that of $SUmRMR$, and for the Arrhythmia dataset this difference is 0.07 with the Naïve Bayes classifier and 0.05 with the SVM. The largest difference where $SUmRMR$ outperformed $PmRMR$ is 0.03.

A good feature ranking method should rank the original features higher

	{1}			{2}			{3}			{4}			{5}		
Dataset	**MI**	**SU**	**P**	**MI**	**SU**	**P**	**MI**	**SU**	**P**	**MI**	**SU**	**P**	**MI**	**SU**	**P**
Arrhythmia	21	23	**41**	18	20	**40**	22	26	**40**	21	24	**41**	19	27	**29**
Congress	35	38	38	34	**39**	33	29	**34**	28	30	**36**	27	25	**36**	31
Credit	35	40	**41**	27	37	**38**	19	**37**	36	16	**36**	33	12	**40**	37
Madelon	35	49	**50**	23	49	49	17	**47**	45	14	48	**49**	14	48	48
Musk 1	40	50	50	41	50	50	38	50	50	37	50	50	38	50	50
Optdigits	47	50	50	40	50	50	40	50	50	40	50	50	40	50	50
Parkinsons	39	**50**	49	35	**49**	48	30	49	49	32	48	48	36	**49**	47
Promoters	37	49	49	39	46	**50**	33	**49**	47	37	47	**49**	32	**47**	44
Soybean (L)	33	40	**50**	38	39	**50**	38	39	**50**	40	38	**50**	39	40	**50**
Soybean (S)	44	48	**49**	41	45	**47**	37	45	**47**	38	43	**48**	39	46	**48**
Spambase	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Splice	50	50	50	50	50	48	50	50	48	50	50	49	50	50	50
TR 11	44	49	**50**	39	50	50	36	50	50	37	47	**50**	39	49	**50**
TR 12	29	40	40	30	40	40	34	40	40	35	38	**40**	32	37	**41**
TR 21	33	**50**	49	25	**50**	49	27	45	**47**	23	49	49	27	47	**49**
TR 23	35	49	50	33	47	**50**	36	49	**50**	35	**50**	49	37	48	**49**
Vehicles	37	48	48	29	**49**	44	32	**47**	43	30	**47**	43	37	46	46
Wine	48	**50**	48	48	**50**	49	44	**50**	48	43	**50**	48	40	50	50
Yeast	37	49	**50**	29	48	**49**	25	45	**50**	27	41	**49**	26	40	**48**
Total	729	872	**902**	669	858	**884**	637	852	**868**	635	842	**872**	632	850	**867**

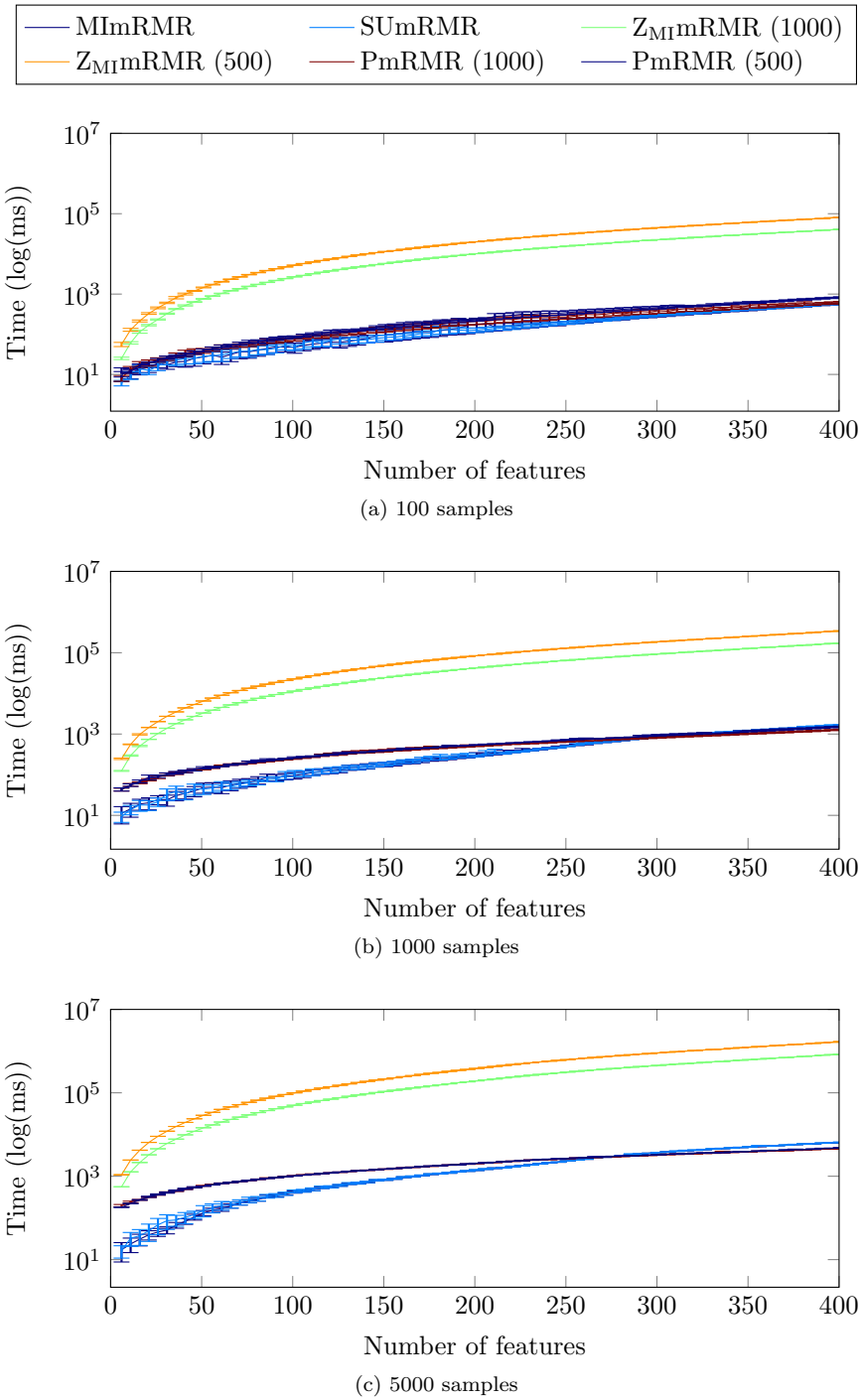
Table 5.4: Total number of the original features from each dataset ranked in the top five by $MImRMR$ (MI), $SUmRMR$ (SU), and $PmRMR$ (P) for different split types and a deform type of $\{5, 10, 20, 30, 40\}\%$. The selection algorithm with highest performance in each case is highlighted in bold. Both $PmRMR$ and $SUmRMR$ outperformed $MImRMR$ in all cases, but neither outperformed the other in general.

than the injected ones, as randomizing values in the copies means that they are worse predictors of the target. The number of times an original feature was ranked in the top five by $MImRMR$, $SUmRMR$ and $PmRMR$ for the datasets with injected features are shown in Table 5.4. Overall, as features were copied more and with more splits, fewer original features were ranked in the top five by all selection methods. Where one selection method ranked more features in the top five than the others, the result is highlighted in bold. The results show that, in the majority of cases, $SUmRMR$ and $PmRMR$ outperformed $MImRMR$ in ranking the original features highly and that $MImRMR$ was most affected by increasing the entropy of features. They also show that feature selection is data dependent, with $SUmRMR$ or $PmRMR$ outperforming each other on datasets consistently over the different redundancy types. When the top ten or twenty features in the rankings are considered, the performances of the selection methods begin to converge, with $MImRMR$ outperforming $PmRMR$

and $SUmRMR$ in some cases including for the Congress, Soybean (small) and Vehicles datasets. These results are however omitted for space reasons.

Finally, in Figure 5.6 we present the times taken to rank all features from simulated binary datasets for different numbers of samples, features and permutations. The datasets are generated to have 100, 1000, 5000 samples with the same percentages of value changes as outlined in Section 5.2.1, but with increasing numbers of feature copies. The number of features in the dataset is shown on the x-axis and the y-axis shows the log time taken to rank all features using the $MImRMR$, $SUmRMR$, $Z_{MI}mRMR$ (with 500 and 1000 permutations) and PC_{MI} (with 500 and 1000 permutations) ranking methods. The time taken to produce the full ranking increased exponentially with the number of features for all selection approaches. The $MImRMR$ and $SUmRMR$ selection methods have very similar computation times, and $MImRMR$ was slightly the slower of the two. This is unexpected as SU requires a small amount of extra computation to normalise the MI value, and therefore was expected to take the longer. The $Z_{MI}mRMR$ selection method, where a new permutation distribution is generated for each redundancy calculation, is by far the slowest method and the computation times of this method increased fastest to the number of features. The computation required by PC_{MI} increases slower with respect the size of the data, and is quicker than $MImRMR$ and $SUmRMR$ when there are over 275 features and 1000 or 5000 samples.

These computation times are consistent with a complexity analysis of the selection methods. The redundancy computation times of both $MImRMR$ and $SUmRMR$ are dependent on both the number of features and the sample size, and is $O(nm + nm^2)$ for n samples and m features. for $Z_{MI}mRMR$ (where relevancy and redundancy is given by Z_{MI}), the number of permutations is also factored into this and the computational complexity is $O(Pnm + Pnm^2)$. The $PmRMR$ method on the other hand is dependent primarily on the number of features and the number of permutations used for the relevancies only, and its complexity is $O(Pnm+Pm^2)$. Although the overhead for computing relevancies



(a) 100 samples

(b) 1000 samples

(c) 5000 samples

Figure 5.6: Computation times for the $MImRMR$, $SUmRMR$, $Z_{MI}mRMR$ (with 500 and 1000 permutations) and PC_{MI} (with 500 and 1000 permutations) methods to rank all features from a simulated binary datasets with 100, 1000 and 5000 samples and increasing numbers of features.

is larger ($O(Pnm) > O(nm)$), $PmRMR$ is computationally more efficient than $MImRMR$ or $SUmRMR$ for large numbers of features and sample sizes when the number of permutations is fixed. This is reflected in our results, where there is a small difference between computation times of $PmRMR$ with 500 and 1000 permutations with larger numbers of features and sample sizes.

5.4 Conclusion

In this chapter a combination of redundant feature selection with permutation normalised correlations was investigated. Specifically, a variant of mRMR was proposed, where relevancy and redundancy are measured by the standard score of MI (Z_{MI}). Because of the computational intractability of performing $m+m^2$ permutation methods to rank all m features in a dataset, the PC_{MI} redundancy metric was proposed. PC_{MI} is computed from permutation distributions produced during relevancy analysis of the features. As a result, all m^2 redundancies can be computed after performing only m permutation methods. On simulated data the PC_{MI} and Z_{MI} had high correlations to each other, whereas their correlations with MI and PMD_{MI} were low. In conclusion therefore, PC_{MI} can be used in redundancy analysis of features during selection.

The range of PC_{MI} is -1 to 1, which is much less than that of Z_{MI}. To use them alongside one another in a selection algorithm this disparity must be accounted for. A normalised variant of mRMR was therefore proposed, that normalised the relevancy and redundancy values in each selection step. We compared the $MImRMR$, $SUmRMR$ and $PmRMR$ feature selectors using example datasets in classification evaluations and inspections of feature rankings. New features, generated from existing ones, were added to the datasets in order to vary the levels of redundancy and bias. This had the effect of increasing the difficulty in selecting good features and building models. In ranking the features of these datasets with added bias and redundancy, we found that $PmRMR$ ranked the original features higher more often than did the

$SUmRMR$ or $MImRMR$ rankings. The original features should have higher performance in reality, as they did not have noise added to them. This result therefore showed that $PmRMR$ outperformed the other methods.

In the classification algorithms, the AUC performances of features selected by $MImRMR$ decreased significantly in many cases with more redundant and bias features inserted. In general the AUC performances did not decrease for features selected using $SUmRMR$ and $PmRMR$. Furthermore, AUC performances were significantly higher for $PmRMR$ than for $MImRMR$ or $SUmRMR$ in some cases, whereas when $SUmRMR$ performed better the improvement was generally smaller. This again showed that $PmRMR$ selected features of higher performances than $MImRMR$ or $SUmRMR$.

Finally, in comparison of computation times of the selection methods, we found $PmRMR$, $MImRMR$ and $SUmRMR$ to be significantly faster than $Z_{MI}mRMR$. $Z_{MI}mRMR$ requires m permutation methods to be performed for each selection step, and therefore this was expected. Furthermore, for big datasets with large feature sets and sample sizes we found that $PmRMR$ was computationally faster than either $MImRMR$ or $SUmRMR$. $PmRMR$ still requires m permutation methods to be performed for the relevancy calculations, so this was expected to be slower. For large numbers of features, however, the redundancy computation of computing Pearson correlations of the permutation distributions becomes faster than the MI and SU computations. This means that, for big datasets large numbers of features, $PmRMR$ is more efficient than $MImRMR$ or $SUmRMR$.

This chapter considered the permutation method without blocking, ignoring temporal aspects for simplicity. In Chapter 6 the redundancy computation methods discussed in this chapter are considered in the presence of autocorrelation with vehicle telemetry data discussed in Chapter 3. The permutation relevancies in $PmRMR$ are computed using the blocked permutation method, as discussed in Chapter 4, and the redundancies are computed from the permutation produced as in this chapter.

CHAPTER 6

Feature selection from vehicle telemetry data

In Chapter 4 we introduced a blocked permutation method and applied it to normalising Mutual Information (MI) values which were used to rank features. Permutation methods were then adapted to efficiently compute redundancies in Chapter 5. In this chapter, we combine these techniques to select features from vehicle telemetry and human activity monitoring data for a variety of tasks, outlined in detail in Chapter 3.

Selecting features from telemetry data is a challenging task that is made more complex when several extracted features are considered. Throughout this thesis, the 28 structural and statistical features listed in Table 3.8 have been extracted from each signal, over 5 different temporal sliding windows. The Coventry-JLR Driver Monitoring Dataset (CoventryDMD) for example, is made up of 494 signals that were used to generate a total of $494 \times 28 \times 5 = 69160$ features. In previous chapters, features have been selected from subsets of between 20 and 30 signals that were chosen manually. In this chapter, we consider the much more challenging problem of automatically choosing features from the full set of signals and features produced during data collection.

We observe that there are two kinds of redundancy between features extracted from signal data, namely that between features of same signal (intra-signal) and that between features extracted from different signals (inter-signal). Also, if one feature of a signal is highly relevant to the target variable, it is likely that other features extracted from this signal will also be relevant. There are several approaches to using these observations in feature selection from telemetry data, of which two are investigated. First, an investigation into the benefits of signal selection, prior to feature selection, is performed for the Road Classi-

fication Dataset (RCD). The aim of selecting from raw signals is to reduce the number of features that are considered for selection, reducing the computation required for the process.

Second, we propose a two-stage selection process to take advantage of these different types of redundancy, considering intra-signal and inter-signal redundancies separately. We illustrate the process on vehicle telemetry signal data for driver monitoring. As well as $MImRMR$, $SUmRMR$ and $PmRMR$, we also apply the $HSICmRMR$ selection method where both relevancy and redundancy are given by the Hilbert-Schmidt Independence Criterion (HSIC) as in Equation 4.14. We evaluate it using the Random Forest, Naïve Bayes, and Multilayer Perceptron machine learning algorithms. Our results show that, although it is less expensive computationally to perform selection prior to feature extraction, the highest classification performance is given by selecting features from those extracted from signals. Furthermore, the two-stage process significantly reduces the computation required because of inter-dependency calculations, while having little detrimental effect on the performance of the feature sets produced.

Finally, after discovering that models built from data collected from several drivers are unsuccessful in predicting cognitive distraction for multiple drivers in Chapter 4, we investigate in further detail the Warwick-JLR Driver Monitoring Dataset (WarwickDMD). Models are built for subsets of drivers to identify which drivers' vehicle telemetry data can be used to successfully predict cognitive distraction. We then refer to the questionnaire given to participants in the study, to look for patterns in the types of drivers where models have best performance.

6.1 Introduction

In previous chapters, a forward selection approach has been considered, such as minimal Redundancy Maximal Relevance (mRMR), which iteratively chooses

the unselected feature that maximises the difference between its relevance to the target and redundancy to other already selected features. Here, we adapt this selection process in two ways: namely, to consider signal selection prior to feature extraction, and to take advantage of redundancy structure in datasets where features are extracted from signals. In forward selection, each candidate feature will fall into one of five types and be:

- irrelevant and of no use in predicting the target,
- relevant but similar to an already selected feature extracted from the same signal,
- relevant but similar to an already selected feature extracted from a different signal,
- relevant but similar to already selected features extracted from both the same signal and different signals,
- relevant and not similar to any other feature.

Of course, only relevant features should be selected, but to reject all redundant features would be incorrect. A feature can be relevant and redundant to varying degrees, for example, and even a highly redundant feature may provide some useful information for a model to learn about the target. Furthermore, the redundancies between features extracted from the same signal and those extracted from different signals have different properties and can be treated independently in the selection process.

Selecting features from datasets where multiple features are extracted over several window lengths from hundreds of signals is expensive computationally, especially when detailed redundancy analysis is performed. We therefore investigate two methods to dealing with this complexity. To reduce drastically the number of features considered, we consider signal selection prior to feature extraction in a small case study with the RCD. Here, raw signals that are irrelevant to the target are discarded before feature extraction, meaning fewer features are considered in the selection process. Second, we consider intra-signal redundancies between features extracted from the same signal separately

to inter-signal redundancies between features of different signals. We propose a two-stage selection process, selecting a small number of features from each signal before combining these to output a final subset.

The remainder of this chapter is structured as follows. In Section 6.2 we investigate selecting signals prior to feature extraction, and consider relevancy and redundancy separately. In Section 6.3 we propose a two-stage feature selection process aimed at minimising feature and signal level redundancies, which reduces the computational cost compared to existing methods. Finally, in Section 6.3.1, we describe our evaluation strategy and present results for the proposed method alongside results existing techniques.

6.2 Benefits of signal selection

In an experiment with a subset of the RCD, we investigate the benefits of signal selection. This subset contains features that are expected to contain no biases, and only four features are extracted from each (excluding their raw values). Two statistical features, the mean and Standard Deviation (STD), and two structural features, the first and second derivatives, are extracted over sliding windows of 2.5 seconds. This length allows sufficient historical data for the features to be of use, while being small enough to be updated rapidly if the conditions change [116]. Also, in a previous study, we have shown that window lengths of over 2.5 seconds do not provide significant increases in performance without causing over-fitting [144].

In many cases of learning from Controller Area Network (CAN)-bus data [60, 103, 145, 161], feature selection is performed after feature extraction has taken place. However, because of their number, selecting from the full set of extracted features is computationally prohibitive. It is beneficial to perform selection on signals prior to feature extraction, because there are fewer signals than total features. Therefore, we investigate signal selection prior to feature extraction and explore the impact of combining relevant and redundant feature selection.

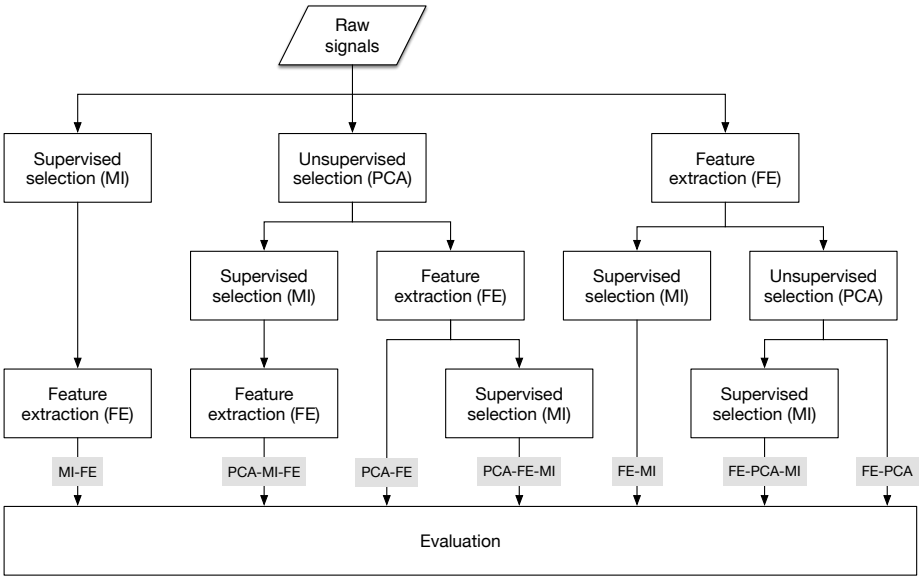


Figure 6.1: Processing methods using data, for PCA, MI and Feature Extraction (FE) in different orders. Some selection is performed on signals, prior to feature extraction. In this diagram, for example, the leftmost path of MI-FE first performs signal selection with MI, and then extracts features on the selected signals.

Figure 6.1 outlines the signal selection, feature extraction and feature selection methods investigated here. The selection methods consist of redundancy selection, Principal Components Analysis (PCA) where each of the Principal Components (PCs) are ranked by the variance, and relevancy selection, where the features are ranked by their MI with the target. The process starts at the top with the raw signal data, and moves downward through paths of feature extraction or selection. At the bottom, an evaluation of the resulting classification model is performed to provide a measure of the quality of the feature set produced. As an example, in the left-most path the signals are ranked by MI prior to feature extraction, which are then all input into the evaluation procedure. We refer to this particular path as MI-FE. Some paths are equivalent and are therefore omitted from our investigations. For instance, any path that has an MI stage followed by PCA is equivalent to performing solely PCA.

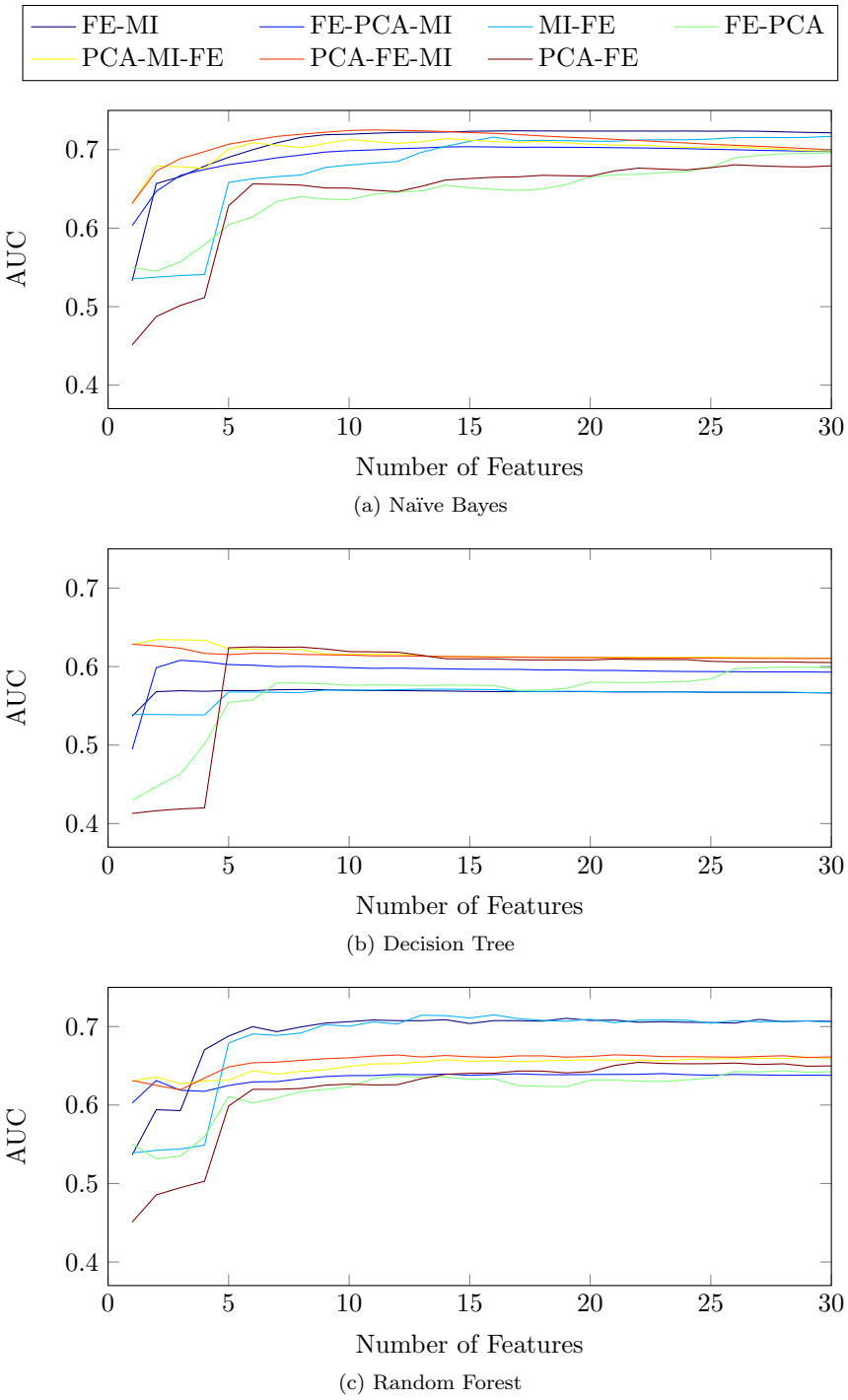
6.2.1 Classification and evaluation

Features selected by a selection path are evaluated using a random subset validation over sub-datasets. In each iteration of the subset validation, a random half of the datasets are used as training data and the other half is used as testing data. There are a total of $\binom{16}{8} = 12870$ possible train-test iterations over the sub-datasets, of which 200 are selected uniformly at random to be performed. The feature selection process is performed on each training data to rank the features. For computational reasons, the evaluation data is sub-sampled by a factor of 10 at this point. Thirty models are then built using different numbers of the ranked features, $(1, 2, \ldots, 30)$, and each are used to label the test dataset.

6.2.2 Results

The Area Under the Receiver Operator Characteristic Curve (AUC) performances for carriageway classification are shown in Figure 6.2, plotted against the number of selected features for the different selection paths. In general the selection paths containing a relevancy selection stage after a feature extraction stage had the highest AUC performances. The AUC performance of the MI-FE selection path was slightly lower than that of FE-MI, signifying that signal selection has a small classification performance cost. A redundancy selection stage did not increase AUC performance significantly, and only with the Decision Tree classifier did PCA-MI-FE, PCA-FE-MI and PCA-FE outperform FE-MI. The poor performance of the Decision Tree classifier with only a relevancy selection stage is likely due to it over-fitting to the highest ranked features, which Naïve Bayes and Random Forest are less affected by. This overfitting problem in relevancy feature selection was addressed in Chapter 4.

A similar pattern in the results is seen in the road classification AUC performances, shown in Figure 6.3. One difference is that the FE-MI and MI-FE selection paths no longer have the highest AUC performances with any classifier, indicating that redundancy is more of an issue with this classification task.

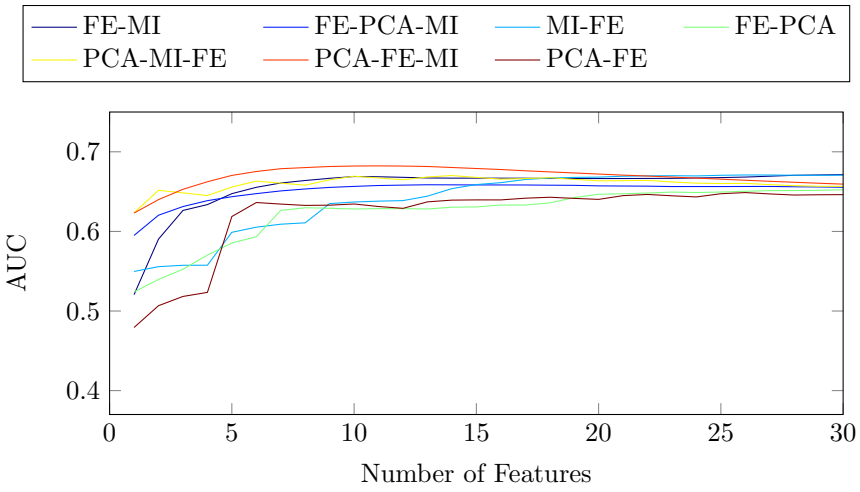


(a) Naïve Bayes

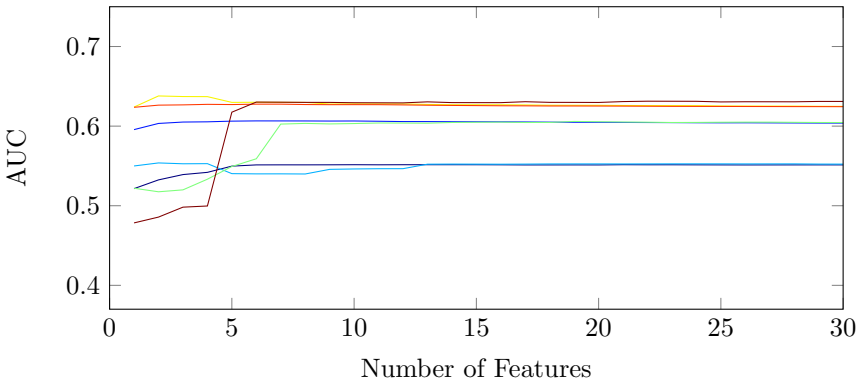
(b) Decision Tree

(c) Random Forest

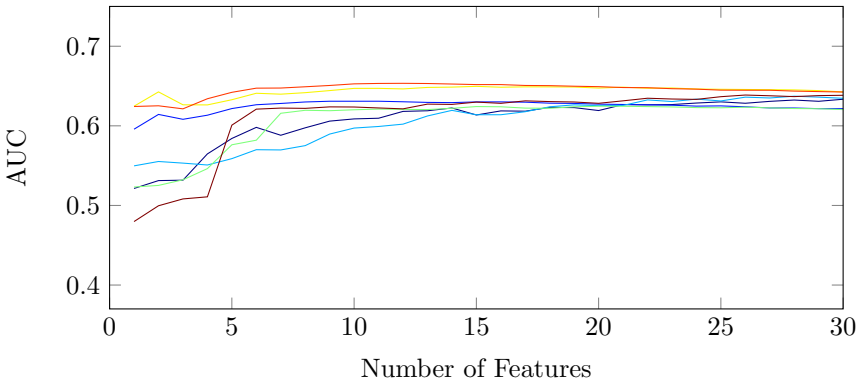
Figure 6.2: Carriageway classification AUC values against number of features used in the (a) Naïve Bayes, (b) Decision Tree and (c) Random Forest classifiers, when selecting features using the different selection paths.



(a) Naïve Bayes



(b) Decision Tree



(c) Random Forest

Figure 6.3: Road-type classification AUC values against number of features used in the (a) Naïve Bayes, (b) Decision Tree and (c) Random Forest classifiers when selecting features using the different selection paths.

Instead, the highest AUC performances are provided by performing an MI stage after a PCA stage, using either the PCA-FE-MI or PCA-MI-FE selection paths. The FE-PCA-MI selection path does not share this high performance, indicating that dealing with redundancy in the signals provides better features in this classification task. The paths containing no relevancy selection, PCA-FE and FE-PCA, again have lower AUC performances, especially for small numbers of features.

6.2.3 Discussion

In summary, these results provide several insights into the best avenues for a data mining approach to environment monitoring problems. They show that considering both redundancy and relevancy in a feature selection process will generally provide the highest performance. In fact, both are necessary for the highest performance in the road type classification task. One exception to this is with the Random Forest model used for the carriageway classification task, which performs best with features selected using only relevancy. Also, any redundancy analysis should be performed on the signals prior to feature extraction, and followed by a relevancy selection stage. Performing only redundancy feature selection does not provide a good feature ranking in any case, which is likely due to its unsupervised nature.

Also, the choice of methods may change depending on requirements of a system with respect to computing efficiency, rather than just predictive performance. For example, performing selection prior to feature extraction as in MI-FE is much less computationally expensive than selecting from the full feature set, while both methods will provide similar performance with 15 features. We find in general, however, that features selected using FE-MI or PCA-FE-MI provide higher AUC performances with fewer features than MI-FE or PCA-MI-FE. This result may be valuable where there is limit on the feasible number of signals that can be used in a model running on the vehicle's electronic control unit. In this case, it would also mean that any selection path including PCA

is unlikely to be of use, because the PCs produced are a linear combination of several inputs. We conclude therefore that FE-MI should be combined with some redundancy analysis other than PCA for the highest performing feature selection methods. Therefore, in the following sections we propose such a feature selection method that, using mRMR, is able to select good feature sets from large feature sets extracted from CAN-data.

6.3 Two stage feature selection

Redundancy in signal data can be considered as either intra-signal, between features extracted from within one signal, or inter-signal, between features extracted from different signals. For instance, in CAN-bus data there is unsurprisingly a large inter-signal redundancy between the features of *EngineSpeed* and *VehicleSpeed* signals. This is confirmed by the Pearson correlation between the raw values of the signals, of 0.94 in analysis of CoventryDMD. There may also be a high intra-signal feature redundancy, as with the minimum, mean and maximum features. This is particularly the case for these features when the temporal window is small and the signal is slowly varying.

Therefore, we propose a two step procedure to remove these intra-signal and inter-signal redundancies by considering them separately. In the first stage, feature selection is performed solely with extracted features from individual signals, aiming to remove intra-signal redundancies. In the second stage, selection is performed on these selected features as a whole, removing inter-signal redundancies. This then produces a final feature set with an expected minimal redundancy for use in a predictive model.

This two-stage process has benefits with regards to computation. For instance, the forward selection method of mRMR requires a great deal of computation with large feature sets. Moreover, large feature sets, such as those extracted from CAN-bus data, often do not fit into memory in their entirety, meaning that subsets of features have to be loaded from disk each time they

are processed. This not only lengthens the feature selection process, but also impacts on the complexity of the implementation. With our two-stage process these issues are avoided as smaller numbers of features are considered at a time and the majority of features are processed only once. Further, features that are processed twice will generally fit into memory together, meaning that each feature is loaded from disk only once.

In using this process, we expect there to be fewer redundancies in the final feature sets because redundancies are removed at both stages. However, returning fewer features in this first stage may reduce the relevance of the selected features to be used in learning. This will particularly be the case when many of the best performing features are from the same signal, but this is assumed to be an extreme and uncommon case.

6.3.1 Evaluation design

To evaluate the two-stage feature selection two types of feature ranking method are considered. The first ranks features by their relevancy only, and the second uses mRMR to consider feature redundancies also. For both relevancy and redundancy computation, we use MI, Symmetrical Uncertainty (SU), HSIC and permutation methods. For relevancy, the features are ranked using Z_{MI} as described in Chapter 4, and $PmRMR$ as introduced in Chapter 5 is used when redundancies are considered. Here, 1000 permutations are used in each case, and block sizes of 3000 are used for the RCD, 1500 for the CoventryDMD and WarwickDMD, and 1000 for the OPPORTUNITY Activity Recognition Dataset (OARD).

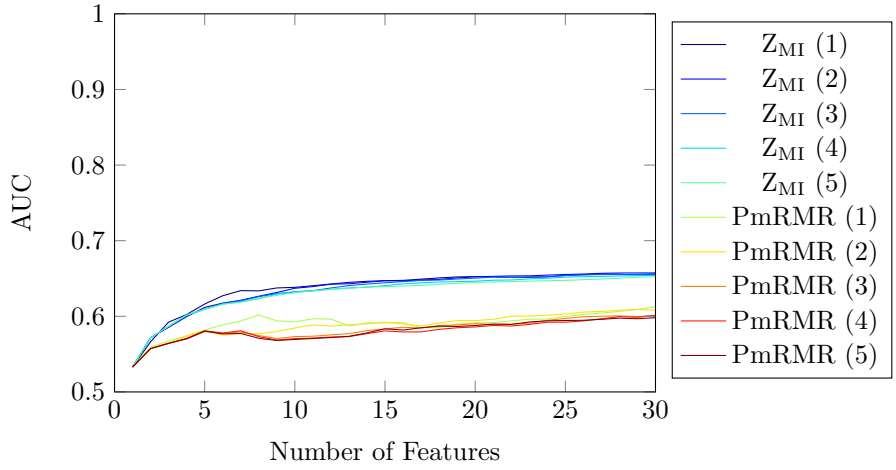
The evaluation structure is the same as for the evaluations in Chapter 4, with twenty train-test iterations and 40% of each journey being used as training data. During the first selection stages, 1, 2, 3, 4, or 5 features are selected from each signal using the training data. These are then combined for the second stage, where a feature ranking is produced and between one and thirty features are selected from the top of the ranking. The selection algorithm used in the

first stage is always the same as the one used in the second stage.

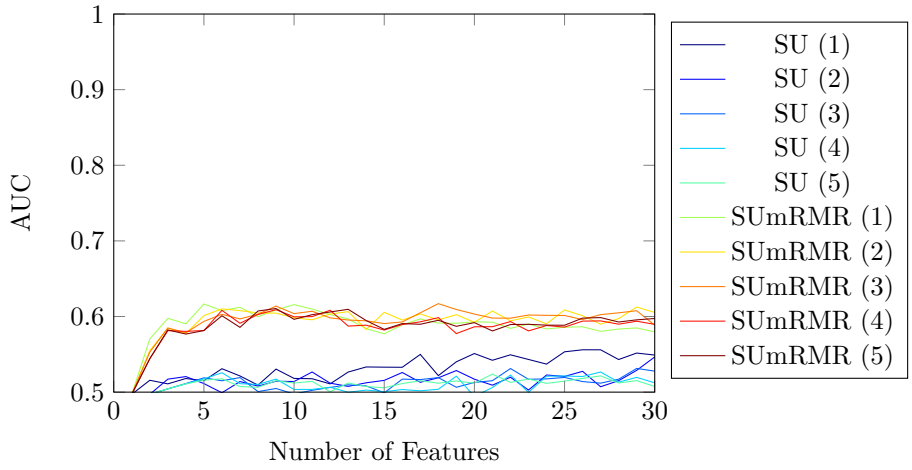
The feature sets are then evaluated using classification algorithms and by inspecting the redundancy levels in selected feature sets. For classification evaluations, the Naïve Bayes, Random Forest and Multilayer Perceptron learning algorithms are used, as implemented in WEKA. The predictions from all train-test iterations are then combined to produce an overall AUC. Redundancy levels are measured by computing the total redundancy in the selected features as in Equation 5.2, with both PC_{MI} in place of $Cor(\cdot)$.

6.3.2 Results

The AUC performances of features selected using (a) Z_{MI} and $PmRMR$ and (b) SU and $SUmRMR$ are shown in Figures 6.4 (for the CoventryDMD), 6.5 (for the RCD with carriageway labelling), 6.6 (for the RCD with road labelling) and 6.7 (for the OARD). Where features were selected with permutation methods, Naïve Bayes was used for classification, and Multilayer Perceptron was used for features selected using SU and $SUmRMR$. These results are representative of the other feature ranking methods, and classifiers including Decision Trees and Random Forests. In general the performance decreased as more features were selected per signal. This was expected, as the feature set in the second stage is larger and contains more redundancy when higher numbers feature per signal are selected in the first stage. It was also expected that the mRMR methods would outperform relevancy only selection, which was true for the majority of cases. For the RCD with road labelling, there was no difference in AUC performance between SU and $SUmRMR$, although $PmRMR$ did outperform Z_{MI}. Also, Z_{MI} outperformed $PmRMR$ for the CoventryDMD, the RCD with road labelling and the OARD. The performances of HSIC and $HSICmRMR$ were more comparable, and only when one feature per signal was selected in the first stage did HSIC have the higher AUC. As in Chapter 4, the WarwickDMD is omitted from these results as performances were low in general, and this is discussed further in Section 6.4.

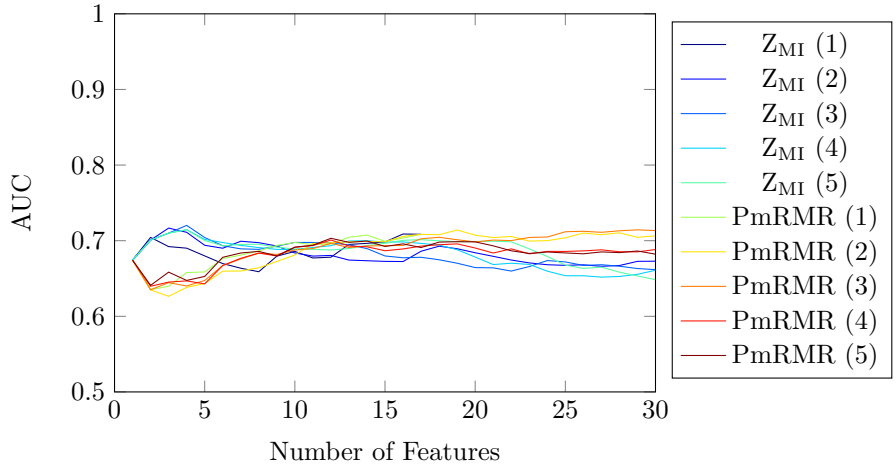


(a) Naïve Bayes

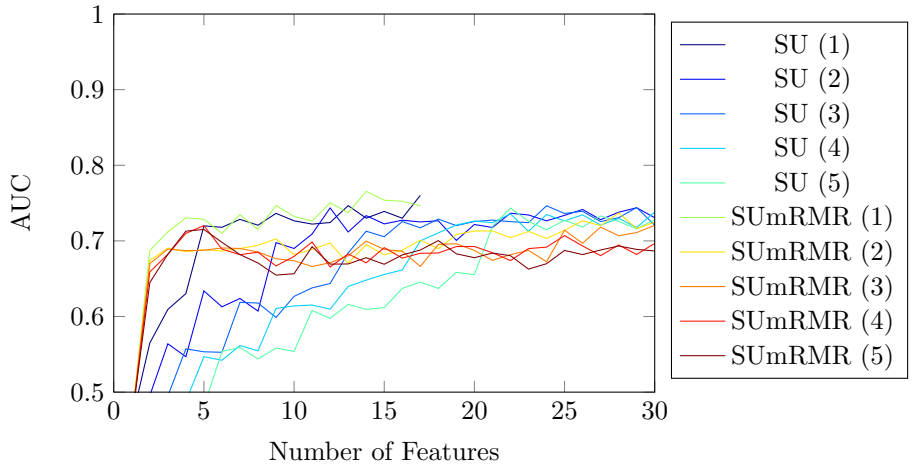


(b) Multilayer Perceptron

Figure 6.4: AUC performances when between one and five features were selected in the first stage from the CoventryDMD using (a) Z_{MI} and $PmRMR$ (with classification performed by Naïve Bayes) and (b) SU and $SUmRMR$ (with classification performed by Multilayer Perceptron).

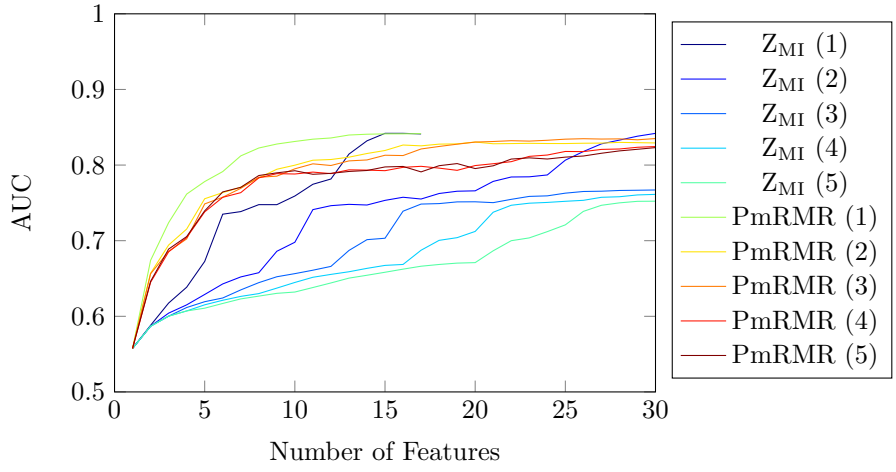


(a) Naïve Bayes

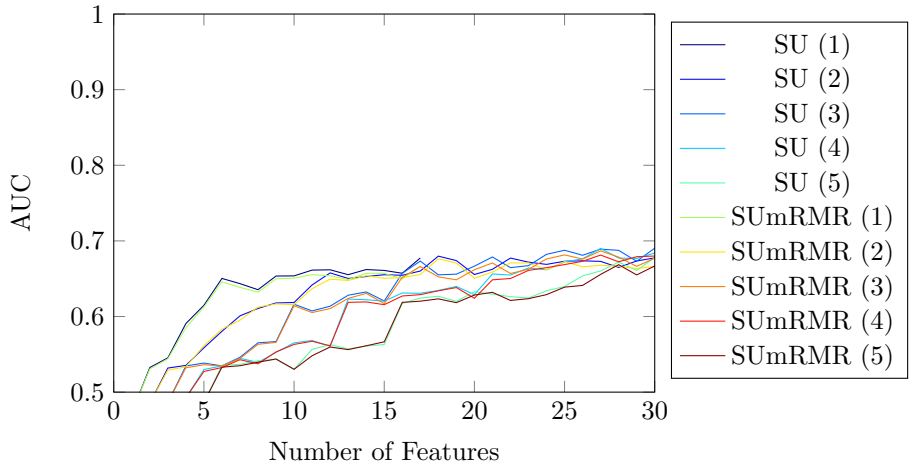


(b) Multilayer Perceptron

Figure 6.5: AUC performances when between one and five features were selected in the first stage from the RCD with carriageway labelling using (a) Z_{MI} and $PmRMR$ (with classification performed by Naïve Bayes) and (b) SU and $SUmRMR$ (with classification performed by Multilayer Perceptron).

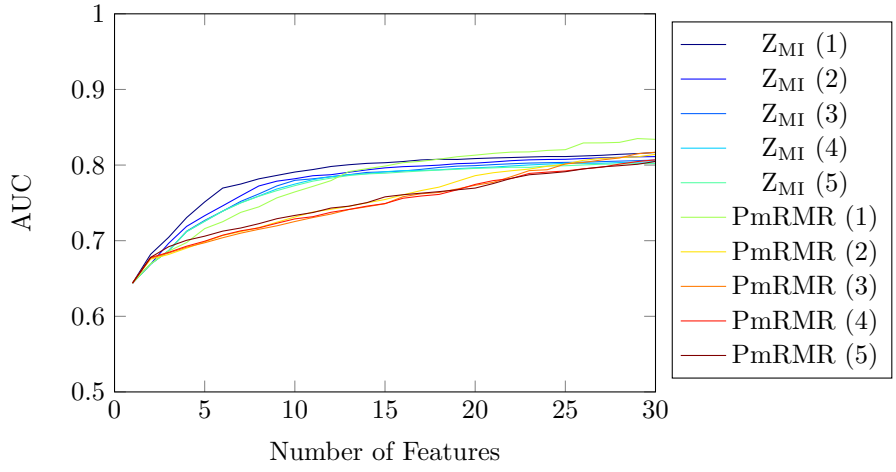


(a) Naïve Bayes

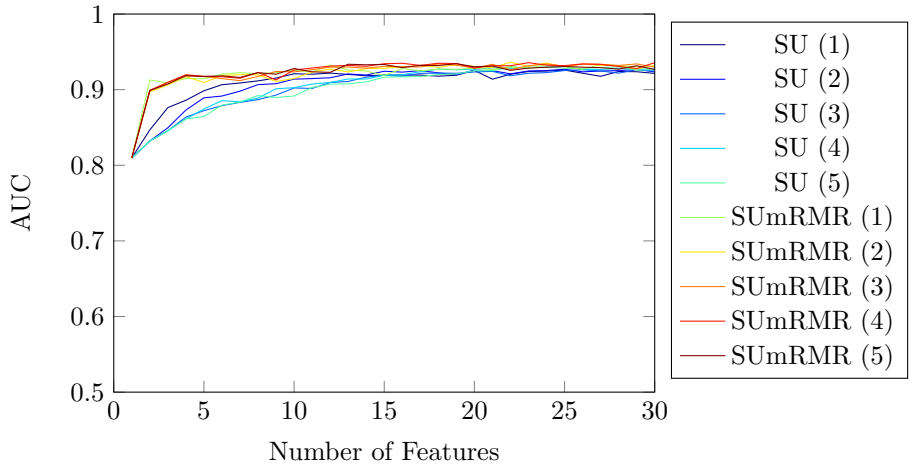


(b) Multilayer Perceptron

Figure 6.6: AUC performances when between one and five features were selected in the first stage from the RCD with road labelling using (a) Z_{MI} and $PmRMR$ (with classification performed by Naïve Bayes) and (b) SU and $SUmRMR$ (with classification performed by Multilayer Perceptron).

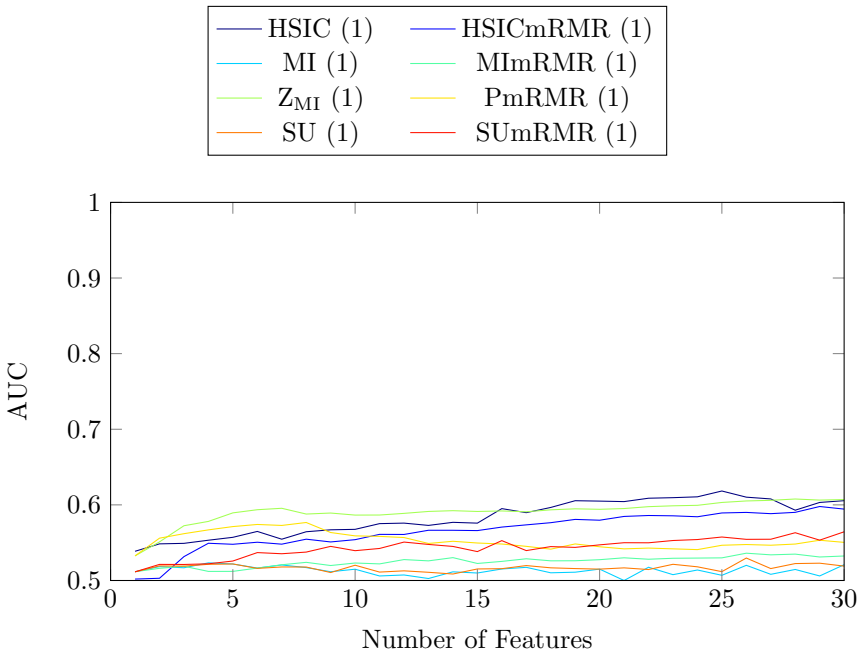


(a) Naïve Bayes

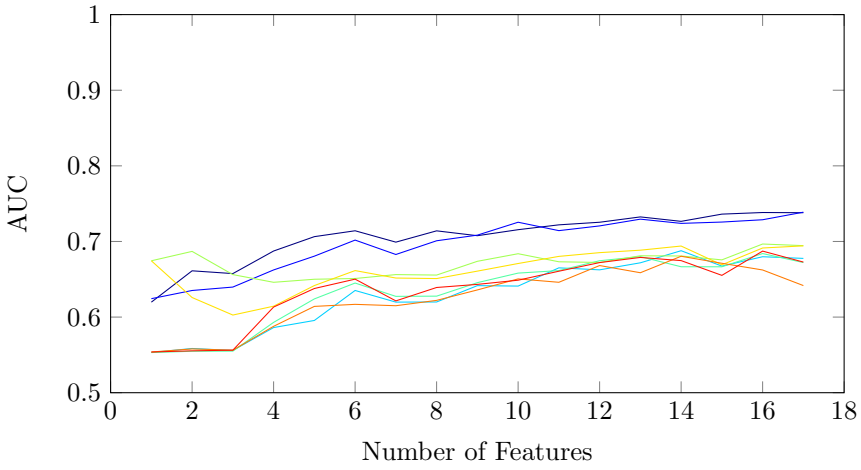


(b) Multilayer Perceptron

Figure 6.7: AUC performances when between one and five features were selected in the first stage from the OARD using (a) Z_{MI} and $PmRMR$ (with classification performed by Naïve Bayes) and (b) SU and $SUmRMR$ (with classification performed by Multilayer Perceptron).

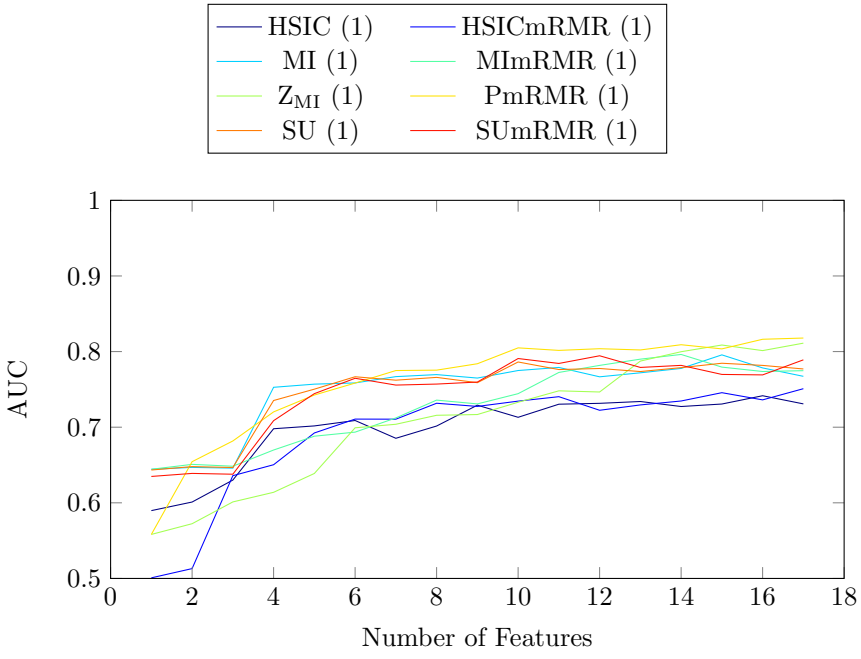


(a) CoventryDMD

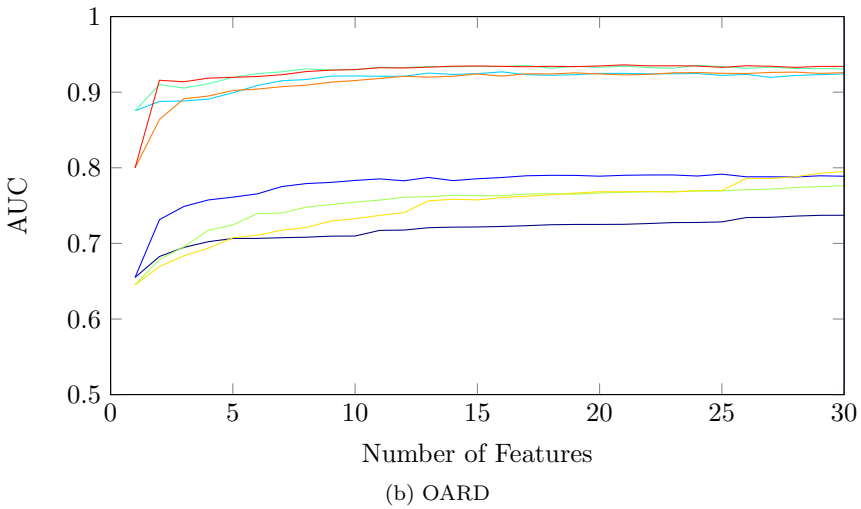


(b) RCD-Carriageway

Figure 6.8: AUC performances of the Random Forest classifier for (a) the CoventryDMD and (b) RCD with carriageway labelling when features were selected using the eight methods and one feature per signal was selected in the first stage.



(a) RCD-Road



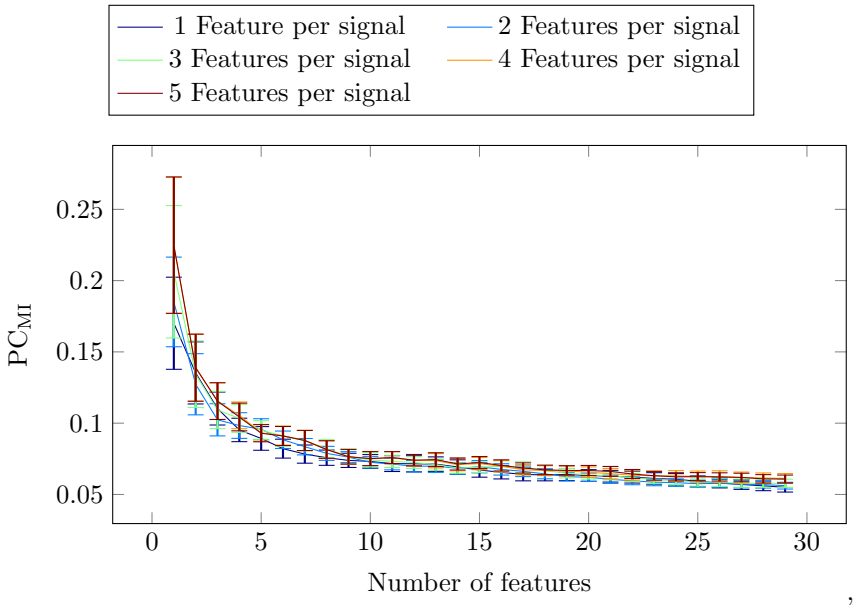
(b) OARD

Figure 6.9: AUC performances of the Random Forest classifier for (a) the RCD with road labelling and (b) the OARD when features were selected using the eight methods and one feature per signal was selected in the first stage.

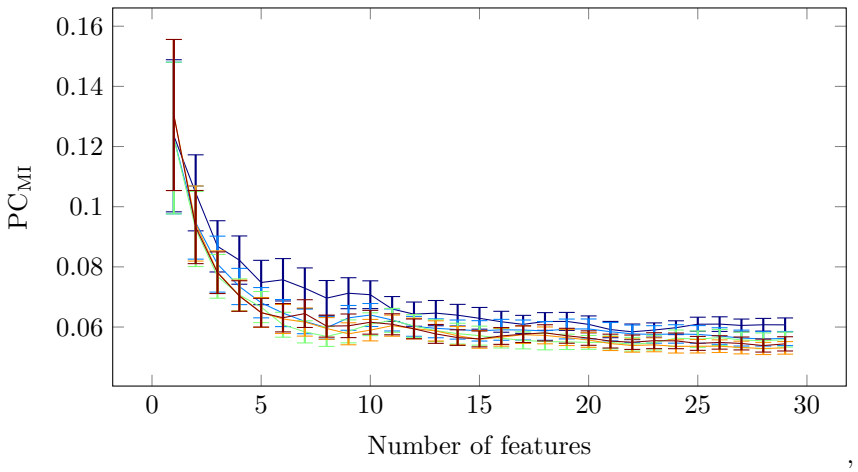
To compare the feature ranking methods used with the two-stage selection process, Figures 6.8 and 6.9 show the AUC performances of each when only one feature per signal was chosen in the first stage from the four datasets. In all cases presented the Random Forest classifier was used, but trends with the other learning algorithms are similar. Ranking features by Z_{MI} had the highest AUC performance with small numbers of features for the CoventryDMD and RCD with road labelling. In the RCD with carriageway labelling, HSIC and $HSICmRMR$ outperformed the other ranking methods with four or more features, and the permutation approaches had higher performance than MI, $MImRMR$, SU and $SUmRMR$.

The AUC performances of the permutation and HSIC approaches for the OARD were much lower than the other methods. These methods gave high rankings to some features with high MI relevancy, that provided the best predictive performance for the dataset. This signifies that the bias present in the vehicle telemetry datasets is not as pronounced as with the OARD, and the permutation or HSIC approaches are not appropriate.

Finally, we inspect feature redundancies after selecting features using the two-stage selection method. Figures 6.10 (for the CoventryDMD), 6.11 (for the RCD with carriageway labelling), 6.12 (for the RCD with road labelling) and 6.13 (for the OARD) show redundancies when selecting different numbers of features using the two-stage process. The redundancy was measured by the mean PC_{MI} between each pair of features, as in Equation 5.2. The results presented are the mean redundancies computed using the training data over the twenty train-test iterations, and the error bars show their standard error. It was expected that the redundancy would be smaller if fewer features were selected from each signal. In fact, however, there was little difference in selecting between one and five features per signal in most cases. In some cases, as with features selected using $PmRMR$ from the RCD or OARD the PC_{MI} redundancy was higher when one feature per signal was selected than in the other cases. Only with features selected using $SUmRMR$ from the RCD with

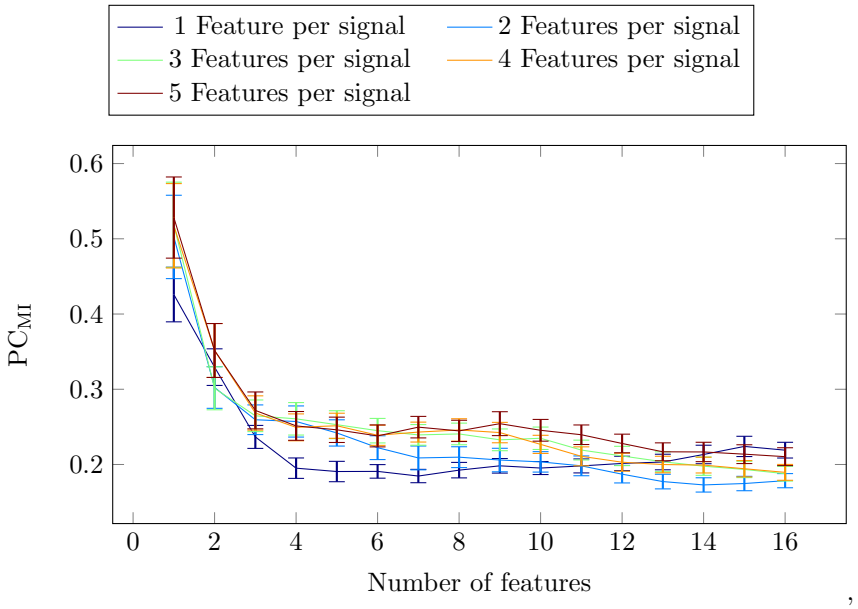


(a) $SUmRMR$

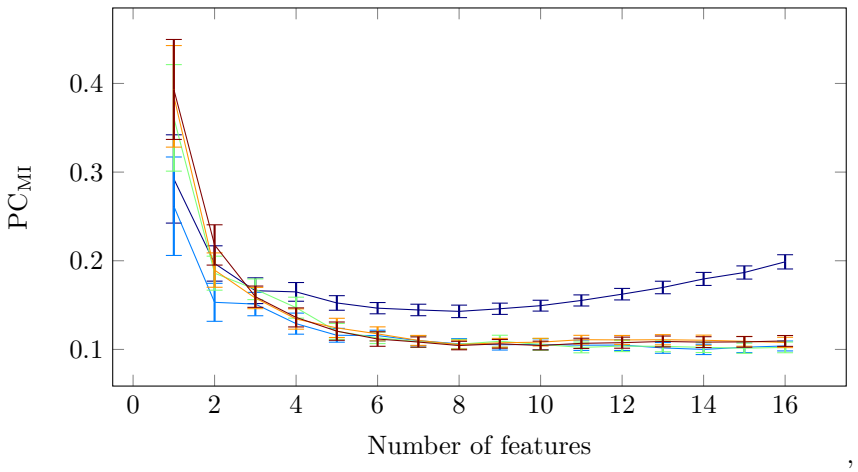


(b) $PmRMR$

Figure 6.10: Redundancies measured in PC_{MI} for different numbers of features selected from the CoventryDMD using (a) $SUmRMR$ and (b) $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage.

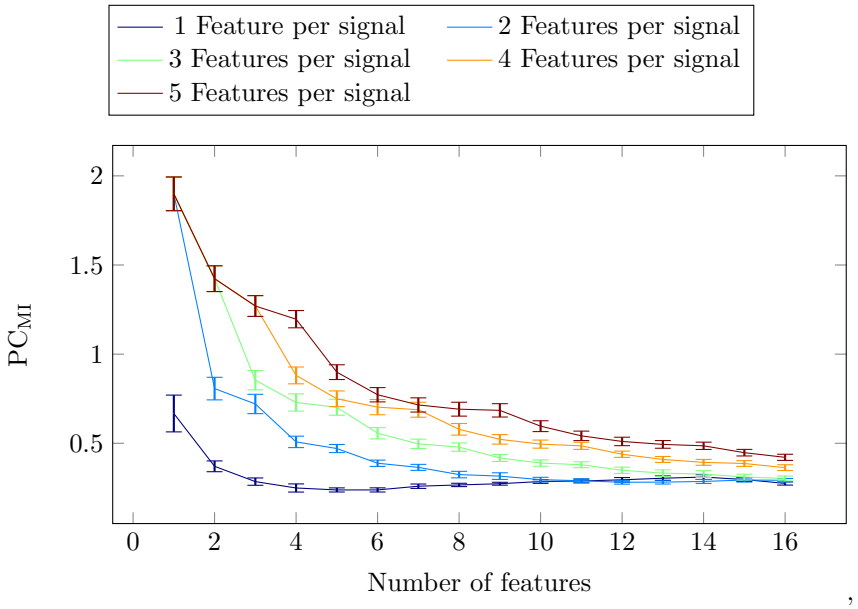


(a) $SUmRMR$

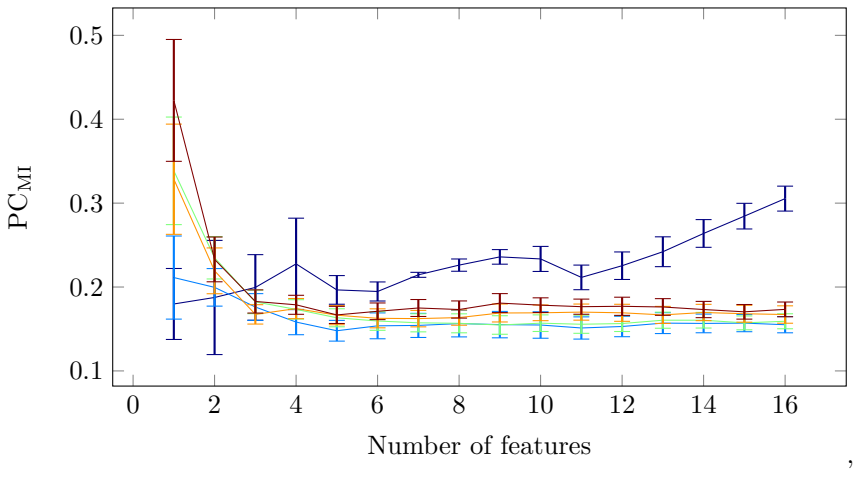


(b) $PmRMR$

Figure 6.11: Redundancies measured in PC_{MI} for different numbers of features selected from the RCD with carriageway labelling using (a) $SUmRMR$ and (b) $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage.

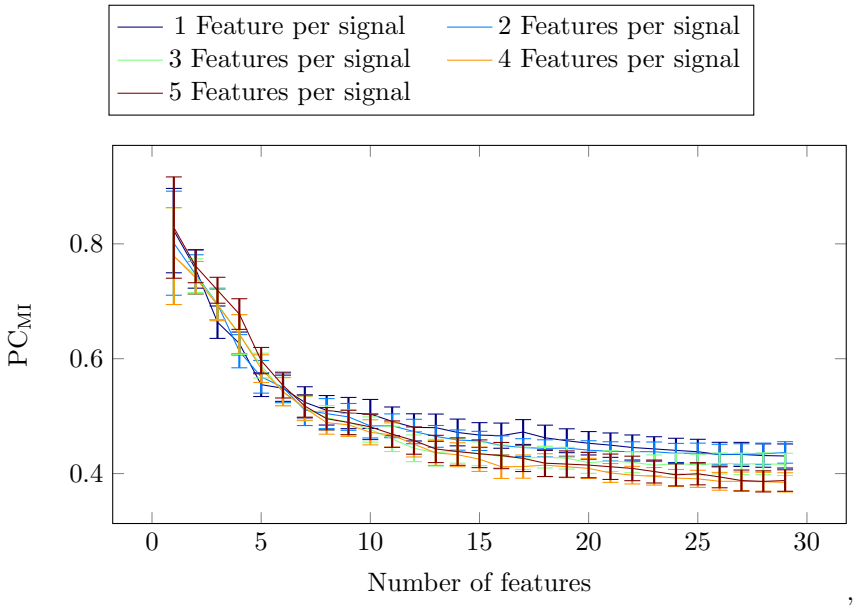


(a) $SUmRMR$

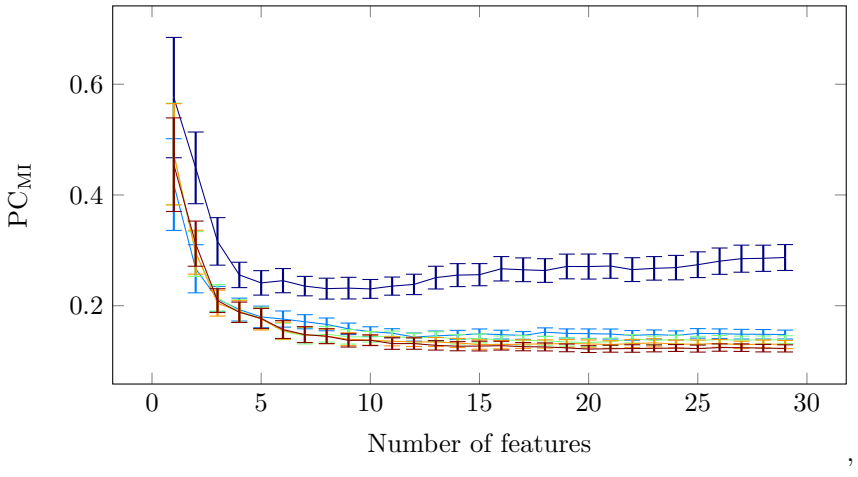


(b) $PmRMR$

Figure 6.12: Redundancies measured in PC_{MI} for different numbers of features selected from the RCD with road labelling using (a) $SUmRMR$ and (b) $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage.



(a) $SUmRMR$



(b) $PmRMR$

Figure 6.13: Redundancies measured in PC_{MI} for different numbers of features selected from the OARD using (a) $SUmRMR$ and (b) $PmRMR$ in a two-stage selection process with between one and five features selected in the first stage.

carriageway labelling and the OARD were the redundancies lower when fewer features selected per signal. These results may be explained by the relevancy of the selected features, as features of higher relevance to one variable often have higher relevance to others, due to their higher entropy and variance levels.

Table 6.1 compares the redundancy in feature sets when selecting ten features with the different feature ranking methods, and is again measured in PC_{MI}. In each case, the redundancy of all thirty selected features is presented, averaged over the twenty train-test iterations. The numbers shown in braces are the standard errors. In the majority of cases where a relevancy only selection method was used, including MI, SU, Z_{MI} and HSIC, the redundancy increased as expected with the number of features selected per signal. The HSIC method with the CoventryDMD had higher redundancy when only one feature was selected per signal than in other cases. In cases where mRMR was employed the redundancies were lower than their relevancy only counterparts. The redundancy also either did not change or decreased slightly as the number of features selected per signal increased. Finally, $PmRMR$ tended to select features with the lowest redundancy in general.

6.4 DMD analysis

In Chapter 4 and earlier in this chapter, we noted that AUC performances in evaluations with the WarwickDMD were very poor. To investigate this further we performed an evaluation for models built with data of different combinations of drivers. Here, three train-test iterations were defined by the activities being performed in the data. For instance, the first iteration had the training data made up of the baseline period and first two secondary task and recovery periods. In the second, the training data was made of the baseline period and the second two task and recovery periods. The training data in the third iteration was made up of the baseline period and the first and last task recovery periods. In each case, the remainder of the samples were used as testing data, meaning that

Selection	1		2		3		4		5	
MI	0.230	(0.019)	0.317	(0.027)	0.399	(0.034)	0.439	(0.026)	0.519	(0.021)
MImRMR	0.105	(0.008)	0.120	(0.012)	0.127	(0.011)	0.131	(0.011)	0.130	(0.010)
SU	0.220	(0.019)	0.294	(0.022)	0.354	(0.022)	0.421	(0.018)	0.509	(0.012)
SUmRMR	0.073	(0.005)	0.073	(0.004)	0.074	(0.005)	0.075	(0.005)	0.075	(0.005)
Z_{MI}	0.371	(0.021)	0.464	(0.035)	0.550	(0.044)	0.601	(0.052)	0.623	(0.050)
PmRMR	0.071	(0.005)	0.064	(0.004)	**0.061**	(0.004)	**0.059**	(0.004)	**0.062**	(0.004)
HSIC	0.578	(0.053)	0.138	(0.010)	0.202	(0.020)	0.278	(0.023)	0.340	(0.016)
HSICmRMR	**0.050**	(0.003)	**0.049**	(0.003)	0.075	(0.005)	0.095	(0.009)	0.116	(0.014)

(a) CoventryDMD

Selection	1		2		3		4		5	
MI	0.259	(0.007)	0.302	(0.010)	0.414	(0.014)	0.521	(0.016)	0.606	(0.016)
MImRMR	0.221	(0.007)	0.202	(0.012)	0.239	(0.015)	0.284	(0.014)	0.302	(0.014)
SU	0.247	(0.010)	0.302	(0.011)	0.412	(0.015)	0.523	(0.017)	0.598	(0.017)
SUmRMR	0.195	(0.009)	0.203	(0.014)	0.235	(0.015)	0.227	(0.012)	0.245	(0.014)
Z_{MI}	0.309	(0.018)	0.443	(0.019)	0.558	(0.019)	0.626	(0.021)	0.640	(0.021)
PmRMR	**0.149**	(0.006)	**0.106**	(0.006)	**0.106**	(0.007)	**0.108**	(0.004)	**0.104**	(0.005)
HSIC	0.369	(0.024)	0.355	(0.010)	0.423	(0.015)	0.492	(0.017)	0.533	(0.012)
HSICmRMR	0.255	(0.008)	0.265	(0.011)	0.264	(0.011)	0.266	(0.015)	0.293	(0.017)

(b) RCD-carriageway

Selection	1		2		3		4		5	
MI	0.281	(0.009)	0.289	(0.010)	0.390	(0.011)	0.511	(0.016)	0.608	(0.023)
MImRMR	0.243	(0.011)	0.225	(0.009)	0.260	(0.010)	0.276	(0.017)	0.311	(0.020)
SU	0.292	(0.010)	0.304	(0.014)	0.406	(0.015)	0.517	(0.017)	0.623	(0.025)
SUmRMR	0.285	(0.010)	0.296	(0.013)	0.389	(0.018)	0.495	(0.023)	0.596	(0.030)
Z_{MI}	0.491	(0.016)	0.632	(0.019)	0.744	(0.011)	0.844	(0.011)	0.901	(0.020)
PmRMR	**0.233**	(0.015)	**0.155**	(0.016)	**0.157**	(0.010)	**0.169**	(0.009)	**0.178**	(0.009)
HSIC	0.354	(0.013)	0.361	(0.012)	0.419	(0.011)	0.486	(0.012)	0.534	(0.012)
HSICmRMR	0.294	(0.011)	0.300	(0.008)	0.302	(0.012)	0.325	(0.014)	0.362	(0.013)

(c) RCD-road

Selection	1		2		3		4		5	
MI	0.679	(0.028)	0.705	(0.025)	0.728	(0.021)	0.759	(0.020)	0.804	(0.020)
MImRMR	0.567	(0.027)	0.599	(0.030)	0.596	(0.026)	0.589	(0.028)	0.591	(0.028)
SU	0.710	(0.032)	0.756	(0.022)	0.782	(0.020)	0.799	(0.021)	0.825	(0.019)
SUmRMR	0.504	(0.025)	0.483	(0.021)	0.477	(0.022)	0.472	(0.022)	0.481	(0.022)
Z_{MI}	0.920	(0.012)	0.921	(0.015)	0.925	(0.016)	0.927	(0.016)	0.928	(0.016)
PmRMR	**0.230**	(0.017)	**0.153**	(0.009)	**0.144**	(0.010)	**0.138**	(0.009)	**0.138**	(0.010)
HSIC	0.711	(0.026)	0.757	(0.033)	0.774	(0.035)	0.789	(0.034)	0.820	(0.030)
HSICmRMR	0.311	(0.024)	0.318	(0.020)	0.347	(0.023)	0.352	(0.022)	0.371	(0.023)

(d) OARD

Table 6.1: Redundancy for the eight selection algorithms measured by SU of top ten features selected from the (a) CoventryDMD, (b) RCD-carriageway, (c) RCD-road, and (d) OARD datasets when between one and five features were selected per signal in the first stage. Standard error is shown in braces.

Rank	Signal	Feature	Window (s)
1	SteeringWheelAngle	Third derivative	0.5
2	BrakePressure	DFT 3	0.5
3	WheelSpeedReR	Third derivative	0.2
4	BrakeSwitchStatus	DFT 5	2.5
5	EPBLongitudinalAcc	DFT 5	1
6	YawRate	DFT 5	5
7	BrakePressureComp	DFT 14	1
8	SteeringWheelAngle	First derivative	2.5
9	SteeringWheelAngleSpeed	DFT 5	1
10	TorqConvStatus	DFT 7	2.5
11	VirtualPedalPosition	DFT 5	1
12	EngineTorqArbitratedModified	Max value	5
13	LateralAcceleration	First derivative	5
14	SuspensionHeightFR	Gradient Up/0/Down	0.2
15	ACCCancelRequest	Convexity	0.5
16	SuspensionHeightRR	DFT 5	2.5
17	TMPedalPos	First derivative	2.5
18	SuspensionHeightFL	DFT 1	5
19	YawRate	DFT 4	2.5
20	WheelSpeedFrR	Second derivative	1

Table 6.2: WarwickDMD features ranked by $PmRMR$ in a two-stage process with one feature selected per signal in the first stage. The ranking was produced from training data made up of the baseline period and first two secondary task and recovery periods of all thirteen drivers.

in each iteration one secondary task period and one recovery period was used to evaluate the model.

To both minimise computation time in these evaluations, and so that the same features were used in each train-test iteration, features were selected from training data of the full set of drivers. The feature set was also reduced prior to automatic selection by removing signals that were known to be irrelevant, such as those known to change value very rarely. Where it was unclear if the feature may be relevant it was kept, and redundancy was not considered in this manual selection. The features selected for use in the first of the three train-test iterations are presented in Table 6.2. Several of the features selected are directly linked to the driver, such as the steering wheel angles and pedal positions. Others are related to the vehicle behaviour, such as wheel rotation speeds, suspension movements and yaw rates. Although these features are not directly related to the driver, they are a representation of how the car was driven, including the speed and turning behaviour.

Using these features, models were then built on the same training data for all combinations of between one and five drivers, and tested on the remaining samples of each driver individually. For example, in the combination with three drivers, 1, 2, and 3, a model was built using their combined training datasets, and tested using the remainder of samples individually for each driver. The predictions made during the testing stage for each driver were then used to produce an individual AUC performance for each of these drivers. Table 6.3 shows the mean AUC performances for each driver with a Naïve Bayes classifier built with data from between 1 and five drivers to predict (a) the distraction status and (b) an increase in heart rate of 5 beats per minute (bpm) or more. The standard error of the AUC values are shown in braces. The AUC performances of several drivers are still very low, and models were unable to predict their distraction statuses or heart rate increases better than random, which would produce AUCs of 0.5.

The models were able to predict the distraction status with moderate success

of four drivers, 2, 4, 8, and 12 in these evaluations with models built using their own data. These drivers were all between 18 and 25 years old and drove less than 5000 miles per year in small vehicles. All of these drivers had fewer than eight years driving experience and three of them had fewer than six. Models built to predict an increase in heart rate had moderate success with a different set of drivers, 3, 4, 9, and 10. For drivers 1, 2, 5 and 11 had performances that were much worse than random. This is unexpected, as the model is predicting the opposite of the ground truth consistently.

In both the classification tasks, AUC performances decreased as the models were built with data from more drivers. For instance, performance in the distraction status task decreased from 0.576 when the model was built on driver 8 only, to 0.510 when it was built on driver 8 and four other drivers. If the model had poor performance even when built with data from one driver, the performance generally tended towards 0.5 as data from other drivers was used. This may indicate that models for driver distraction should be specific to the driver as well as the vehicle, as was reported by Jo et al. [68] when modelling driver fatigue.

To investigate if the difficulty of the secondary task had any effect on the performance of models, we inspect the individual folds in the train-test iteration. Tables 6.4 show the AUC performances models built with data from individual drivers over three train-test iterations. In each, the training data is again made up of data from the baseline periods, and two secondary tasks plus their subsequent recovery periods. The remaining samples from the other secondary task period and its associated recovery period make up the testing data. In some cases there were no significant increase in heart rate for a driver present the testing data. In these cases the AUC undefined, and this is signified in the table as *na* and are ignored from the mean calculations. The models tested on data from the 2-back task had slightly higher performance than those tested on the other two on average for both distraction status and increase in heart rate, which was expected. However, we found no consistent difference

Driver	1 driver	2 drivers	3 drivers	4 drivers	5 drivers
1	0.521 (na)	0.504 (0.013)	0.504 (0.005)	0.501 (0.003)	0.506 (0.002)
2	0.558 (na)	0.537 (0.013)	0.542 (0.006)	0.543 (0.003)	0.542 (0.002)
3	0.518 (na)	0.518 (0.006)	0.528 (0.002)	0.533 (0.001)	0.536 (0.001)
4	0.555 (na)	0.540 (0.009)	0.543 (0.004)	0.538 (0.002)	0.538 (0.001)
5	0.448 (na)	0.472 (0.007)	0.494 (0.004)	0.507 (0.002)	0.515 (0.001)
6	0.482 (na)	0.495 (0.015)	0.483 (0.006)	0.487 (0.003)	0.486 (0.002)
7	0.511 (na)	0.488 (0.010)	0.501 (0.004)	0.499 (0.002)	0.498 (0.001)
8	0.576 (na)	0.531 (0.007)	0.521 (0.004)	0.514 (0.002)	0.510 (0.001)
9	0.454 (na)	0.473 (0.007)	0.470 (0.003)	0.467 (0.002)	0.467 (0.001)
10	0.454 (na)	0.483 (0.006)	0.493 (0.003)	0.495 (0.002)	0.497 (0.001)
11	0.433 (na)	0.468 (0.013)	0.476 (0.004)	0.474 (0.002)	0.474 (0.001)
12	0.563 (na)	0.563 (0.009)	0.561 (0.004)	0.564 (0.002)	0.571 (0.001)
13	0.467 (na)	0.485 (0.009)	0.484 (0.005)	0.484 (0.003)	0.488 (0.002)

(a) Distraction status

Driver	1 driver	2 drivers	3 drivers	4 drivers	5 drivers
1	0.328 (na)	0.343 (0.004)	0.344 (0.002)	0.348 (0.001)	0.351 (0.001)
2	0.316 (na)	0.352 (0.021)	0.353 (0.009)	0.351 (0.004)	0.350 (0.003)
3	0.520 (na)	0.538 (0.003)	0.543 (0.003)	0.541 (0.002)	0.540 (0.001)
4	0.581 (na)	0.594 (0.005)	0.596 (0.002)	0.600 (0.001)	0.603 (0.001)
5	0.309 (na)	0.328 (0.008)	0.344 (0.003)	0.349 (0.001)	0.353 (0.001)
6	0.507 (na)	0.526 (0.009)	0.533 (0.004)	0.541 (0.002)	0.547 (0.001)
7	0.494 (na)	0.480 (0.008)	0.470 (0.004)	0.465 (0.002)	0.461 (0.001)
8	0.485 (na)	0.492 (0.004)	0.500 (0.002)	0.505 (0.001)	0.509 (0.001)
9	0.542 (na)	0.521 (0.008)	0.516 (0.003)	0.522 (0.002)	0.525 (0.001)
10	0.709 (na)	0.705 (0.007)	0.694 (0.002)	0.688 (0.001)	0.684 (0.001)
11	0.334 (na)	0.340 (0.006)	0.330 (0.003)	0.328 (0.001)	0.326 (0.001)
12	0.468 (na)	0.495 (0.011)	0.507 (0.007)	0.510 (0.004)	0.513 (0.002)
13	0.460 (na)	0.474 (0.006)	0.474 (0.002)	0.477 (0.001)	0.477 (0.001)

(b) Heart rate increase (5 bpm)

Table 6.3: Mean AUC performances when building models for different combinations of drivers and testing on individual driver data for (a) the distraction status (*normal* or *distracted*) and (b) a increase in heart rate (*baseline* or *increase by 5 bpm*). The standard error is presented in the braces after each AUC value.

in performance of models evaluated on data from the different task difficulties. Models with drivers 3, 4 and 8, performed significantly better in the distraction status classification when tested on the 0-back task than for the other two tasks, while drivers 6, 7, and 13 had higher AUC performances for the others tasks. In predicting distraction status, driver 2 showed more consistent performance across the tasks and in predicting an increase in heart rate driver 10 showed best performance across the train-test iterations.

There was also no effect in the order that the tasks were presented to the drivers for the performance of models predicting distraction status. The mean AUC performance of these models for predicting the first presented task was 0.516, while for the second and third presented tasks it was 0.469 and 0.524 respectively. In predicting heart rate the AUC performances for the first, second and third tasks were 0.514, 0.515 and 0.530 respectively, which can be viewed as decreasing in performance for tasks later in the trial. However, because the AUC values are less than 0.5 the classifiers are very close to or worse than that of a random model, they cannot be used as basis for conclusions.

6.5 Conclusions

In this chapter we have investigated using a known redundancy structure in features extracted from signal data to both speed up and increase performance of the feature selection process. We first considered signal selection prior to feature extraction and found the highest performances when MI was used to select features extracted from PCs of the signals. This approach, however, is not suitable as the PCs are often a linear combination of several signals and features may therefore be too complex to process in a vehicle.

Concluding that redundancy analysis is likely to be required for the highest performance, we then employed mRMR in a two-stage feature selection process. This two-stage selection process first selected a number of features extracted from signals individually, before combining them for a final selection stage. We

Driver	0-back	1-back	2-back
1	0.482 (3)	0.606 (1)	0.475 (2)
2	0.552 (2)	0.568 (3)	0.553 (1)
3	0.539 (1)	0.515 (3)	0.499 (2)
4	0.542 (1)	0.499 (2)	0.623 (3)
5	0.496 (3)	0.365 (2)	0.483 (1)
6	0.424 (1)	0.467 (3)	0.555 (2)
7	0.343 (2)	0.595 (1)	0.595 (3)
8	0.653 (1)	0.621 (2)	0.454 (3)
9	0.496 (3)	0.400 (2)	0.464 (1)
10	0.443 (2)	0.422 (1)	0.497 (3)
11	0.504 (1)	0.351 (3)	0.444 (2)
12	0.581 (2)	0.476 (1)	0.631 (3)
13	0.450 (1)	0.316 (2)	0.635 (3)
Mean	0.500	0.477	0.531

(a) Distraction status

Driver	0-back	1-back	2-back
1	na (3)	0.423 (1)	0.562 (2)
2	na (2)	0.448 (3)	0.500 (1)
3	0.495 (1)	0.557 (3)	0.508 (2)
4	0.498 (1)	0.569 (2)	0.677 (3)
5	na (3)	0.405 (2)	0.521 (1)
6	0.463 (1)	0.555 (3)	0.504 (2)
7	0.410 (2)	0.646 (1)	0.427 (3)
8	0.519 (1)	0.505 (2)	0.432 (3)
9	0.520 (3)	0.602 (2)	0.503 (1)
10	0.747 (2)	0.683 (1)	0.697 (3)
11	0.498 (1)	na (3)	0.505 (2)
12	0.366 (2)	0.499 (1)	0.540 (3)
13	0.439 (1)	0.494 (2)	0.448 (3)
Mean	0.495	0.532	0.525

(b) Heart rate increase (5 bpm)

Table 6.4: AUC performances for each train-test iteration with data from individual drivers to predict (a) the distraction status (*normal* or *distracted*) and (b) an increase in heart rate (*baseline* or *increase by 5bpm*). The numbers in braces indicate the position of that task in the trial, i.e. driver 1 performed the 1-back task, followed by the 2-back and then the 0-back.

compared eight feature selection methods, four that ranked features by their relevancy (namely MI, SU, HSIC and Z_{MI}), and four that used mRMR to consider feature redundancy in both selection stages (namely $MImRMR$, $SUmRMR$, $HSICmRMR$, and $PmRMR$). We found that this two-stage feature selection process performed best in classification evaluations when one feature was selected per signal in the first stage, and performance decreased in general as more features were considered. Features selected from the OARD using permutation methods had particularly poor AUC performances compared to using the other selection methods, which had very similar performance. For other datasets, and particularly the CoventryDMD which contains the largest number of signals and most bias, the AUC performance of Z_{MI} and $PmRMR$ was more comparable to the other methods and better for small numbers of features. The HSIC methods had highest AUC performances for the RCD with carriageway labels.

We also inspected the redundancy levels of features selected using the two-stage selection method. Redundancy levels increased when more features per signal were selected in the first stage in almost all cases, and the lowest redundancy levels were generally produced by $PmRMR$. Therefore, $PmRMR$ should be used to select features from CAN-data when minimum redundancy and high performance is required. For other temporal data with fewer biases, such as the OARD, other methods should be used, such as $SUmRMR$. Furthermore, it is likely that permutation methods are computationally quicker than HSIC or $HSICmRMR$. A fair comparison in computation times cannot be made between these two methods here, however, as the permutation method implemented for this thesis is optimised whereas the HSIC implementation is not.

Finally, AUC performances for the WarwickDMD were generally low for evaluations presented in Chapter 4 and this chapter. We therefore investigated the details of this, by building models for different combinations of drivers and making predictions of distraction status and increases in heart rates. We found

that for some inexperienced drivers, where the testing data was made of samples from harder secondary tasks, AUC performances were better than random predictions. This was especially the case when models were built specifically for those drivers, and decreased when models were built for those drivers and others. For the other drivers, models again had very low AUC performances and were no better than a random classifier. The performances of models for predicting the distraction status and increases in heart rates for individual drivers in different sections of the trials were then investigated. We found here that there was no consistent relationship between model predictive performance and what the driver was doing at the time. For instance, the performances of models predicting that the driver was distracted were not different for each of the 0-, 1- and 2-back tasks.

CHAPTER 7

Discussion and conclusions

In this thesis a data mining methodology was developed for building models from vehicle telemetry data. Feature selection techniques were developed for selecting features from such data, with an aim of assessing driver workload. These techniques were been evaluated using both feature rank inspection and classification evaluations, both with vehicle telemetry data and with other example datasets from the UCI and Tuned IT repositories.

The data mining methodology used was based on the general data mining methodology [3, 70, 160], described in Section 2.1. It began with the creation of databases that could be processed by machine learning algorithms. These databases were described in Chapter 3 and included three vehicle telemetry datasets, namely the Road Classification Dataset (RCD), Coventry-JLR Driver Monitoring Dataset (CoventryDMD), and Warwick-JLR Driver Monitoring Dataset (WarwickDMD). Once the database is created, it is then engineered or processed in ways that learning algorithms can be applied to it to produce predictive models. Finally, the models built are evaluated to estimate how they would be expected to perform on new data, and the whole process is refined and iterated on.

The focus of this thesis was in the data engineering stages, and in particular the feature selection stage. Feature extraction was considered for each of the temporal datasets, where temporal summaries were extracted over sliding windows of telemetry signals. In the methodology, the data is then sampled if appropriate, to either reduce their size for processing or to rectify class imbalance. Some machine learning algorithms cannot be directly applied to numeric data such as vehicle telemetry, so a discretization step can be used depending on

the algorithms to be applied later in the methodology [3, 160]. Tree based learning algorithms, for example, usually require the data to be categorical and the features extracted from vehicle telemetry signals must therefore be discretised first. A Multilayer Perceptron, however, takes numerical inputs and discretisation is not required. For such learning algorithms, normalisation may be used instead of discretisation — where the numerical features are normalised to have a fixed range such as between -1 and 1.

In Chapters 4, 5, and 6 the feature selection step of the methodology was explored. The temporal permutation method was developed in Chapter 4, and the dynamic blocking strategy with a cyclic shift was found to be most appropriate for the vehicle telemetry datasets. Of the permutation statistics used, Z_{MI} and MD_{MI} were found to produce feature rankings with fewer bias features than Mutual Information (MI), Symmetrical Uncertainty (SU), and Hilbert-Schmidt Independence Criterion (HSIC), while still having comparable Area Under the Receiver Operator Characteristic Curve (AUC) performances to HSIC. Redundancy in the absence of temporal data and autocorrelations was considered in Chapter 5, using simulated data and example datasets listed in Section 3.4. Using measures such as Z_{MI} as a measure of both relevancy and redundancy is computationally infeasible for big datasets with large numbers of features. A redundancy measure, PC_{MI}, was therefore introduced and was shown to be fast to compute and share properties with Z_{MI}. In classification evaluations, using Z_{MI} as a relevancy measure and PC_{MI} for redundancy in $PmRMR$, was shown to outperform $MImRMR$ and $SUmRMR$.

The findings from Chapters 4, 5 were then combined in Chapter 6. The ordering of feature extraction and feature selection was first considered, before a two-stage feature selection process was explored. It was concluded that extracting features from signals prior to feature selection with MI provided the highest AUC performances with low computational complexity. In the first stage of the two-stage process, features were selected from each signal individually, before the selected features were combined for a second stage of selection.

Selecting fewer features from each signal in the first stage meant that there was less redundancy in the final selected set, and AUC performances also increased.

The remainder of this chapter reviews the contributions made in this thesis with respect to the objectives outlined in Chapter 1, discusses the major limitations of this research, and identifies directions for future work.

7.1 Contributions

1. **Developing an unbiased relevancy measure for temporal variables based on the permutation method.**

   In Chapter 4 we developed a blocked permutation method for feature selection from temporal data. Permutation methods are able to mitigate for biases caused by the Multiple Comparison Procedure (MCP), in the data and feature selection process, but cannot be directly applied to temporal data such as vehicle telemetry. By treating the data in blocks of different sizes and applying a cyclic shift as proposed by [2], we applied the permutation method in ranking features from vehicle telemetry data with known biases. We proposed two non-parametric methods of normalising MI using this blocked permutation method and compared these against one parametric method, and ranking features by their MI, SU, and HSIC with the target variable. We found that fewest bias features were ranked highly by the permutation methods, although slightly higher AUC performances were achieved in a classification evaluation when features were selected using HSIC. From this we make two conclusions. First, k-folds cross evaluation is not sufficient to evaluate a feature set in this domain, as HSIC often highly ranked features with known biases that would be of no use in a general setting. Second, permutation methods, and in particular Z_{MI}, are appropriate for use in feature selection from Controller Area Network (CAN)-bus data. Our investigations were limited in two key respects, however. First, only CAN-bus data was used in finding sta-

ble points in block size for the permutation distributions. Second, the datasets used were made up from sets of contiguous samples that were each only 4 to 5 times as large as the suitable block sizes. The divisions in the data where one journey starts and another ends may have an effect on the block size required for a stable permutation distribution. The journey lengths are typical of data in this domain, however, so these conclusions are representative. To overcome these limitations, analysis with other types of temporal data made of larger contiguous sequences of samples is required.

2. **Establishing a method for feasibly computing unbiased feature redundancies using the permutation method.**

   In Chapter 5, while not considering temporal aspects, we introduced a method for estimating feature redundancy by comparing permutation distributions produced in relevancy computations. We showed that this metric is related to Z_{MI}, and used it in the $PmRMR$ method. In classification evaluations with several datasets available in online repositories we showed that the AUC performance is comparable to $MImRMR$ and $SUmRMR$. To increase the difficulty in feature selection we added new features with different numbers of values that were generated from existing ones. We found that the performance of $MImRMR$ decreased as more features were added with higher dimensionalities, whereas it did not in general for $SUmRMR$ or $PmRMR$. Original features that were expected to be better predictors of the target due to added noise in the extra features, were also ranked higher by $PmRMR$ than by $MImRMR$ or $SUmRMR$. Finally, $PmRMR$ was computationally faster than either $MImRMR$ or $SUmRMR$ for large feature sets and sample sizes when ranking full feature sets, making it preferable for large datasets such as those found in the automotive domain. A main limitation of this research is the use of minimal Redundancy Maximal Relevance (mRMR) and the process used for normalising relevancy and redundancy in each selection iteration. In

future work we intend to use the permutation redundancy measure with other redundancy selection methods such as feature clustering [9, 87] and optimise the weights of redundancy or relevancy terms using k-folds cross validation.

3. **Using known redundancy structure in features extracted from signal data with signal selection, feature extraction, and feature selection.**

   In Chapter 6 we considered the benefits of signal selection prior to feature extraction and feature selection with the RCD. As multiple features are often extracted from each signal, reducing the number of signals before feature extraction reduces the computation required for feature selection. We found, however, that signal selection provided the highest AUC performances when Principal Components Analysis (PCA) was performed on signals, before extracting features from the Principal Components (PCs) and selecting these based on relevancy (i.e. PCA-FE-MI). This work is limited as it considered redundancy and relevancy separately, and using features of PCs made of linear combinations of several signals is not suitable for the vehicular environment. This is both because of the limited computing power in vehicles, and because models with several complex inputs are not easily vetted for safety concerns.

   Therefore, also in Chapter 6 we introduced the two-stage feature selection process where features are first selected from individual signals before being combined for a second selection stage. Here we applied relevancy only selection and mRMR to select between one and five features from each signal with four different relevancy and redundancy measures. We found that selecting fewer features from each signal provided lower redundancy levels and higher AUC performances in the majority of classification evaluations. In this work we assumed that each signal has an equal probability of having a predictive feature, leading to the conclusion that selecting one

feature per signal is preferable. This may not be the case in general, however, and more investigations into signal selection are required to determine whether a signal should be disregarded entirely or if one or more features should be extracted from it. This work is also limited as the classification evaluations were found to be insufficient in Chapter 4. Further evaluation is difficult without collecting more data for analysis, however.

4. **Advancing the process of feature selection from vehicle telemetry data for classification problems such as driver workload estimation.**

   The methodology used in this thesis and described in Chapter 2, enables models to be built that successfully make predictions from telemetry data. It has been demonstrated using telemetry datasets described in Chapter 3, from which features were selected, and models were built and evaluated. In environment monitoring, models built using the RCD were shown to predict if a road had one or more lanes, and determine the road type as defined by a UK governmental classification. Models built using the CoventryDMD were able to determine whether the driver was performing a distraction task or not, while those built using the CoventryDMD had low performance in general. This was discussed in Chapters 4 and 6. In Chapter 6, we applied our data mining methodology and built models for different combinations of drivers in this dataset. We found that, for experienced older drivers the models were still no better than a random classifier. For some younger drivers with fewer than eight years experience, however, we found that these models were capable of predicting the distraction status, and increases in heart rate could be predicted with some accuracy. These conclusions are again limited due to lack of data, however. Mehler et al. [99] and Reimer et al. [127] also found differences in driving behaviour with respect to the age and experience of a driver. Further research into this area is required to investigate the possibility of building different predictive models for different kinds of driver.

7.2 Directions for future research

As well as specific directions of future work outlined in the contributions above, we outline here some more general directions for the work in this thesis.

- **The standardisation of data collection for driver monitoring research.**

  Data collection for driver monitoring has many challenges, including synchronising data streams and deriving a ground truth. In this thesis three forms of ground truth have been used, road type, driver distraction status, and heart rate. Each of these describes the driver state in different ways, from the driving environment to their physiology. It remains an open question in the driver monitoring field which kinds of ground truth best describe cognitive workload. Physiological measures are often considered to be best (e.g. [23, 57, 99, 161]), but require personalised baselines to determine changes in driver status and often require invasive tools to capture the data. Once data is captured, synchronising the streams to other data sources, such as CAN-data is also a challenge to this research, which complicated systems based on Global Positioning Satellites (GPS) or synchronising pulses are often used to overcome. A simplified and standard mechanism is therefore required in future, to enable accurate capture of data in ever more realistic scenarios.

- **Identifying other accessible signals for driver distraction research that are easily added to the vehicle environment.**

  This thesis has focussed on the use of CAN-bus telemetry signals, which are behavioural measures of the driver and vehicle in the driving environment. The set of signals in the telemetry collected was not exhaustive, however, as more are added to the modern vehicle each year. For example, lane position data was not available in vehicles used for data collection in this thesis, although it may be a useful feature in models considered. Also, physiological signals may prove to be more accurate while being non-

intrusive. This would allow them to easily be used as inputs to predictive models, rather than in research conditions only.

- **Investigating other mechanisms for relevancy and redundancy assessment from permutation distributions.**

  The work in Chapter 4 was focussed mainly on the blocking process in the permutation method to apply it to temporal data. Two methods for relevancy assessment were introduced, namely MR_{MI} and MD_{MI}, but the investigations were not exhaustive. For instance, ranking features with low p-values in a separate tranche to those with high p-values, as used by Radivojac et al. [118]. Likewise, in Chapter 5 we proposed to use PC_{MI} as a redundancy measure and this had a correlation of 0.89 with Z_{MI} in simulations with features of different dimensionalities. There are several other distance or similarity measures that could be used in place of Pearson's Correlation Coefficient (PCC), such as the cosine distance or Spearman Correlation Coefficient, and may have higher performances in some cases than PC_{MI}.

- **Using permutation methods for redundancy feature selection in frameworks such as Least Angle regression.**

  Recently there have been efforts to use HSIC in a redundancy feature selector using the Least Absolute Shrinkage and Selection Operator (LASSO) regression [164], and Least Angle Regression (LARS) [165]. We believe that these optimisation algorithms may be also applicable to redundancy feature selection with the permutation method. Specifically, both methods are unaffected by scale differences in the relevancy and redundancy scores, and so they avoid issues with normalisation in the mRMR selection process.

7.3 Final remarks

In this thesis, we have developed a data mining methodology for driver monitoring, with an aim to build predictive models using CAN-bus telemetry data as input and parameters describing the driver and driving environment as outputs. We have focussed on input selection for such models, and proposed the use of permutation methods to mitigate biases caused by the MCP that are prevalent in this domain. In Chapter 4 we developed permutation methods for use with temporal data, in Chapter 5 we proposed a permutation redundancy measure enabling efficient permutation redundant feature selection. These were then combined in a two-stage temporal redundancy feature selection process in Chapter 6, to take advantage of known redundancy structure in features extracted from signals.

This thesis considered only automatic feature selection, where algorithms decided on the features to be used in predictive models. Automated methods were chosen because engineers often have preferences that may cause suboptimal features to be selected. Automated methods are dependent on the data used, however, which may not be fully descriptive of the problem. In fact, in many cases it is infeasible to collect such data, so some amount of human guidance is required in the selection process. We therefore propose that automated methods guide the selection process, but do not perform it in full. For example, an automated method may propose several features, based on data collected, which may be of use in predicting a certain target. A domain expert should then analyse these features further, and use their knowledge and experience in deciding whether to use a feature or not.

We have found that CAN-bus data is unable to predict the cognitive workload status of drivers reliably. Having said this, we found features directly linked to the driver to be the best predictors, such as features of the steering wheel angle and brake position. Combining these inputs with some non-intrusive physiological measures may eventually yield an effective real-time distraction

monitoring system for a driver.

Even in the presence of bias and redundancy in features, we have shown that it is possible to select good features for use in predictive models. We believe that this is a key step towards developing a data mining methodology for learning from telemetry data in the vehicle domain.

References

[1] U. Acharya, K. Joseph, N. Kannathal, Choo Lim, and Jasjit Suri. Heart rate variability: a review. *Medical and Biological Engineering and Computing*, 44(12):1031–1051, 2006.

[2] Daniela Adolf, Sebastian Baecke, Waltraud Kahle, Johannes Bernarding, and Siegfried Kropf. Applying multivariate techniques to high-dimensional temporally correlated fMRI data. *Journal of Statistical Planning and Inference*, 141(12):3760–3770, June 2011.

[3] Charu Aggarwal. *Managing and mining sensor data.* Springer, Boston, MA, 2013.

[4] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengaue. Permutation importance: A corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, March 2010.

[5] Cláudia Antunes and Arlindo Oliveira. Temporal data mining: An overview. In *KDD Workshop on Temporal Data Mining*, pages 1–13. ACM New York, NY, August 2001.

[6] Georges Aoude, Vishnu Desaraju, Lauren Stephens, and Jonathan How. Driver behavior classification at intersections and validation on large naturalistic data set. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):724–736, June 2012.

[7] Francis Bach and Michael Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, March 2003.

[8] Adam Berger. Error-correcting output coding for text classification. In *Workshop on Machine Learning for Information Filtering*, pages 1–8. Technische Universität Dortmund, August 1999.

[9] José Bins and Bruce Draper. Feature selection from huge feature sets. In *International Conference on Computer Vision*, volume 2, pages 159–165, July 2001.

[10] Wolfram Boucsein. *Electrodermal activity*. Springer, Boston, MA, 2012.

[11] James Bradley. *Distribution-free statistical tests*. Prentice-Hall, Englewood Cliffs, NJ, 1968.

[12] Philipp Caffier, Udo Erdmann, and Peter Ullsperger. Experimental evaluation of eye-blink parameters as a drowsiness measure. *European Journal of Applied Physiology*, 89(3):319–325, March 2003.

[13] Brad Cain. A review of the mental workload literature. Technical report, Defence Research and Development Toronto (Canada), July 2007.

[14] Thomas Carlson and Thomas Austin. Development of speed correction cycles. Technical report, Sierra Research Inc, Sacramento, CA, April 1997.

[15] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling TEchnique. *Journal of Artificial Intelligence Research*, 16:321–357, February 2002.

[16] Su-Fen Chen. Redundant feature selection based on hybrid GA and BPSO. In *International Conference on Communication Software and Networks*, pages 414–418. IEEE, May 2011.

[17] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David Wu, and Andrew Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *International Conference on Document Analysis and Recognition*, pages 440–445. IEEE, September 2011.

[18] Nicholas Cottrell and Benjamin Barton. The role of automation in reducing stress and negative affect while driving. *Theoretical Issues in Ergonomics Science*, 14(1):53–68, 2013.

[19] Joseph Coughlin, Bryan Reimer, and Bruce Mehler. Monitoring, managing, and motivating driver safety and well-being. *IEEE Pervasive Computing*, 10(3):14–21, July 2011.

[20] Jacob Crossman, Hong Guo, Yi Murphey, and John Cardillo. Automotive signal fault diagnostics - part I: Signal fault analysis, signal segmentation, feature extraction and quasi-optimal feature selection. *IEEE Transactions on Vehicular Technology*, 52(4):1063–1075, July 2003.

[21] Houtao Deng, George Runger, and Eugene Tuv. Bias of importance measures for multi-valued attributes and solutions. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning*, volume 6792 of *Lecture Notes in Computer Science*, pages 293–300. Springer Berlin Heidelberg, June 2011.

[22] UK Department for Transport. Guidance on road classification and the primary route network. Online, January 2012.

[23] Yanchao Dong, Zhencheng Hu, Keiichi Uchimura, and Nobuki Murayama. Driver inattention monitoring system for intelligent vehicles: A review. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):596–614, June 2011.

[24] Eugene Edgington and Patrick Onghena. *Randomization Tests*. Chapman & Hall/CRC, Sound Parkway NW, FL, 2007.

[25] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: Using a mobile sensor network for road surface monitoring. In *International conference on Mobile systems, applications, and services*, pages 29–39. ACM New York, NY, June 2008.

[26] Michael Ernst. Permutation methods: A basis for exact inference. *Statistical Science*, 19(4):676–685, November 2004.

[27] Tulga Ersal, Helen Fuller, Omer Tsimhoni, Jeffrey Stein, and Hosam Fathy. Model-based analysis and classification of driver distraction under secondary tasks. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):692–701, September 2010.

[28] Pablo Estevez, Michel Tesmer, Claudio Perez, and Jacek Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, February 2009.

[29] Konrad Etschberger. *Controller area network: Basics, protocols, chips and applications*. IXXAT Automation GmbH, 2001.

[30] Mohammad Farsi, Karl Ratcliff, and Manuel Barbosa. An overview of controller area network. *Computing Control Engineering Journal*, 10(3): 113–120, June 1999.

[31] Usama Fayyad and Keki Irani. Multi-interval discretization of continuous valued attributes for classification learning. In *International Joint Conference on Articial Intelligence*, volume 2, pages 1022–1027, September 1993.

[32] Ronald Fisher. "The coefficient of racial likeness" and the future of craniometry. *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, 66:57–63, January 1936.

[33] Marco Flores, José Armingol, and Arturo de la Escalera. Driver drowsiness detection system under infrared illumination for an intelligent vehicle. *IET Intelligent Transport Systems*, 5(4):241–251, December 2011.

[34] The Royal Society for the Prevention of Accidents. Driver fatigue and road accidents. Online, June 2011.

[35] Marie-Josée Fortin and Geoffrey Jacquez. Randomization tests and spatially auto-correlated data. *Bulletin of the Ecological Society of America*, 81(3):201–205, July 2000.

[36] Marie-Josée Fortin, Geoffrey Jacquez, and Bill Shipley. Computer intensive sampling methods in ecology. *Encyclopedia of Environmetrics*, 1: 399–402, January 2002.

[37] D. François, V. Wertz, and M. Verleysen. The permutation test for feature selection by mutual information. In *European Symposium on Artificial Neural Networks*, pages 239–244, April 2006.

[38] Eibe Frank and Ian Witten. Using a permutation test for attribute selection in decision trees. In *International Conference on Machine Learning*, pages 152–160. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1998.

[39] Cecille Freeman, Dana Kulić, and Otman Basir. An evaluation of classifier-specific filter measure performance for feature selection. *Pattern Recognition*, 48(5):1812–1826, May 2015.

[40] Phillip Good. *Permutation tests: A practical guide to resampling methods for testing hypotheses*, volume 2. Springer Science & Business Media, New York, 2000.

[41] Timothy Gordon and Zevi Bareket. Vibration transmission from road surface features: Vehicle measurement and detection. Technical report, The University of Michigan, Ann Arbor, MI, January 2007.

[42] Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In *Neural Information Processing Systems*, pages 585–592. Curran Associates, Inc., December 2008.

[43] Hong Guo, Jocob Crossman, Yi Murphey, and Mark Coleman. Automotive signal diagnostics using wavelets and machine learning. *IEEE Transactions on Vehicular Technology*, 49(5):1650–1662, September 2000.

[44] Hong Guo, Lindsay Jack, and Asoke Nandi. Feature generation using genetic programming with application to fault classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 35(1): 89–99, February 2005.

[45] Ling Guo, Daniel Rivero, Julián Dorado, Cristian Munteanu, and Alejandro Pazos. Automatic feature extraction using genetic programming: An application to epileptic EEG classification. *Expert Systems with Applications*, 38(8):10425–10436, August 2011.

[46] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[47] Eija Haapalainen, SeungJun Kim, Jodi F Forlizzi, and Anind K Dey. Psycho-physiological measures for assessing cognitive load. In *International conference on Ubiquitous computing*, pages 301–310. ACM, 2010.

[48] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *The International Society of Music Information Retrieval*, pages 339–344, August 2010.

[49] David Hand. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123, October 2009.

[50] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of data mining*, volume 1. The MIT Press, Cambridge, MA, 1 edition, 2001.

[51] Alexander Hapfelmeier and Kurt Ulm. A new variable selection approach using random forests. *Computational Statistics & Data Analysis*, 60:50–69, April 2013.

[52] Sandra Hart and Lowell Staveland. Development of NASA-TLX (task load index): Results of empirical and theoretical research. In Peter Hancock

and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988.

[53] Werner Hauptmann, Friedrich Graf, and Kai Heesche. Driving environment recognition for adaptive automotive systems. In *IEEE International Conference on Fuzzy Systems*, volume 1, pages 387–393. IEEE, September 1996.

[54] Toshinori Hayashi and Keiichi Yamada. Predicting unusual right-turn driving behavior at intersection. In *IEEE Intelligent Vehicles Symposium*, pages 869–874. IEEE, June 2009.

[55] Haibo He and Edwardo Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

[56] Qichang He, Wei Li, and Xiumin Fan. Estimation of driver's fatigue based on steering wheel angle. In Don Harris, editor, *Engineering Psychology and Cognitive Ergonomics*, volume 6781 of *Lecture Notes in Computer Science*, pages 145–155. Springer Berlin Heidelberg, July 2011.

[57] Jennifer Healey and Rosalind Picard. Detecting stress during real-world driving tasks using physiological sensors. *IEEE Transactions on Intelligent Transportation Systems*, 6(2):156–166, June 2005.

[58] Gunawan Hermana, Bang Zhanga, Yang Wanga, Getian Yec, and Fang Chena. Mutual information-based method for selecting informative feature sets. *Pattern Recognition*, 46(12):3315–3327, December 2013.

[59] Jin Huang and Charles Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, March 2005.

[60] Xi Huang, Ying Tan, and Xingui He. An intelligent multifeature statistical approach for the discrimination of driving conditions of a hybrid electric

vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 12(2): 453–465, June 2011.

[61] Emmanuel Ifeachor and Barrie Jervis. *Digital signal processing: A practical approach*. Pearson Education, New York, NY, 2 edition, 2002.

[62] Paul Jansen, Wannes van der Mark, Johan van den Heuvel, and Frans Groen. Colour based off-road environment and terrain type classification. In *IEEE International Conference on Intelligent Transportation Systems*, pages 216–221. IEEE, September 2005.

[63] Nathalie Japkowicz and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, 2011.

[64] Kashif Javed, Haroon Babri, and Mehreen Saeed. Feature selection based on class-dependent densities for high-dimensional binary data. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):465–477, March 2012.

[65] David Jensen and Paul Cohen. Multiple comparisons in induction algorithms. *Machine Learning*, 38(3):309–338, March 2000.

[66] Qiang Ji, Zhiwei Zhu, and Peilin Lan. Real-time nonintrusive monitoring and prediction of driver fatigue. *IEEE Transactions on Vehicular Technology*, 53(4):1052–1068, July 2004.

[67] Shengyi Jiang and Lianxi Wang. Unsupervised feature selection based on clustering. In *IEEE International Conference on Bio-Inspired Computing: Theories and Applications*, pages 263–270. IEEE, September 2010.

[68] Jaeik Jo, Sung Lee, Kang Park, Ig-Jae Kim, and Jaihie Kim. Detecting driver drowsiness using feature-level fusion and user-specific classification. *Expert Systems with Applications*, 41(4):1139–1152, March 2014.

[69] Karl Johansson, Martin Törngren, and Lars Nielsen. Vehicle applications of controller area network. In Dimitrios Hristu-Varsakelis and William Levine, editors, *Handbook of networked and embedded control systems*, pages 741–765. Birkh"auser Boston, MA, 2005.

[70] George John. *Enhancements to the data mining process.* PhD thesis, stanford university, Stanford, CA, March 1997.

[71] Mehmed Kantardzic. *Data mining: Concepts, models, methods, and algorithms.* Wiley-IEEE Press, 2011.

[72] Steffen Kempe and Rudolf Kruse. Mining temporal patterns in an automotive environment. In José Luis Verdegay, Manuel Ojeda-Aciego, and Luis Magdalena, editors, *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 521–528. University of Málaga, Málaga, June 2008.

[73] Ata Khan. Bayesian-monte carlo model for collision avoidance system design of cognitive connected vehicle. *International Journal of Intelligent Transportation Systems Research*, 11(1):23–33, January 2013.

[74] Ata Khan, Ataur Bacchus, and Stephen Erwin. Surrogate safety measures as aid to driver assistance system design of the cognitive vehicle. *IET Intelligent Transport Systems*, 8(4):415–424, June 2014.

[75] Claudia Kirch. Block permutation principles for the change analysis of dependent data. *Journal of Statistical Planning and Inference*, 137(7): 2453–2474, July 2007.

[76] Ron Kohavi and George John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, December 1997.

[77] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, July 1995.

[78] Rudolf Kruse, Matthias Steinbrecher, and Christian Moewes. Data mining applications in the automotive industry. In Michael Beer, Rafi Muhanna, and Robert Mullen, editors, *International Workshop on Reliable Engineering Computing*, pages 23–40. Research Publishing Services, March 2010.

[79] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann Publishers Inc., San Francisco, CA, July 1997.

[80] Lukasz Kurgan and Krzysztof Cios. Caim discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, February 2004.

[81] Nojun Kwak and Chong-Ho Choi. Input feature selection by mutual information based on Parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1667–1671, December 2002.

[82] Soumendra Lahiri. *Resampling methods for dependent data.* Springer, New York, NY, 2003.

[83] Reza Langari and Jong-Seob Won. Intelligent energy management agent for a parallel hybrid vehicle-part I: System architecture and design of the driving situation identification process. *IEEE Transactions on Vehicular Technology*, 54(3):925–934, May 2005.

[84] Daniel Larose. *Discovering knowledge in data: An introduction to data mining.* Wiley Publishing, Hoboken, NJ, 2014.

[85] Erich Lehmann and Joseph Romano. *Testing statistical hypotheses.* Springer, New York, NY, 2006.

[86] Aiguo Li and Baonan Wang. Feature subset selection based on binary particle swarm optimization and overlap information entropy. In *Interna-*

tional Conference on Computational Intelligence and Software Engineering, pages 1–4. IEEE, December 2009.

[87] Guangrong Li, Xiaohua Hu, Xiajiong Shen, Xin Chen, and Zhoujun Li. A novel unsupervised feature selection method for bioinformatics data sets through feature clustering. In *IEEE International Conference on Granular Computing*, pages 41–47. IEEE, August 2008.

[88] Li Li, Klaudius Werber, Carlos Calvillo, Khac Dinh, Ander Guarde, and Andreas König. Multi-sensor soft-computing system for driver drowsiness detection. In Václav Snášel, Pavel Krömer, Mario Köppen, and Gerald Schaefer, editors, *Soft Computing in Industrial Applications*, volume 223 of *Advances in Intelligent Systems and Computing*, pages 129–140. Springer International Publishing, 2014.

[89] Xiaobo Li and Richard Dubes. Tree classifier design with a permutation statistic. *Pattern Recognition*, 19(3):229–235, 1986.

[90] Charles Liu, Simon Hosking, and Michael Lenné. Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. *Journal of Safety Research*, 40(4):239–245, August 2009.

[91] Xu-Ying Liu, Qian-Qian Li, and Zhi-Hua Zhou. Learning imbalanced multi-class data with optimal dichotomy weights. In *IEEE International Conference on Data Mining*, pages 478–487. IEEE, December 2013.

[92] John Ludbrook and Hugh Dudley. Why permutation tests are superior to t and F tests in biomedical research. *The American Statistician*, 52(2): 127–132, March 1998.

[93] Bryan Matthews, Santanu Das, Kanishka Bhaduri, Kamalika Das, Rodney Martin, and Nikunj Oza. Discovering anomalous aviation safety events using scalable data mining algorithms. *Journal of Aerospace Information Systems*, 10(10):467–475, 2013.

[94] Evangelos Mazomenos, Dwaipayan Biswas, Amit Acharyya, Taihai Chen, Koushik Maharatna, James Rosengarten, John Morgan, and Nick Curzen. A low-complexity ECG feature extraction algorithm for mobile healthcare applications. *IEEE Journal of Biomedical and Health Informatics*, 17(2): 459–469, March 2013.

[95] Joel McCall and Mohan Trivedi. Human behavior based predictive brake assistance. In *IEEE Intelligent Vehicles Symposium*, pages 8–12. IEEE, June 2006.

[96] Joel McCall and Mohan Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *Proceedings of the IEEE*, 95(2): 374–387, February 2007.

[97] Joel McCall, David Wipf, Mohan Trivedi, and Bhaskar Rao. Lane change intent analysis using robust operators and sparse bayesian learning. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):431–440, September 2007.

[98] Natasha McCarthy. Autonomous systems: social, legal and ethical issues. pages 1–19, November 2009.

[99] Bruce Mehler, Bryan Reimer, and Joseph Coughlin. Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task: An on-road study across three age groups. *Human Factors*, 54(3):396–412, April 2012.

[100] Ingo Mierswa and Katharina Morik. Automatic feature extraction for classifying audio data. *Machine Learning*, 58(2-3):127–149, February 2005.

[101] Rahul Mundke, Sachitanand Malewar, and Kavi Arya. Use of data recorder for driver rating. *SAE Transactions Journal of Passenger Cars: Electronic and electrial Ssytems*, April 2006.

[102] Yi Murphey, Jacob Crossman, Zhihang Chen, and John Cardillo. Automotive fault diagnosis - part II: A distributed agent diagnostic system. *IEEE Transactions on Vehicular Technology*, 52(4):1076–1098, July 2003.

[103] Yi Murphey, ZhiHang Chen, Leo Kiliaris, Jungme Park, Ming Kuang, Abul Masrur, and Anthony Phillips. Neural learning of driving environment prediction for vehicle power management. In *IEEE International Joint Conference on Neural Networks*, pages 3755–3761. IEEE, June 2008.

[104] Tien Nguyen, Manuel Avila, and Stéphane Begot. Detection of defects in road surface by a vision system. In *IEEE Mediterranean Electrotechnical Conference*, pages 847–851. IEEE, May 2008.

[105] Thomas Nichols and Andrew Holmes. Nonparametric permutation tests for functional neuroimaging: A primer with examples. *Human Brain Mapping*, 15(1):1–25, October 2002.

[106] Ian Noy, Tracy Lemoine, Christopher Klachan, and Peter Burns. Task interruptability and duration as measures of visual distraction. *Applied Ergonomics*, 35(3):207–213, May 2004.

[107] Markus Ojala. *Randomization algorithms for assessing the significance of data mining results.* Aalto University, Helsinki, Finland, 2011.

[108] Markus Ojala and Gemma Garriga. Permutation tests for studying classifier performance. *The Journal of Machine Learning Research*, 99: 1833–1863, March 2010.

[109] John Opdyke. Fast permutation tests that maximize power under conventional Monte Carlo sampling for pairwise and multiple comparisons. *Journal of Modern Applied Statistical Methods*, 2(1):27–49, May 2003.

[110] Jason Osborne. *Best practices in data cleaning.* Sage, Thousand Oaks, CA, 2012.

[111] Jungme Park, Zhihang Chen, Leonadis Kiliaris, Yi Murphey, Ming Kuang, Andrew Phillips, and Abul Masrur. Intelligent vehicle power control based on machine learning of optimal control parameters and prediction of road type and traffic congestion. *IEEE Transactions on Vehicular Technology*, 58(9):4741–4756, November 2009.

[112] Annie Pauzie. A method to assess the driver mental workload: The driving activity load index (dali). *IET Intelligent Transport Systems*, 2(4): 315–322, December 2008.

[113] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, august 2005.

[114] Joseph Pompei, Taly Sharon, Stephen Buckley, and James Kemp. An automobile-integrated system for assessing and reacting to driver cognitive load. In *IEEE Proceedings of Convergence*, pages 411–416. SAE Press, October 2002.

[115] Andrey Ptitsyn, Sanjin Zvonic, and Jeffrey Gimble. Permutation test for periodicity in short time series data. *BMC bioinformatics*, 8(1):1–9, October 2007.

[116] Liu Qiao, Mitsuo Sato, and Hiroshi Takeda. Learning algorithm of environmental recognition in driving vehicle. *IEEE Transactions on Systems, Man and Cybernetics*, 25(6):917–925, June 1995.

[117] Liu Qiao, Mitsuo Sato, Kenichi Abe, and Hiroshi Takeda. Self-supervised learning algorithm of environment recognition in driving vehicle. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(6):843–850, November 1996.

[118] Predrag Radivojac, Zoran Obradovic, Keith Dunker, and Slobodan Vucetic. Feature selection filters based on the permutation test. In Jean-

François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *European Conference on Machine Learning*, volume 3201 of *Lecture Notes in Computer Science*, pages 334–346. Springer Berlin Heidelberg, September 2004.

[119] Arjun Raj, Dilip Krishna, Ramachandran Priya, Kumar Shantanu, and Selvaraj Niranjani Devi. Vision based road surface detection for automotive systems. In *International Conference on Applied Electronics*, pages 223–228, September 2012.

[120] Mohammad Norouziand Mani Ranjbar and Greg Mori. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2735–2742. IEEE, June 2009.

[121] Marc'aurelio Ranzato, Y-lan Boureau, and Yann Cun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1185–1192. Curran Associates, Inc., March 2008.

[122] Donald Redelmeier and Robert Tibshirani. Association between cellular-telephone calls and motor vehicle collisions. *New England Journal of Medicine*, 336(7):453–458, February 1997.

[123] Donald Redelmeier and Robert Tibshirani. Car phones and car crashes: Some popular misconceptions. *Canadian Medical Association Journal*, 164(11):1581–1582, 2001 2001.

[124] Michael Regan. Driver distraction: Reflections on the past, present and future. *Journal of the Australasian College of Road Safety*, 16(2):22–33, December 2005.

[125] Michael Regan, Charlene Hallett, and Craig Gordon. Driver distraction and driver inattention: Definition, relationship and taxonomy. *Accident Analysis & Prevention*, 43(5):1771–1781, September 2011.

[126] Bryan Reimer, Joseph Coughlin, and Bruce Mehler. Development of a driver aware vehicle for monitoring, managing & motivating older operator behavior. In *Proceedings of Intelligent Transportation Systems – America*, pages 1–9, June 2009.

[127] Bryan Reimer, Bruce Mehler, Ying Wang, and Joseph Coughlin. A field study on the impact of variations in short-term memory demands on drivers' visual attention and driving performance across three age groups. *Human Factors*, 54(3):454–468, February 2012.

[128] Joao Rodrigues, Francisco Vieira, Tiago Vinhoza, Joo Barros, and Joao Cunha. A non-intrusive multi-sensor system for characterizing driver behavior. In *IEEE Conference on Intelligent Transportation Systems*, pages 1620–1624. IEEE, September 2010.

[129] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Sesam Sagha, Hamidreza Bayati, Marco Creatura, and José del Millàn. Collecting complex activity datasets in highly rich networked sensor environments. In *International Conference on Networked Sensing Systems*, pages 233–240. IEEE, June 2010.

[130] Riaz Sayed and Azim Eskandarian. Unobtrusive drowsiness detection by neural network learning of driver steering. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215 (9):969–975, September 2001.

[131] Caroline Schiessl. Subjective strain estimation depending on driving manoeuvres and traffic situation. *IET Intelligent Transport Systems*, 2(4): 258–265, December 2008.

[132] Stefan Schneegass, Bastian Pfleging, Nora Broy, Frederik Heinrich, and Albrecht Schmidt. A data set of real world driving to assess driver work-

load. In *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 150–157. ACM, October 2013.

[133] Miranda Schreurs and Sibyl Steuwer. Autonomous driving political, legal, social, and sustainability dimensions. In Markus Maurer, J. Christian Gerdes, Barbara Lenz, and Hermann Winner, editors, *Autonomes Fahren*, pages 151–173. Springer Berlin Heidelberg, 2015.

[134] Keiji Shibata, Tatsuya Furukane, Shohei Kawai, and Yuukou Horita. Distinction of wet road surface condition at night using texture features. *Electronics and Communications in Japan*, 97(6):51–57, June 2014.

[135] Paolo Soda and Guilio Iannello. Decomposition methods and learning approaches for imbalanced dataset: An experimental integration. In *International Conference on Pattern Recognition*, pages 3117–3120. IEEE, August 2010.

[136] Joonwoo Son, Myoungok Park, and Hosang Oh. Sensitivity of multiple cognitive workload measures: A field study considering environmental factors. In *Adjunct Proceedings of the Automotive User Interfaces and Vehiclular Applications*. ACM, October 2012.

[137] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 13(1):1393–1434, May 2012.

[138] Qinbao Song, Jingjie Ni, and Guangtao Wang. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):1–14, August 2013.

[139] Alan Stevens, Gary Burnett, and Tim Horberrym. A reference level for assessing the acceptable visual demand of in-vehicle information systems. *Behaviour & Information Technology*, 29(5):527–540, February 2010.

[140] Jane Stutts, Donald reinfurt, Loren Staplin, and Eric Rodgeman. *The role of driver distraction in traffic crashes.* AAA Foundation for Traffic Safety, Washington, DC, 2001.

[141] Hongbin Sun, Hao Wang, Boming Zhang, and Feng Zhao. PGFB: A hybrid feature selection method based on mutual information. In *International Conference on Fuzzy Systems and Knowledge Discovery*, pages 2862–2871. IEEE, August 2010.

[142] Isabelle Tang and Toby Breckon. Automatic road environment classification. *IEEE Transactions on Intelligent Transportation Systems*, 12(2): 476–484, June 2011.

[143] Fabio Tango and Marco Botta. Evaluation of distraction in a driver-vehicle-environment framework: An application of different data-mining techniques. In Petra Perner, editor, *Proceedings of the Industrial Conference on Advances in Data Mining: Applications and Theoretical Aspects*, volume 5633 of *Lecture Notes in Computer Science*, pages 176–190. Springer Berlin Heidelberg, July 2009.

[144] Phillip Taylor, Fatima Adamu-Fika, Sarabjot Anand, Alain Dunoyer, Nathan Griffiths, and Thomas Popham. Road type classification through data mining. In *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 233–240. ACM, October 2012.

[145] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Alain Dunoyer, Thomas Popham, and Zhou Xu. Feature selection in highly redundant signal data: A case study in vehicle telemetry data and driver monitoring. In *International Workshop on Autonomous Intelligent Systems: Multi-Agents and Data Mining*, pages 25–36, June 2013.

[146] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Derrick Watson, Zhou Xu, and Thomas Popham. Warwick-JLR Driver Monitoring Dataset

(DMD): A public dataset for driver monitoring research. In *Cognitive Load and In-Vehicle Human-Machine Interaction*, pages 1–4, October 2013.

[147] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Thomas Popham, Zhou Xu, and Alain Dunoyer. Redundant feature selection for telemetry data. In Longbing Cao, Yifeng Zeng, Andreas Symeonidis, Vladimir Gorodetsky, Jörg Müller, and Philip Yu, editors, *Agents and Data Mining Interaction*, volume 8316 of *Lecture Notes in Computer Science*, pages 53–65. Springer Berlin Heidelberg, May 2014.

[148] Phillip Taylor, Nathan Griffiths, and Abhir Bhalerao. Redundant feature selection using permutation methods. In *Automatic Machine Learning Workshop*, pages 1–8, July 2015.

[149] Phillip Taylor, Nathan Griffiths, Abhir Bhalerao, Derrick Watson, Zhou Xu, Adam Gelenscer, and Thomas Popham. Developing a public driver monitoring dataset. In *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, September 2015.

[150] Patrick Tchankue, Janet Wesson, and Dieter Vogts. The impact of an adaptive user interface on reducing driver distraction. In *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '11, pages 87–94, New York, NY, 2011. ACM.

[151] Bernd Ludwig Tobias Islinger, Thorsten Köhler. Using the fast fourier transformation for analyzing the steering wheel angle in distracted driving situations. In *Adjunct Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 21–22. ACM New York, NY, November 2011.

[152] Kari Torkkola, Noel Massey, and Chip Wood. Detecting driver inattention in the absence of driver monitoring sensors. In *International Conference*

on Machine Learning and Applications, pages 220–226. IEEE, December 2004.

[153] Cheng-Jung Tsai, Chien-I Lee, and Wei-Pang Yang. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, 178(3):714–731, February 2008.

[154] La Vinh, Nhuyen Thang, and Young-Koo Lee. An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information. In *IEEE/IPSJ International Symposium on Applications and the Internet*, pages 395–398. IEEE, July 2010.

[155] Haixun Wang, Jian Yin, Jian Pei, Philip Yu, and Jeffrey Yu. Suppressing model overfitting in mining concept-drifting data streams. In *International Conference on Knowledge Discovery and Data Mining*, pages 736–741. ACM New York, NY, August 2006.

[156] Jane Wang, Pamela Lee, and Martin McKeown. A novel segmentation, mutual information network framework for EEG analysis of motor tasks. *BioMedical Engineering OnLine*, 8(1):1–19, May 2009.

[157] Rui Wang and Srdjan Lukic. Review of driving conditions prediction and driving style recognition based control algorithms for hybrid electric vehicles. In *IEEE on Vehicle Power and Propulsion Conference*, pages 1–7. IEEE, September 2011.

[158] Wiebke Werft, Axel Benner, and Annette Kopp-Schneider. On the identification of predictive biomarkers: Detecting treatment-by-gene interaction in high-dimensional data. *Computational Statistics & Data Analysis*, 56 (5):1275–1286, May 2012.

[159] Daniel Wilks. Resampling hypothesis tests for autocorrelated fields. *Journal of Climate*, 10(1):65–82, May 1997.

[160] Ian Witten, Eibe Frank, and Mark Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, San Francisco, CA, 2011.

[161] Martin Wollmer, Christoph Blaschke, Thomas Schindl, Björn Schuller, Berthold Färber, Stefan Mayer, and Benjamin Trefflich. Online driver distraction detection using long short-term memory. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):273–324, June 2011.

[162] Feng Xie, Andy Song, and Vic Ciesielski. Human action recognition from multi-sensor stream data by genetic programming. In Anna Esparcia-Alcázar, editor, *Applications of Evolutionary Computation*, volume 7835 of *Lecture Notes in Computer Science*, pages 418–427. Springer Berlin Heidelberg, April 2013.

[163] Takehisa Yairi, Yoshinobu Kawahara, Ryohei Fujimaki, Yuichi Sato, and Kazuo Machida. Telemetry-mining: A machine learning approach to anomaly detection and fault diagnosis for space systems. In *IEEE International Conference on Space Mission Challenges for Information Technology*, pages 476–484. IEEE, July 2006.

[164] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, 26(1):185–207, January 2014.

[165] Makoto Yamada, Avishek Saha, Hua Ouyang, Dawei Yin, and Yi Chang. N3LARS: Minimum redundancy maximum relevance feature selection for large and high-dimensional data. *arXiv preprint*, Novenmber 2014.

[166] Tomoya Yamada and Takakazu Sugiyama. On the permutation test in canonical correlation analysis. *Computational Statistics & Data Analysis*, 50(8):2111–2123, April 2006.

[167] Yan Yang. *The effects of increased workload on driving performance and*

visual behaviour. PhD thesis, University of Southampton, Southampton, UK, June 2011.

[168] Kristie Young and Michael Regan. Driver distraction: A review of the literature. *Distracted driving. Sydney, NSW: Australasian College of Road Safety*, pages 379–405, 2007.

[169] Richard Young, Li Hsieh, and Sean Seaman. The tactile detection response task: Preliminary validation for measuring the attentional effects of cognitive load. In *International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, pages 71–77, 2013.

[170] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, October 2004.

[171] Xinhua Zhang, Le Song, Arthur Gretton, and Alex Smola. Kernel measures of independence for non-IID data. In *Advances in Neural Information Processing Systems*, pages 1937–1944. Curran Associates, Inc., December 2009.

[172] Yilu Zhang, Yuri Owechko, and Jing Zhang. Learning-based driver workload estimation. In Danil Prokhorov, editor, *Computational Intelligence in Automotive Applications*, volume 132 of *Studies in Computational Intelligence*, pages 1–24. Springer Berlin Heidelberg, 2008.

[173] Yilu Zhang, William Lin, and Yuen Kwok Chin. A pattern-recognition approach for driving skill characterization. *IEEE Transactions on Intelligent Transportation Systems*, 11(4):905–916, December 2010.

[174] Chunxiao Zhou and Yongmei Wang. New blockwise permutation tests preserving exchangeability in functional neuroimaging. In *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6977–6980. IEEE, September 2009.

[175] Meilan Zhou, Xue Ao, and Jian Wang. Fault diagnosis of automobile based on CAN bus. In Luo Qi, editor, *Information and Automation*, volume 86 of *Information and Automation*, pages 317–323. Springer Berlin Heidelberg, November 2011.