UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Wearable device localisation and its effect on activity recognition using machine learning

by

Damian de Arruda

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

June 2019

© University of Pretoria

SUMMARY

WEARABLE DEVICE LOCALISATION AND ITS EFFECT ON ACTIVITY RECOGNITION USING MACHINE LEARNING

by

Damian de Arruda

Supervisor(s):	Hancke, Gerhard & Myburgh, Herman
Department:	Electrical, Electronic and Computer Engineering University
University:	of Pretoria
Degree:	Master of Engineering (Computer Engineering) localisation,
Keywords:	context awareness, human activity recognition, ma-chine
	learning, classification, feature selection

Many developments have been observed from research into activity recognition. Alongside these developments, many challenges have also been identified which affect the design, implementation and evaluation of the activity recognition systems performance. One such challenge is the successful inclusion of contextual awareness in order to improve the system's performance. This research seeks to examine the effect of localising a wearable device, in the activity recognition problem. Three machine learning models were implemented, which make use of the on-body device location in different ways. The first model contains no knowledge of the on-body device location, the second model contains the encoded location of the device as a feature in the dataset, the third model separates each dataset according to their corresponding location, with each location being treated as an independent problem. A final fourth model was proposed and implemented which attempts to closely emulate the best performing model of the previous three, while being fully autonomous. The autonomy is achieved by applying another classification step to determine the device location and then performing activity recognition.

The performance of each model was tested using various combinations of feature selection algorithms and classifiers. When using no location information, model 1 generated a classification accuracy of

© University of Pretoria

89%; using the location as an encoded feature inserted into the dataset, model 2 yielded a classification accuracy of 90.2%. Classification of the activities when considering training data only from the location of the wearable device, model 3 generated an average accuracy of 95.5%. The fully autonomous model 4, which was based on the activity recognition in model 3, managed to achieve a 94.5% classification accuracy. These results show that using the location of the device to give the system added context, makes a statically significant impact on the performance of the system.

© University of Pretoria

ACKNOWLEDGEMENTS

This research was supported by the South African Research Chairs Initiative (SARChI) Research Chair in Advanced Sensor Networks, co-hosted by the University of Pretoria and the Council for Scientific and Industrial Research (CSIR) Meraka Institute. The financial assistance of the National Research Foundation (NRF) and the Council for Scientific and Industrial Research (CSIR) towards this research is hereby acknowledged. Opinions expressed and conclusions arrive at, are those of the author and are not necessarily to be attributed to the NRF and CSIR. This research was also supported by the Centre for Connected Intelligence (CCI) at the University of Pretoria.

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

LIST OF ABBREVIATIONS

BAN	Body Area Networks
CFS	Correlation Feature Selection
FFT	Fast Fourier Transform
FSA	Feature Selection Algorithm
GPS	Global Positioning System
HAR	Human Activity Recognition
LA	Left Ankle
LAD	Least Absolute Deviations
LSE	Least Square Errors
LT	Left Thigh
LUA	Left Upper Arm
LW	Left Wrist
MAE	Mean Absolute Error
MaxRel	Maximum Relevance
MFCC	Mel-Frequency Cepstral Coefficients
MID	Mutual Information Difference
MIQ	Mutual Information Quotient
ML	Machine Learning
MLSE	Mean Squared Logarithmic Error
mRMR	Minimum-Redundancy Maximum-Relevancy
MSE	Mean Squared Error
OLS	Ordinary Least Squares
OvO	One-vs-One
OvR	One-vs-Rest
RA	Right Ankle
RBF	Radial Basis Function
RMS	Root Mean Square
RSSI	Received Signal Strength Indicator
RT	Right Thigh
RUA	Right Upper Arm
RW	Right Wrist
SVM	Support Vector Machine
WSN	Wireless Sensor Networks

© University of Pretoria

TABLE OF CONTENTS

© University of Pretoria

© University of Pretoria

© University of Pretoria

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Context awareness is the ability of a system to use information about its environment and adapt its operations accordingly. The assumption of context-aware applications is that the context information assists in achieving the applications goal. The on-body device location, which in this research provides context to the activity recognition problem, may be an example of imperfect contextual information [1]. Imperfect contextual information may lead to a negative impact on the activity recognition models' classification accuracy. Contextual imperfection is characterised into several types, with each having their own source, persistence and origin of inaccuracy. The first source of contextual imperfection is called 'sensed', which is obtained from physical sensors, has low persistence and its inaccuracies arise from sensor failures, delays and misinterpretations. The next type is called 'static', which is obtained from a user, has infinite persistence and is subject to human error.

1.1.2 Research

Several studies have been conducted which examine localisation or activity recognition individually using various methods. One study has examined the combination of the two ideas of localising and activity recognition. Each study has been performed using their own choice of locations, activities, features, feature selection algorithms and classifiers. For this reason there is a large discrepancy between the results of the studies, with very little in common between them with which to compare their results.

© University of Pretoria

This study sought to use a multi-faceted approach by combining several feature selection algorithms, several classifiers, a generic set of activities, a generic set of on-body device locations and then using these combinations in several machine learning models. The results for each model can then be compared to determine each combinations efficacy. The effect of applying the location as contextual information was investigated in terms of the two types of imperfect contextual information as mentioned previously. These types will be applied differently for each model.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The objectives of the study include the implementation and comparison of various activity recognition models, with each of the three models using the on-body device location differently. A final model will be developed which autonomously determines the on-body device location and then performs activity recognition in the same manner as the best performing of the first three models. The implementation of these models contains several steps including data collection, feature extraction, feature selection and finally classification. Once all models have been implemented, comparisons will be made using their classification accuracies, standard deviations, features, feature selection algorithms and classifiers. The following questions will be investigated during the research:

- Is there a noticeable difference in activity recognition accuracy when using the on-body device location in various ways?
- Is there a list of features that can completely accurately determine a users activity, regardless of on-body device location?
- Is a fully automated system capable of achieving results that closely compare to that of a pre-programmed activity recognition system?

1.3 HYPOTHESIS AND APPROACH

To determine the impact of localisation of nodes for activity recognition, the selected approach was to use three different machine learning models, each with varying amounts of known location information. Each model with location information (either known or unknown) will include the on-body location in one of three ways:

© University of Pretoria

- Model 1 will not use the location in any form.
- Model 2 will have the encoded location appended to the feature space as a feature.
- Model 3 will limit the data used in the training dataset based on the location.

Each model will be tested using various combinations of feature selection algorithms and classifiers, to remove as much bias as possible, to ensure that results are objective. The results of each combination in each model will be compared with each other to reveal some answers of research questions. Once this initial testing is complete, Model 4, will localise the device autonomously and then perform activity recognition using the location in the same manner as the best performing of the first three models. Finally the results of all 4 models will be compared.

1.4 RESEARCH GOALS

The goals of the research are to generate a robust set of results that may give some clarity on the various unknowns of activity recognition and localisation of wearable devices, such as feature subsets, classifier and feature selection algorithm combinations. A final goal is to produce a fully automated system capable of generating results that are competitive against systems with prior knowledge or context of the user or environment. By evaluating the results generated by each model, a better understanding of the knowledge gain, contribution and effectiveness of the application of a devices on-body location as contextual information in activity recognition problems, can be obtained.

1.5 RESEARCH CONTRIBUTION

This research is intended for publication and has been submitted to IEEE Sensors for peer review: de Arruda, D., Hancke, G. and Myburgh, H. (2019). A comparative analysis of the effects of wearable device localisation on human activity recognition. IEEE Sensors, PP.

1.6 OVERVIEW OF STUDY

This dissertation is divided into six chapters. Chapter 2 is a literature study investigating the current states of technology, as well as design considerations of systems of this type. These themes include

© University of Pretoria

sensors, machine learning, wearable devices, human activity recognition and localisation. Chapter 3 explicates the methodology of the study, relating to the implementation of the several stages of all the machine learning models. These stages include data collection, feature extraction, feature selection and classification. The implementation of the structure of the machine learning models, experimental procedure, approach and generation of results are also included in this chapter. Chapter 4 presents the results generated by each machine learning model, including an examination of the performance of the features, feature selection algorithms, classifiers, activities and the effect of localisation on the performance of the system. Chapter 5 presents an analysis of the results presented in Chapter 4, by considering the various facets of the experiment. Finally, Chapter 6 presents conclusions of the study linked to the analysis and discussion presented in Chapter 5 as well as final comments regarding the research questions of the experiment.

© University of Pretoria

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

This chapter examines the various facets related to the topic of investigation. Various hardware and software are discussed which highlight various requirements of all systems of this nature. A discussion of sensor hardware characteristics and design choices is in Section 2.2. In Section 2.3, machine learning is defined and various algorithms and types of machine learning are reviewed. The current state and challenges faced by wearable devices is explored in Section 2.4. Types and factors to consider are examined for current day human activity recognition in Section 2.5. Section 2.6 contains an overview of approaches to localisation of devices.

2.2 SENSORS

Sensors are devices that are capable of detecting physical phenomena within the direct environment of their placement. Essentially a sensor is a transducer that converts some form of signal or stimulus, into a signal representing information about the system. Types of sensors in broad categories include: Proximity, Thermal, Force, Pressure, Optical, Navigational, Ionising radiation, Flow, Environmental, Electrical, Chemical, Automotive and Acoustic. Sensors have significant properties which determine their quality; These properties are listed in Table 2.1 and 2.2:

Sensors are also divided into categories of Active and Passive sensors. The distinction between the two categories is quite important to the design of a system as each have their own requirements for operation. Active sensors operate similarly to other electrical devices such as Operational Amplifiers or Transistors, where their operation requires the use of an external power supply or excitation voltage.

© University of Pretoria

Table 2.1. Characteristics of Sensors

Characteristic	Description
Range	Every sensor has a minimum and maximum value of the parameter it is able to measure. This means that if the input is not in the range of the sensor, the output will not be accurate.
Drift	As a sensor ages or heats up, the input value will not correspond exactly to the expected output, meaning that there will be an error in the measured values. This minor shift in expected value to measured value is the drift.
Sensitivity	Sensitivity is defined as the amount the output changes per unit of the inputs change. Sensitivity of the sensor may be constant or exponential depending on the composition of the sensor.
Selectivity	Selectivity is a characteristic referring to the sensors ability to not be affected by another property other than the one it is measuring. An example of bad selectivity would be a speed sensor being affected by air pressure.
Resolution	A sensors resolution is the smallest change in the input property the sensor is able to detect.
Response and Recovery Time	The response time of a sensor is a measure of the time it takes to react to a change in the input property from zero. The recovery time is the measure of time the sensor takes to react to a change in the input property to zero.
Linearity	Linearity is an expression to determine the extent to which the sensitivity of the sensor curves. If the sensitivity remains constant then the sensor is linear, but if the sensitivity curves it is then non-linear.

Passive sensors however are more similar to Resistors, Inductors or Capacitors. These components generate signals without any excitation voltage [2, 3].

Sensors can also be either digital or analogue. The choice of this depends on the application. Digital sensors could for example output a binary representation of a signal, whereas an analogue signal will

© University of Pretoria

Table 2.2. Continuation of Table 2.1

Characteristic	**Description**
Hysteresis	Hysteresis of the sensor is a measure of the sensor's outputs dependant on the input parameter increasing or decreasing. If the input parameter is increasing from zero, the output may not be exactly equal if the input parameter is decreasing to zero at the same points.
Precision	Precision is the ability of the sensor to be able to produce the same output when repeatedly exposed to the same input. Ideal sensors would produce an identical output for every measurement with the same input.
Accuracy	The accuracy of the sensor is defined as the closeness of the output to the expected real value.
Offset	The offset of a sensor is a measure of the output when the input as at zero. The output of an ideal sensor should correspond exactly to a zero input.

output a smooth, continuous signal.

Notwithstanding all the characteristics, types, classes etc of sensors, choosing a sensor is affected by various other factors of the system can which include [4]:

- Physical: This includes considerations in terms of power requirements, weight and size limitations.
- Operating conditions: The main contributor to this characteristic is environmental conditions which include factors such as temperature.
- Immediate data: Characteristics of the data which may include resolution, framerate, latency etc.
- Capabilities: This is a consideration of the functional applications of the raw and derived data obtained.
- Calibration: A consideration of the individual and joint sensor calibration information or algorithms.
- Confidence: An important consideration that can ultimately decide whether a system is worth using.

© University of Pretoria

2.3 MACHINE LEARNING

Machine Learning (ML) is a paradigm which refers to a method of improving performance of solving a problem, by learning from past experience. ML is a branch of artificial intelligence which operates on the assumption that given some data, a machine can learn how to solve a problem. An ML model will automatically learn without external influence from a human [5]. These types of systems are created with a specific purpose. It is basically a combination of computer science, statistics and mathematics. Advancements in the field of ML have inspired many applications or uses in various other fields. Fields such as speech recognition, computer vision, surveillance, security, automation control or empirical science experiments are being adapted or make use of ML in some shape or form. The scope of ML is becoming overwhelmingly large and has inspired many ideas of how to train a model to produce better results [6]. There are several methods in which an ML model can learn, which include:

- Supervised Learning: Training data is already labelled, meaning the result is known with its corresponding input.
- Unsupervised Learning: Training data has no label, meaning the results are up for interpretation by the ML model itself.
- Semi-supervised Learning: Training data for this type of learning has both labelled and unlabelled data.
- Reinforcement Learning: This type of learning relies on a reward function, where choosing some action provides the maximum reward based on the input data.
- Learning to Learn: This type of learning, or meta-learning, applies ML to meta data extracted from ML experiments, with the goal of understanding how to adjust the bias introduced, to learn better, hence the term ’learning to learn’ [7].

Various algorithms have been designed to physically train a model to predict or classify a target variable. Some of the algorithms are only applicable to certain types of ML models as described above. These algorithms include [8, 9]:

- Regression: This type of algorithm analyses the relationship between the target and input variables. Examples of this type of algorithm amongst others include, Linear Regression, Ordinary Least Squares Regression and Stepwise Regression.

© University of Pretoria

- Instance-based: This type of algorithm stores instances of training data to develop a definition for the target. When the algorithm receives new data it determines a result based on previously previously stored data. Examples of this include k-nearest-neighbours and vector quantization.
- Regularisation: This algorithm tries to counteract the effects of over-fitting. This is typically paired up with regression algorithms. Examples include Ridge Regression and Least-Angle Regression.
- Decision Tree: This algorithm creates a tree like structure solving problems based on constraints. Once a decision is made, it reaches a new set of decisions, until a final decision is made. Example of this include C4.5 or Classification and Regression Trees.
- Bayesian: This algorithm uses Bayes theory to determine a statistical likeliness between an input and a target. Examples of this include Naive Bayes and a Bayesian Network.
- Clustering: This algorithm finds patterns in datasets and creates clusters based on those patterns. Examples include K-Means or Spectral Clustering.
- Neural network: This is modelled after a human brain, passing weighted combinations of inputs through activation functions to generate a result. Examples of this are Multilayer perceptrons and Recurrent Neural Networks.
- Dimensionality Reduction: This type of algorithm is used to reduce the complexity of a dataset while still maintaining its ability to correctly classify data. This is typically used in conjunction with supervised learning to improve its speed, accuracy and slightly reduce over-fitting due to the dataset containing less irrelevant data. Examples include Principle Component Analysis and Correlation Feature Selection.
- Ensemble: This algorithm integrates several models to increase the robustness over an individual model. An example f this is Bootstrapped Aggregation, more commonly known as Bagging.

The combination of model and algorithm is dependant on the problem that is being solved, or the data that is available in the system.

Although these methods are actively being researched, over-fitting is a problem that still plagues the world of ML. Over-fitting occurs when a problem is too complex to solve given some dataset. A model will fit the data too closely, while the data may contain noisy and outlier samples. This results in a poor generalisation and the ML algorithm will therefore give inaccurate predictions with new or unseen data. There are various methods which can be used to combat over-fitting, one of which

© University of Pretoria

includes splitting data into a training and testing set [10]. This split is typically 70% for training and 30% for testing. For this method, training the model is done using the training set and validation or testing is done using the testing set.

Another method to counter over-fitting is cross-validation, where a model is trained n times where n is the number of subsets of a dataset, where each subset is used for validation and $n-1$ subsets are used for training. This approach tries to make use of as much data as possible to train the model. Another more brute-force approach to combat over-fitting is training various algorithms and comparing the performance using an independent dataset. Once the training and testing has been completed, the best performing model is chosen for use in the system.

Studies performed all make use of instance-based algorithms while making use of dimensionality reduction techniques, using several combinations of classifiers such as k-Nearest Neighbours, Support Vector Machines, Hidden Markov Models and Artificial Neural Networks, to increase the effectiveness of activity recognition system [11–14].

2.4 WEARABLE DEVICES

As smartphones penetrate the worlds markets and are becoming increasingly present in people's lives, the trend of physically wearing smart devices is also increasing. These devices are often referred to as wearable devices or simply, wearables. Examples of these devices include smart phones, smart glasses, smart watches and even E-Textiles [15].

The purpose of these wearable devices is to typically collect or sense parameters of the users environment and may often help the user to perform small tasks such as checking text messages or other viewing information. Example applications include navigation, banking, health monitoring and even sport analytics. These applications are typically designed to make the user's task easier. It is estimated that in 2019, over 200 million wearable devices will be in use worldwide [16].

With the rise in wearable technologies, there has been a rise in the research of the field. This gave rise to the term Body Area Networks (BAN). A BAN device is typically embedded in or on a person's body and is designed to connect these devices to the internet. Once again there are many applications

© University of Pretoria

for this technology such as remote-health monitoring [17]. There are various challenges that BANs face including:

- Security: Data collected from users may be confidential or private. Therefore it's important to protect that information which may only for certain users. There are various ways to secure a network by ensuring confidentiality, authentication and integrity of data being transferred.
- Interoperability: It is reasonable to assume that at a certain point in the future, people may be wearing multiple BANs simultaneously, and interoperability will ensure that should it be necessary, data would be able to transfer seamlessly between devices and or networks.
- Data Management: As BANs are able to sample large amounts of data, the processes for handling and maintaining it is of utmost importance to ensure its integrity.
- Price: It's expected that for users to be able to make use of technologies in BANs, they must be affordable so that not only an elite few are able to afford them.
- Deployment: BANs are required to be lightweight and non-intrusive so that users are not hindered during their normal activities.
- Performance: Ensuring that BANs operate consistently regardless of environment, including sensor accuracy and robustness of the network.

Some of these issues listed amongst others are addressed by the IEEE 802.15.6 standard for Wireless Body Area Networks. However, not all challenges have been addressed because the field is ever expanding.

2.5 HUMAN ACTIVITY RECOGNITION

Human Activity Recognition (HAR) is a field which has gained particular interest in medical, security and military applications. For example, patients who suffer of heart problems may be required to follow certain exercise regiments which form part of their recovery or treatment. This means that the ability to remotely monitor someone's activities such as walking or cycling can provide vital information to an interested party, and may give context to sensor readings which may not have been previously understood.

There are two categories of HAR systems [12]:

© University of Pretoria

- Vision based activity recognition - This is a classic activity recognition method which relies on appropriately placed cameras to capture human activity information. This method uses computer vision techniques. This technique is very susceptible to external interference such as variable lighting or even amount of people in a frame. It may also be computationally expensive depending on the quality of images that need to be processed.
- Sensor based activity recognition - This type of activity recognition relies on the deployment of sensors in an environment. This type of activity recognition is made up of two sub-categories, namely:

  1. Wearable sensor based activity recognition - This type of AR requires sensors to be physically attached to a person. The type of sensors attached can vary widely depending on the type of activities which need to be identified.
  2. Object usage-based activity recognition - This type of AR attaches sensors to an object with which a person interacts. This type of AR does not explicitly define an activity like the wearable sensor's alternative, it instead infers the activity being performed based on the object being used. For example, if a person is using weights, it can be assumed they are exercising.

The type of activity recognition system to be implemented generally has to consider the factors shown in Table 2.3 and Table 2.4 [11]. Several studies have all made use of wearable sensor activity recognition [11–14], with these same studies monitoring activities such as drinking, closing windows, chopping, stirring a bowl, playing tennis, walking, jogging, running, sitting, lying, standing and climbing up and down stairs. None of these studies individually included combinations of all types of activities (periodic, sporadic and static). All of the systems described are all executed offline, with either a stateful or stateless model. They are also typically user independent, with segmented recognition.

2.6 LOCALISATION

Localisation is the term used for finding the geographical position of a device in an environment. There are several examples of this technology with the Global Positioning System (GPS) being an example. This example uses satellites to determine the position of the device world-wide and is used

© University of Pretoria

Table 2.3. Factors to consider in a Human Activity Recognition system

Execution	Offline	The system first records sensor data and then activity recognition is performed. This is typically used in applications that are non-interactive such as e-Health.
	Online	The system collects and processes sensor data in real-time. This is typically used in activity-based computing or human and computer interactions within interactive applications
Generalisation	User Independent	The system is capable of operating effectively with many users.
	User Dependant	The system is designed for a specific user. This typically increases the performance in comparison to a user independent case, however it does not work effectively for other users.
	Temporal	The system should compensate for temporal variance introduced by external factors, such as drifting sensor response and sensor displacement.
Recognition	Continuous	The system is able to distinguish occurrences of activities within a sensor data input stream.
	Segmented	The sensor data is segmented at the beginning and end of an activity by a third-party. The model classifies the segmented sensor data into an activity class. The third-party may be an external system or a user who is conducting the experiment.

for navigation. Localisation by extension also has many applications in Wireless Sensor Networks (WSN), such as environmental, industrial and medical monitoring applications [18].

Typically, localisation of devices or nodes in a WSN is performed using Radiolocation, which is the process of determining the location using radio waves. There are several techniques to this approach such as using the Received Signal Strength Indicator (RSSI), Time of Arrival (TOA) or Angle of Arrival (AoA). A disadvantage of using this type of localisation is the requirement of having fixed reference nodes with which to make estimates. In the context of BANs, it is implausible to think that someone using a BAN in their everyday lives would remain in a position that is reachable by these fixed reference nodes. It is for this reason that in the use of localisation in BANs that the developer(s)

© University of Pretoria

Table 2.4. Continuation of Table 2.3

Activities	Periodic	This describes activities which are periodic in nature, such as biking, running or walking. For classification, spatial features and/or sliding window segmentation are used.
	Sporadic	These activities are scattered amongst other activities and occur at irregular intervals. Segmentation is a key factor as it allows for the isolation of the activity within the data.
	Static	These activities are static in nature, meaning there is no movement involved. Examples include lying down or sitting.
System Model	Stateless	The system does not include any information regarding the state of the environment in which it is placed. The activities are classified only by recognising specific signals from the sensor.
	Stateful	The system models the environment and makes use of contextual information such as the location of objects, to enhance the performance of the activity recognition model.

makes similar considerations to that of HAR, such as Execution, Generalisation, Recognition and System Model.

Typically, the accuracy or application specific requirements typically dictate the approach that will or will not be used to meet those requirements. Localising nodes in a BAN is no exception, meaning that if the desired outcome is to determine the location of the nodes in a specific environment, then radiolocation is a possible solution to the problem. However, if it is required that the results be more specific to determine the exact body part which a BANs nodes are placed, then perhaps a different approach is necessary.

The added flexibility of placing nodes on a person's body means that these devices may move while they perform any given activity. As in [19], the objective was to determine the on-body location of nodes with the goal of monitoring vital signs on a patient's body. The decided upon approach was to measure atmospheric air pressure compared to other nodes without the requirement of fixed reference nodes. Another example of on-body device localisation in [20] uses inertial sensors to find patterns in the motion of the BAN node while the user performs activities to determine its on-body location.

© University of Pretoria

Two other studies place the wearable devices on the chest, wrist, hip, thigh, foot, and arms [11, 12]. Two other studies have placed the wearable devices in fixed locations such as the belt or shirt pocket, making the assumption that those locations provide the most information [13, 14].

2.7 CHAPTER SUMMARY

This chapter begins by investigating various aspects of sensors including characteristics, design considerations and the various types of sensors available. This is followed by an in-depth look into the various state of the art algorithms, methods and challenges faced when implementing ML. Furthermore, the various applications and factors of both wearable devices and HAR models are discussed. Finally, the design considerations and techniques of device localisation are explored.

© University of Pretoria

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

This chapter discusses the implementation and approach to be used to investigate the research topic in detail. The research contains several machine learning models which are further discussed in Section 3.2. The various steps for collecting training data are explained in Section 3.3. A discussion on the approach to data pre-processing can be found in Section 3.4. In Section 3.5 the methods for feature selection are expanded upon. A review of various classification algorithms appears in Section 3.6. Finally the approach to generating results is discussed in Section 3.7.

3.2 MACHINE LEARNING MODELS

This section discusses four configurations of machine learning models with varying amounts of information known about the location of the device. Each model is separated into two phases, the training and execution phase. The training phase is used to gather training data, which is then used to develop the classifiers understanding of the problem. The execution phase is a simpler implementation of the training phase, because only relevant data is extracted and processed using the already trained classifier. The execution phase can therefore be viewed as less computationally expensive. The design considerations of the various stages of each model are discussed in Chapter 2.

3.2.1 Model 1

Model 1 which is shown in Figure 3.1 is a basic machine learning model. In the training phase, data is collected as described in the experimental setup. After pre-processing, features of the data are

© University of Pretoria

extracted and then the feature selection algorithm decides on the relevant features; These relevant features are then used to train the classifier. Once training is complete, the system transitions into the execution phase. During the execution phase, data is collected from randomly placed nodes on the body. Feature extraction in this phase is different to the training phase. In the training phase, there is no indication of which features prove the most useful, therefore, a large list of features are extracted and processed. However, in the execution phase the feature selection algorithm has been trained to know which features are relevant. This allows the system to only extract features which are relevant to the problem and ignore all others. These relevant features are then passed into the already trained classifier, which then determines the activity being performed.

The data being collected and processed in Model 1 has no knowledge of the on-body location of the nodes. All the data collected is combined into a single, large data set. This means that the feature selection algorithm must consider a large amount of data, from various locations during all activities and must find distinguishing features that are independent of the location of the node. Since this model has the least prior knowledge and a large data set, it is expected that this model will yield the lowest average classification accuracy.

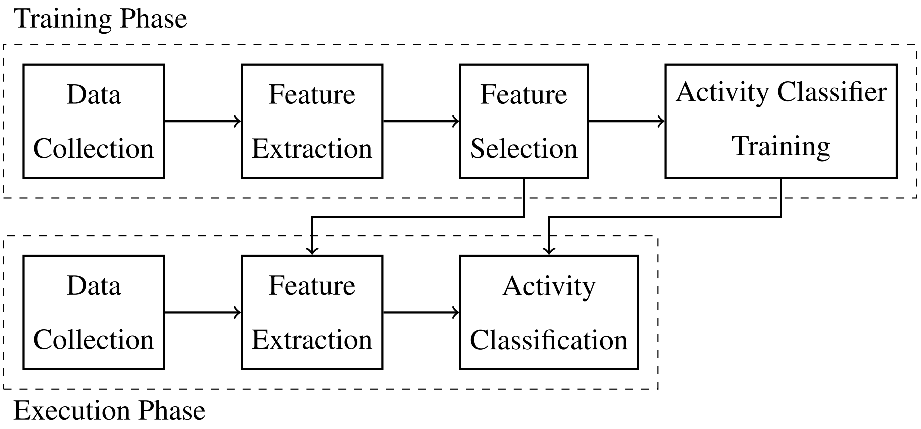


Figure 3.1. Block diagram for Model 1.

3.2.2 Model 2

Model 2 which is shown in Figure 3.2 is an extension of Model 1. The structure remains the same as Model 1 up until the feature selection stage in the training phase. Before the feature selection stage in the training phase, the known location of the sensor node is appended to the extracted feature dataset

© University of Pretoria

as an encoded integer representation of the location. This was achieved by representing the Left Upper Arm as 1, Right Upper Arm as 2, etc, until the Right Ankle was represented as an 8.

This approach was used because the classifiers chosen are only capable of processing real-valued input data. The feature selected dataset including the encoded device location is used to train the classifier. During the execution phase the same logic is followed, however the user would be expected to configure the node with the device location before use. This allows the extracted features to be appended with the corresponding integer representation of the known location of the device.

Similarly to Model 1, this model also makes use of a single large dataset for training. This model uses the same extracted features as Model 1, with the exception of an added column of data, containing the location of the device; As a result Model 2 contains more contextual information than Model 1. Even with a slightly more complex dataset, intuition suggests that this model will yield a higher classification accuracy than that of Model 1.

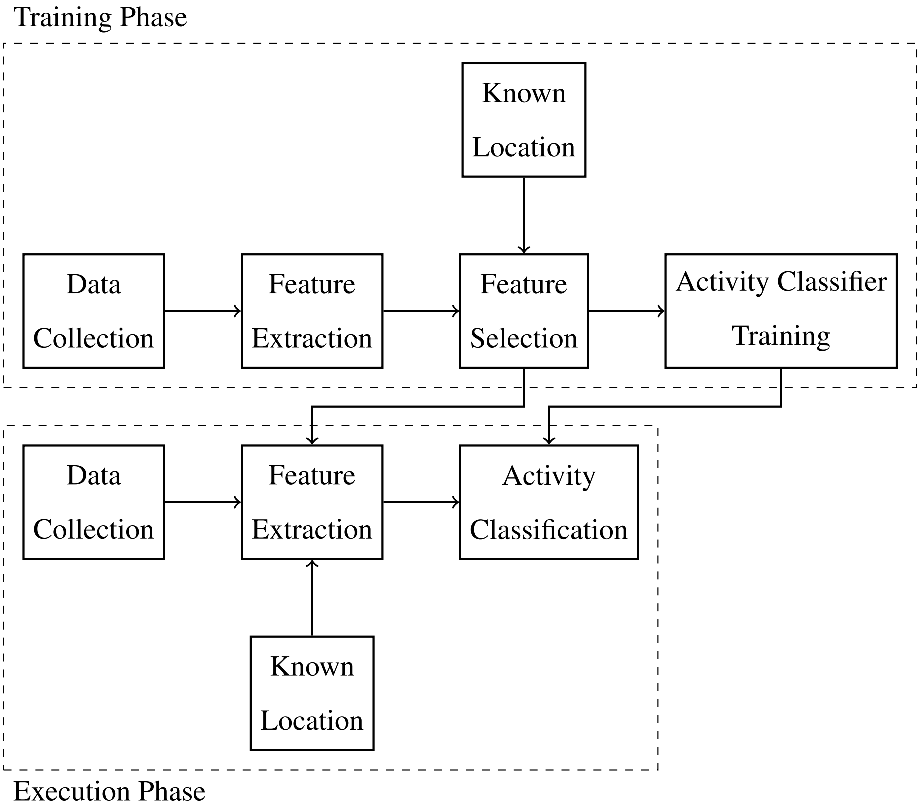


Figure 3.2. Block diagram for Model 2.

© University of Pretoria

3.2.3 Model 3

Model 3 shown in Figure 3.3 is a further extension on Model 2. The premise behind this extension is that no single feature selection algorithm and classifier combination would completely accurately model all the activities regardless of the on-body device location. Intuitively it could be said that each node's location may require a different feature selection algorithm and classifier combination to correctly classify the activities. An example of this may be using Correlation Feature Selection (CFS) with a Support Vector Machine (SVM) to model activities of the Right Ankle with 100% accuracy, but if the node is instead attached to the Left Wrist, it may only produce results up to 70% accuracy. The assumption is that there exists some combinations of feature selection algorithm and classifier that can model the activities for each on-body device location with accuracies of 100%.

In this model, instead of using the on-body location as a feature in the dataset as in Model 2, the location is used rather to determine which dataset is used for training and which features and classifier are used for execution. This means there will be eight training datasets, each corresponding to an on-body device location with each dataset containing data for all activities. Similar to Model 2, this model requires the user to configure the node for the execution phase so that its location is known. It is expected that this model will yield the highest average classification accuracy, because the best performing combination of feature selection algorithm and classifier is used for each specific on-body device location.

3.2.4 Model 4

The final model, Model 4 is an attempt to replicate as closely as possible the results of the best performing of the previous models without the need for the user to configure the node before the execution phase. This model will therefore determine the location of the node and then subsequently pass that location information to the best performing activity recognition model. Model 4 localisation is comparable to Model 1, however, instead of the output being an activity, the output is the location of the node. Depending on which of the previous models is the best overall performer, Model 4 will make use of one or nine datasets.

© University of Pretoria

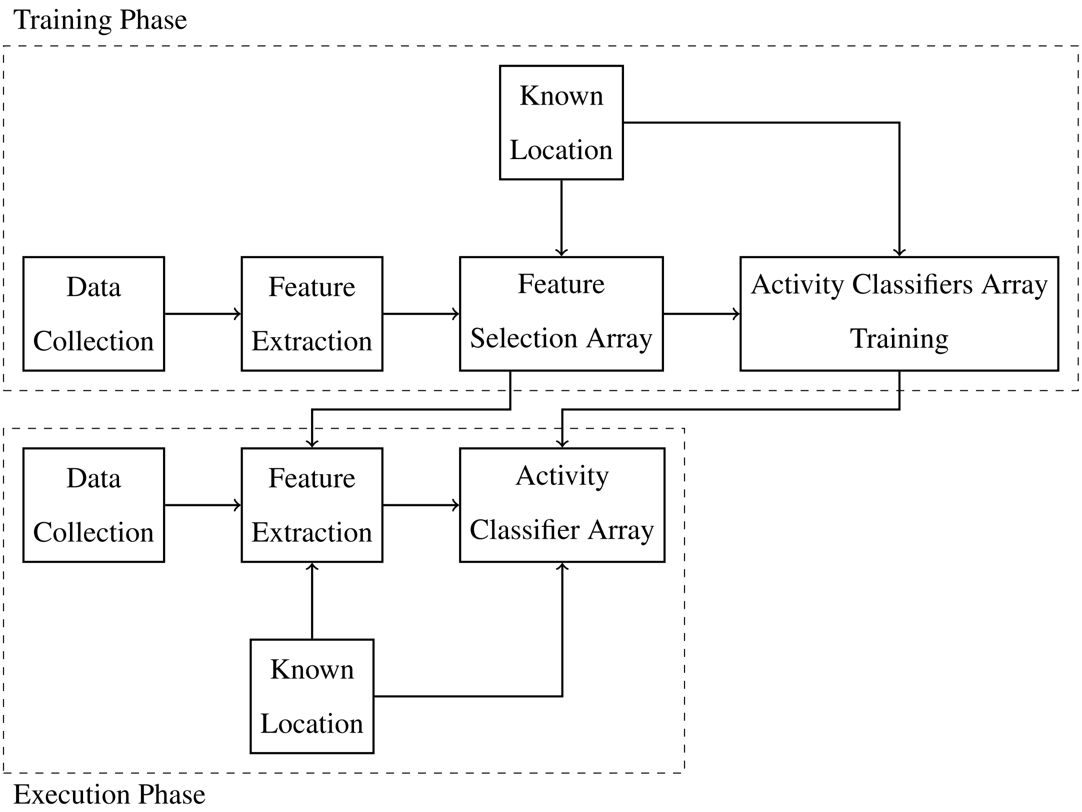


Figure 3.3. Block diagram for Model 3.

3.3 DATA COLLECTION

To begin the study, data from the various activities chosen needs to be sampled from test subjects. The only requirement of the wearable device from which data is collected, is that it contains a triaxial Accelerometer, Gyroscope and Magnetometer, or a 9 Degree-of-Freedom Inertial Measurement Unit. To do this, an Android application was created which samples the Accelerometer, Gyroscope and Magnetometer of a Samsung Galaxy S6 which was running Android Lollipop version 5.1 (API level 22).

To capture all data that may be relevant to the problem, it is important to consider how fast a human typically moves. A study has shown that an average human can take between 61 and 100 steps a minute [21]. From this it is possible to extrapolate that a sampling frequency of 2.02Hz to 3.33Hz is sufficient to meet the Nyquist sampling theorem criteria. The study also showed that while speed-walking, a person can take up to 300 steps in 1 minute, which effectively triples the sampling frequency required. This means that a sampling frequency of at least 10Hz is required to capture the motion of a

© University of Pretoria

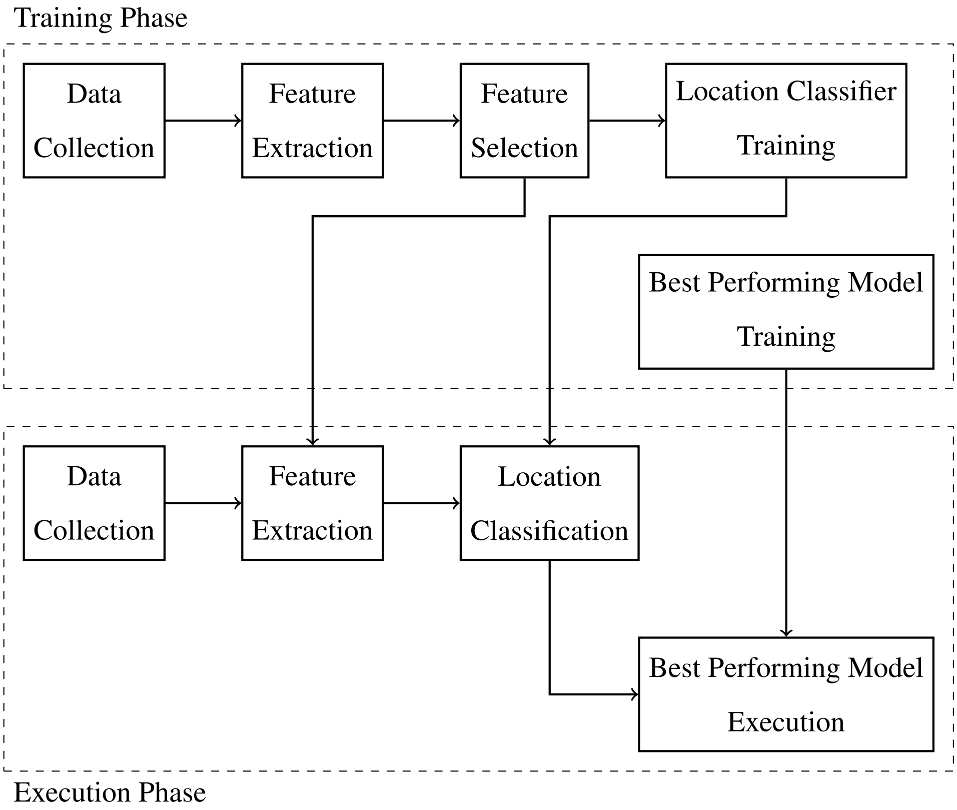


Figure 3.4. Block diagram for initial design of Model 4.

step.

In the Android API, it is possible to choose a sensor delay relevant to the purpose of the application [22]. This sensor delay is the interval at which sensor events are sent to the application. The sensor delays are defined as constants; in ascending order of delay these are: SENSOR_DELAY_FASTEST, SENSOR_DELAY_GAME, SENSOR_DELAY_NORMAL and SENSOR_DELAY_UI. These delays correspond to 0 microsecond, 20 000 microseconds, 60 000 microseconds and 200 000 microseconds respectively. Since it is known that a sampling frequency of approximately only 10Hz is required, SENSOR_DELAY_GAME with a delay of 20 000 microseconds was chosen. This corresponds to a sampling frequency of 50Hz, which will account for extreme cases where movements above a 10Hz sampling frequency threshold would be insufficiently recorded.

In order to isolate the movements, instructions are sent from a server to start and stop recording sensor samples. This was achieved by creating a multi-threaded Python server, where the Android application established a connection to the server and awaits instruction. If the Android application receives a

© University of Pretoria

START instruction, the application stores the X, Y and Z samples of each of the sensors discussed above in memory after every sensor delay. The sensor orientations for the Samsung Galaxy S6 are shown in Figure 3.5. Once the application receives a STOP instruction, the application then writes the values stored in memory to a .csv file, then transfers it for storage on the server. This happens simultaneously for all devices with a connection to the server, as each connection is maintained in its own thread. This stored data is then labelled according to the device location and activities performed, then awaits further processing.

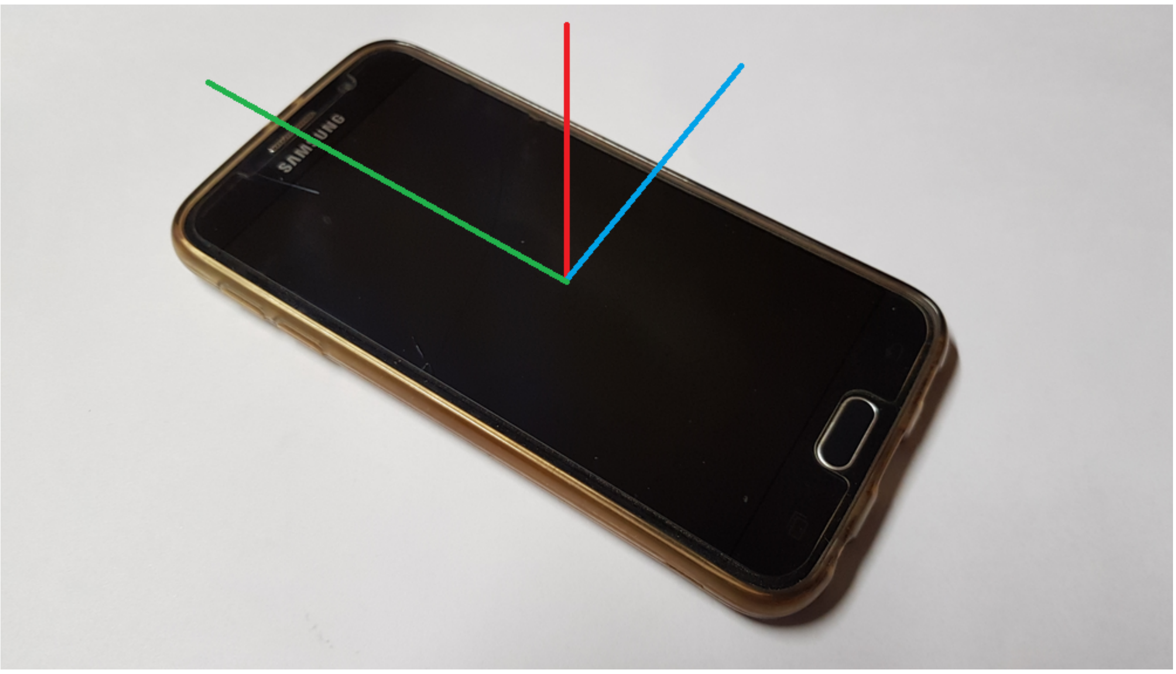


Figure 3.5. Samsung Galaxy S6 sensor orientation indicating positive X (Blue, Pitch), Y (Green, Roll) and Z (Red, Azimuth) axes.

3.3.1 Experimental setup and procedure

Several experiments were conducted on various test subjects where each test subject had four Samsung Galaxy S6 smartphones attached to them using a smartphone armband cover that is strapped on either on the left or right side of their body, at the corresponding positions as shown in Figure 3.6. These positions (Upper Arms, Wrists, Thighs and Ankles) were chosen because they are the most common placements of wearable devices [23]. Once the smartphones were attached, each smartphone established a connection to the multi-threaded Python server over Wi-Fi, where the Wi-Fi access point and server was created and hosted on a Laptop. Once a connection was established the experiments could proceed.

© University of Pretoria

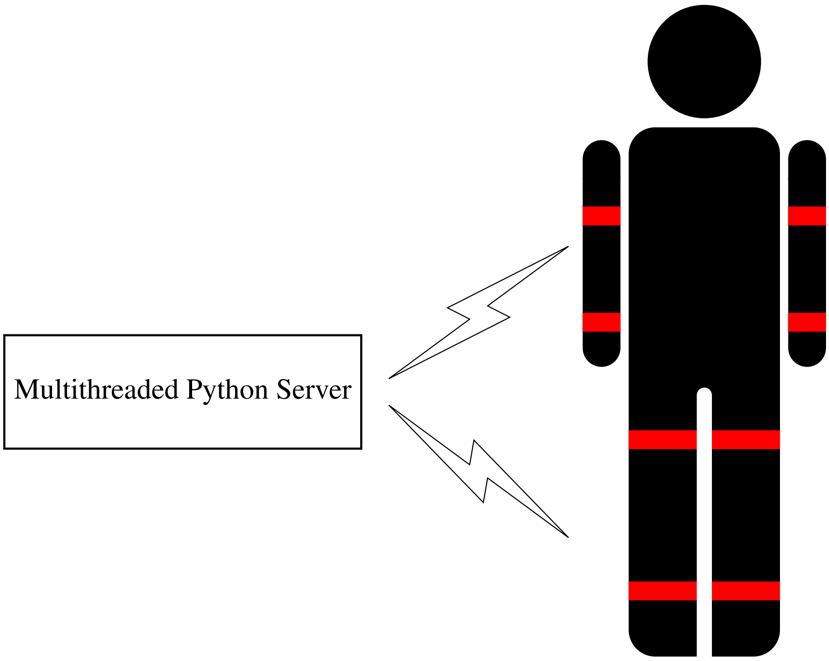


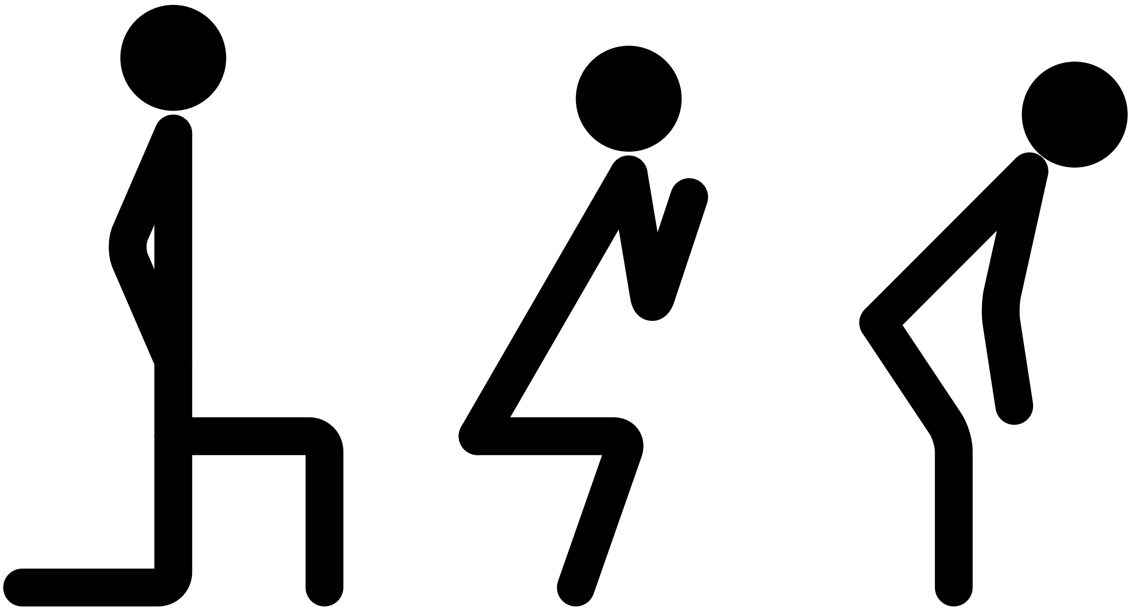
Figure 3.6. Python server with wireless communication to sensors placed on the upper arms, wrists, thighs and ankles.

It is important to note that each test subject obtained medical clearance to participate in these experiments. Each test subject was declared healthy and none suffer from any conditions such as Parkinson's disease, which could impair the movement of the individual. Each test subject would perform a series of movements or activities while the server sent instructions for the smartphones to start and stop recording. The test subjects were asked to perform each movement three times to allow for multiple recordings of the same movement. The test subjects were asked to lunge with their right foot forward, lunge with their left foot forward, squat, hinge, walk, jog, run, stand still and lying down. The ideal poses for lunges, squats and hinges can be seen in Figure 3.7. Once all the activities were completed, the smartphones would be taken off and placed on either the left or right side of the test subject, depending on which side had already been measured. This series of tests would be completed on all test subjects.

3.4 PRE-PROCESSING

Before the data is classified, pre-processing needs to be applied to the sampled sensor signals. This section discusses the steps involved in processing the signal, which includes a discussion on applying

© University of Pretoria



(a) Lunge **(b)** Squat **(c)** Hinge

Figure 3.7. Ideal poses for lunges, squats and hinges.

a sliding window and a discussion on the mathematical formulas used to determine discrete temporal or spatial features of the sampled signals.

3.4.1 Time-shifting

A sliding window or windowing, is a technique which is adapted from the Internets Transmission Control Protocol (TCP/IP) as a method of controlling the flow of packets between two nodes on a network[1]. By using this technique, algorithms are able to evaluate portions of a signal independently of the remaining portion of the signal. This technique is useful to introduce the ability of the system to function in a time-delayed environment with signals of unknown length.

A sliding window has a set width, and slides across a signal at a configured sliding interval. An example of this would be, if a device is attached to a persons leg, the data known to the system may be a time-delayed version of what the sensors are currently measuring. The system would potentially be unable to make a proper classification and therefore the systems accuracy could decrease. By using a sliding window and evaluating individual sections of the movement signal, it is then possible to

[1] http://searchnetworking.techtarget.com/definition/sliding-windows

© University of Pretoria

compare that to a windowed time-delayed training signal and therefore increase the system's accuracy. This is possible because each windowed section of the signal, still belongs to the leg class.

The Vector in Figure 3.8 is a row vector containing 7 randomly generated movement signal samples of a random human body part.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{bmatrix}$$

Figure 3.8. Vector containing 7 samples of a random movement.

If it is known that the sample rate of the system is 3 samples per second and the system is trained with 1-second long instances of samples, it is then possible to apply a sliding window to the original signal and extract every possible one second long instance from that signal. This is done by creating a window of width 3 and a sliding increment of 1. By applying this window to the signal and storing the results, the 3 x 5 Matrix in Figure 3.9 is generated. Each row corresponds to a 1-second long instance of samples in the movement signal and can be further processed individually.

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \\ 5 & 6 & 7 \end{bmatrix}$$

Figure 3.9. Matrix generated after time-shifting the vector in Figure 3.8.

As discussed in Section 3.3, the sensor delay was set to SENSOR_DELAY_GAME, which is a delay of 20 000 microseconds, corresponding to a 50Hz sampling frequency. For the purposes of this study a 1-second long window, with a sliding interval of 0.02 seconds (1 sample) was chosen. This means that each window will contain 50 samples, and will have a 98% overlap with adjacent windows.

© University of Pretoria

3.4.2 Feature extraction

Various studies have so far shown that no definitive list of features produce optimal results for localisation or activity recognition of the on-body devices [11,23–27]. These papers used various combinations of Mean, Variance, Skewness, Zero Crossing Rate, Median, Root Mean Square (RMS), Kurtosis, Mean Crossing Rate, Standard Deviation, Mean Derivatives, Interquartile Range, Maximum Energy, Mel-Frequency Cepstral Coefficients (MFCC), Spectral Entropy, Sum of Power Wave, Frequency Range Power, Fast Fourier Transform (FFT) Magnitude and Pairwise Correlation.

For this reason, the Mean, Variance, Median, RMS, Peak-to-Peak Amplitude, Average Power, Fundamental Frequency, Peak FFT Amplitude and Interquartile Range have been used as features to solve the problem, as it is uncertain which features are necessary.

The above mentioned features are extracted from a windowed signal and are combined into a unitless row vector as shown in (3.1). $\overline{\boldsymbol{F}_{XYZagm}}$ represents feature vectors of signals measured by the Accelerometer, Gyroscope and Magnetometer, in the X, Y and Z axes. Each combination of measurement device and axis direction form a separate feature vector. In the equation, μ indicates the Mean, σ^2 the Variance, Med the Median, X_{rms} the RMS, X_{pk-pk} the Peak-to-Peak amplitude, P the Average Power, f_0 the Fundamental Frequency, X_{FFT_A} the Peak FFT Amplitude and IQR the Interquartile Range.

$$\overline{\boldsymbol{F}_{XYZagm}} = \begin{bmatrix} \mu & \sigma^2 & Med & X_{rms} & X_{pk-pk} & P & f_0 & X_{FFT_A} & IQR \end{bmatrix} \tag{3.1}$$

Each independent feature vector combination as described above is ultimately combined into one large feature vector $\overline{\boldsymbol{FV}_i}$, which contains all the features from each sensor, in each direction, in one single vector; i is determined by the sample number and the sliding window length. The resultant feature vector $\overline{\boldsymbol{FV}_i}$ is shown in (3.2). This vector is a 1 x n*N vector, where n is the amount of features extracted from a signal and N is the amount of signals that have been measured. By substituting (3.1) into (3.2), $\overline{\boldsymbol{FV}}$ becomes a 1 x 83 vector.

$$\overline{\boldsymbol{FV}_i} = \begin{bmatrix} \overline{\boldsymbol{F}_{Xa}} & \overline{\boldsymbol{F}_{Ya}} & \overline{\boldsymbol{F}_{Za}} & \overline{\boldsymbol{F}_{Xg}} & \overline{\boldsymbol{F}_{Yg}} & \overline{\boldsymbol{F}_{Zg}} & \overline{\boldsymbol{F}_{Xm}} & \overline{\boldsymbol{F}_{Ym}} & \overline{\boldsymbol{F}_{Zm}} \end{bmatrix} \tag{3.2}$$

© University of Pretoria

3.5 FEATURE SELECTION

Dimensionality reduction is the term used to describe the process of reducing the complexity of a feature space, by reducing the amount of features under consideration. There are multiple benefits that can be gained by reducing the complexity of the dataset, of which the most important are, reducing the computational complexity, as well as potentially increasing the accuracy of the system. Feature selection is a process which examines features and determines their relevance or contribution to solving the problem and then filters out features deemed irrelevant.

There are three classes of feature selection algorithms (FSA) namely, Filter, Wrapper and Embedded Methods. Filter methods operate independently of a classifier, therefore presenting a set of generic features which have some correlation to the target variable. Filter methods are particularly effective in computation times and are robust against overfitting. Wrapper methods work differently to filter methods, because they work with a classifier to evaluate the strength of a chosen feature subset. There are two main disadvantages of using wrapper methods which are: Increased overfitting when there are too few observations and significant computation time when there are large numbers of variables [28]. Embedded methods are a hybrid between filter and wrapper methods to take advantage of their best attributes. For the purposes of this study, filter methods will be used as each subsystem is independent and the goal is to determine the relevance of the location to the problem and not rank the feature selection algorithms.

3.5.1 Correlation feature selection

Correlation Feature Selection is an example of a filter algorithm, which ranks subsets of features using a heuristic evaluation function based on correlation. It is based on Pearson's correlation coefficient and is developed to favour feature subsets that contain features with high correlation to a class and low correlation with other individual features [29]. The evaluation function output is known as the heuristic merit and is expressed in (3.3).

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{r_{ff}}}} \tag{3.3}$$

Equation 3.3 shows M_S, which is the heuristic merit of the feature subset, k is the amount of features in

© University of Pretoria

the feature subset, $\overline{r_{cf}}$ is the average feature to class correlation and $\overline{r_{ff}}$ is the average feature to feature correlation. The parameters $\overline{r_{ff}}$ and $\overline{r_{cf}}$ are calculated for each feature subset and are a measure of feature quality.

3.5.1.1 Feature quality

To quantify the quality of features, three different versions of CFS were developed. Each version makes use of a different quality measuring principle, namely, *Relief* (CFS-RELIEF), Symmetrical Uncertainty (CFS-UC) and Minimum Description Length (CFS-MDL). Each version of CFS has its own advantages and disadvantages when considering the amount of features, classes and sample size.

Symmetrical Uncertainty develops a model to estimate the probability of a value in a feature. It is biased towards features with many possible values and quantifies the amount of information gained after observing events. Entropy is the measure of uncertainty or information gained. The entropy of feature Y with value y is shown in (3.4) and the entropy of feature Y after observing feature X with value x is shown in (3.5). The information gain is then expressed in (3.6) and the symmetrical uncertainty coefficient is shown in (3.7).

$$H(Y) = -\sum_{y \varepsilon Y} p(y) log_2(p(y)) \tag{3.4}$$

$$H(Y \mid X) = -\sum_{x \varepsilon X} \sum_{y \varepsilon Y} p(y \mid x) log_2(p(y \mid x)) \tag{3.5}$$

$$gain = H(Y) - H(Y \mid X) \tag{3.6}$$

$$coefficient = 2 \times \left(\frac{gain}{H(Y) + H(X)} \right) \tag{3.7}$$

Relief is an algorithm that assigns weights to features and is particularly sensitive to interactions between features. It determines these weights by measuring the difference in probabilities of values for features in different classes and the probabilities of those values for the same class. Equation 3.8 shows a simplified version of determining the weights for each feature. The average of these weights is used to determine the correlations described in (3.3).

© University of Pretoria

$$Relief_X = P(X \mid DifferentClass) - P(X \mid SameClass) \tag{3.8}$$

The minimum description length principle compresses data as much as possible while still modelling a system, by reducing the length of the data. Operating similarly to symmetric uncertainty, it also uses entropy to measure the information gain. The MDL quality attribute is shown in (3.9), where $Prior_{MDL}$ is the description length of class labels prior to partitioning, $Post_{MDL}$ is the description length post partitioning and n is the number of training samples. Partitioning is performed by separating features based on values of another feature.

$$MDL = \frac{Prior_{MDL} - Post_{MDL}}{n} \tag{3.9}$$

A study between CFS-RELIEF, CFS-UC and CFS-MDL showed that CFS-RELIEF and CFS-UC gave better results for small datasets (Less than 200 samples). However, CFS-MDL gave better results with larger datasets [30]. For this reason CFS-MDL will be used, as the dataset contains several thousand samples.

3.5.1.2 Search strategies

To reduce the computational complexity of the algorithm, three search strategies were developed, which allows a user to decide how they wish to find the best feature subset. The three strategies are, Forward Selection, Backward Elimination and Best First.

- Forward Selection starts with a feature subset that is empty and adds a random feature at a time, while evaluating the merit, until no single feature addition results in a subset with higher merit.
- Backwards Elimination works inversely to Forward Selection, where it starts with a full feature subset and recursively eliminates features until the merit of the feature subset starts degrading.
- The best first search can search backwards or forwards through the feature space and is more exhaustive than Backwards Elimination or Forward Selection. It will stop searching the feature space until five successive fully developed subsets do not have higher merit than a previous best.

© University of Pretoria

There is no evidence suggesting that either search strategy outperforms another, so for this work Forward Selection was chosen.

3.5.2 Minimum-redundancy maximum-relevancy

Mimimum-Redundancy Maximum-Relevancy (mRMR) is a feature selection algorithm that attempts to rank features with high correlation to the target variable whilst having a low correlation between the features themselves [31]. It operates by first discretising the feature set and then applying various formulas to determine their rank. Given two random variables the mutual information is defined in terms of the probability density function $p(x), p(y)$ and $p(x,y)$, as shown in (3.10).

$$I(x,y) = \int\int p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{3.10}$$

Maximum Relevance is a search of features which approximates $D(S,c)$, which is the mean value of all mutual information between a feature x_i and class c in the feature set S as expressed in (3.11).

$$max\, D(S,c),\ D = \frac{1}{|S|} \sum_{x_i \in S} I(x_i,c) \tag{3.11}$$

If two features are highly dependant on one another, the class discriminating power of the classifier would not change if one of the features were removed. Therefore, Minimum-redundancy is defined in (3.12) to select mutually exclusive features where x_i and x_j are features of the feature set S.

$$min\, R(S),\ R = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i; x_j) \tag{3.12}$$

Feature ranking is done by using one of three formulas below, to determine Maximum-Relevance (MaxRel), Mutual Information Difference (MID) or Mutual Information Quotient (MIQ) which are represented in (3.13), (3.14) and (3.15) respectively [31, 32].

$$MaxRel = max\, \Phi(D,R),\ \Phi = D \tag{3.13}$$

$$MID = max\, \Phi(D,R),\ \Phi = D - R \tag{3.14}$$

$$MIQ = max\, \Phi(D,R),\ \Phi = \frac{D}{R} \tag{3.15}$$

© University of Pretoria

Given the parameters available when performing mRMR feature selection, a threshold for discretisation of 0.25 was chosen, as well as using all three feature ranking formulas, MaxRel, MID and MIQ to rank the features within the feature space.

3.6 CLASSIFICATION

Classification is the process of categorising an idea or object which is recognised. In Machine Learning a predictive model approximates a function to map input variables to a discrete output variable. In this section, various classifiers that use supervised learning are discussed. Each classifiers chosen configurations are then discussed in a later section of the chapter.

3.6.1 K-nearest neighbours

The k-Nearest Neighbour(s) (k-NN) classifier is possibly the simplest classifier to be implemented [33]. It operates by assigning a class to an unlabelled input feature vector based on a majority vote (out of k votes) from training feature vectors using a given metric. For this classifier there is no specific training phase, only the storage of training feature vectors in a feature space, with which to compare input feature vectors.

3.6.1.1 Metrics

For a k-NN classifier there are various metrics available, however each metric is intended for use with specific types of data within the feature space. The four types of metrics are applicable in real-valued feature spaces, angular two-dimensional feature spaces, integer-valued feature spaces and boolean-valued feature spaces [34]. Since the dataset for this work contains real-values, only real-valued feature space metrics will be considered. Examples of these metrics are the Euclidean distance, Manhattan distance, Chebyshev distance, Minkowski distance and Mahalanobis distance [35]. In the following set of equations, **X** represents an unlabelled input vector and **Y** represents a labelled training vector.

- Euclidean Distance is the length of the line-segment between points in a space. It is represented in (3.16).

© University of Pretoria

$$d(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{3.16}$$

- Manhattan distance is based on a grid, with multiple paths to traverse between points. The distance itself is calculated as the sum of differences of each component in the vector and is represented in (3.17).

$$d(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{N}\Big(|x_i - y_i|\Big) \tag{3.17}$$

- Chebyshev distance is defined as the greatest difference between two coordinates in any dimension. It is represented in (3.18).

$$d(\mathbf{X},\mathbf{Y}) = \max_{1\leq i\leq N}\Big(|x_i - y_i|\Big) \tag{3.18}$$

- Minkowski distance is a generalised formula for distance, which can be defined as the power-mean of the difference between vectors. For $p = 1$, the distance is equivalent to the Manhattan distance, for $p = 2$ the distance is equivalent to the Euclidean distance and for $p = \infty$ the distance is equivalent to the Chebyshev distance. For the Minkowski distance to be usable as a metric, p must be larger than or equal to 1. It is represented in (3.19).

$$d(\mathbf{X},\mathbf{Y}) = \left(\sum_{i=1}^{N}|x_i - y_i|^p\right)^{\frac{1}{p}} \tag{3.19}$$

- Mahalanobis distance is a metric of how many Standard Deviations away a vector is from the Mean of a distribution. It inherently takes into consideration the correlations of a dataset. Equation 3.20 shows this distance, where $\vec{\mu_y}$ is the vector of means for each training feature and S^{-1} is the inverse covariance matrix. If the covariance matrix is the Identity Matrix, then the Mahalnobis distance is equivalent to the Euclidean distance.

$$d(\mathbf{X}) = \sqrt{(\mathbf{X} - \vec{\mu_y})^T S^{-1}(\mathbf{X} - \vec{\mu_y})} \tag{3.20}$$

There is no evidence that suggests any particular one of these metric outperforms another in any given scenario, thus the Euclidean Distance has been chosen for this research [35].

© University of Pretoria

3.6.1.2 Decision making

As stated previously, k-NN does not have a specific training phase and makes a classification based on a vote count. When an input vector needs to be classified, the algorithm determines the distance to every single training vector that has been stored, making it an instance-based learning algorithm. This makes it an exhaustive algorithm, therefore, using very large datasets can be computationally expensive. The voting strategy employed by this algorithm counts the k closest training vectors to the input vector and labels the input vector based on a majority vote. It as been shown that for $k \geq 3$ there is no significant increase in classification accuracy [36].

As the algorithm progresses, it implicitly builds decision boundaries in the n-dimensional feature space, where each hypersurface contains a class. These hypersurfaces can be visualised as a Voronoi diagram. An example of this is shown in Figure 3.10, where there are six classes in a two-dimensional space.

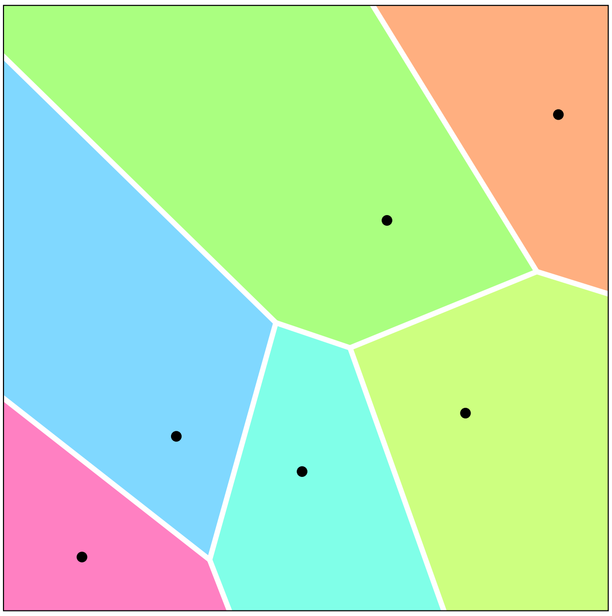


Figure 3.10. Example Voronoi diagram of a random two-dimensional problem with 6 classes.

© University of Pretoria

3.6.1.3 Tie-breaking

In the case of the vote count between classes being equal, or the distance between the input and training feature vectors of opposing classes are equal, there are several tie-breaking algorithms that can be implemented. In the case of vectors that are equally distant from each other, one such solution is to determine the dot product of the three vectors (one input vector and the two training vectors). Once the dot product has been performed, the winning vote is awarded to the dot product with the smallest scalar output. Another solution to the equal distance problem is simply a coin-flip or random assignment. Since they are equally likely, there is a 50% chance the vote will be incorrect. In the case of equal amounts of votes, a simple tie-breaker would be to dynamically change the value of k until there is a clear winner, or similarly to the previous case of equal distance, a coin-flip or random assignment can be made. However, intuitively, by adjusting k it is assumed that the outcome would have higher accuracy than a coin-flip.

3.6.2 Naive bayes

Naive Bayes (NB) is a probabilistic classifier which applies Bayes' theorem to a set of features to determine the probability of them belonging to a class. This assumes that each feature is independent of others. Equation 3.21 shows Bayes' theorem of conditional probability, where $\boldsymbol{X}$ is the input feature vector and C_k is an instance of a class. All calculations are performed for all k classes and the predicted class is chosen by maximum likelihood, which is the maximum of $P(C_k \mid \boldsymbol{X})$.

$$P(C_k \mid \boldsymbol{X}) = \frac{P(\boldsymbol{X} \mid C_k)\, P(C_k)}{P(\boldsymbol{X})} \tag{3.21}$$

Given a set of training data, prior probabilities are able to be calculated. Once all posterior probabilities for each class have been calculated, the maximum probability over all classes is chosen and a classification is made. In the unlikely event that the posterior probabilities are equal, the classifier must employ a tie-breaking algorithm to choose a class. Typically if these probabilities are equal, the class can be chosen randomly between the tied classes. Another approach is by choosing the class with the highest prior probability. A major advantage of using the NB classifier is that it requires a small amount of training data to accurately estimate a class [26]. Figure 3.11 shows an example Bivariate

© University of Pretoria

Gaussian probability of an input feature vector belonging to a certain class.

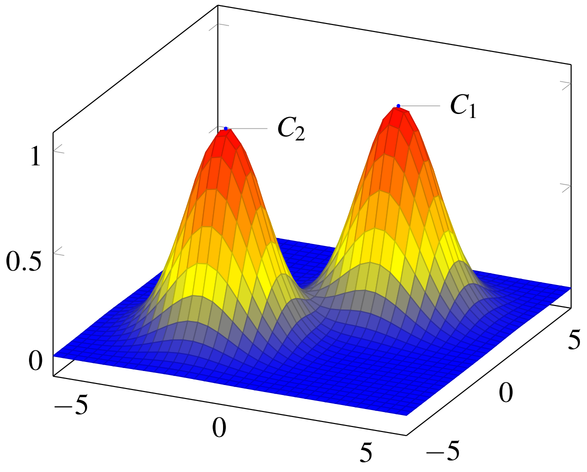


Figure 3.11. Bivariate Gaussian probability distribution of two different example classes.

Various models of the classifier using different probability distributions have been created. These models are used based on various criteria of the input data. The distributions of features are called the *event model* of the classifier.

3.6.2.1 Gaussian naive bayes

When the data is continuous, the assumption is that the values for each class are distributed according to a Gaussian distribution [37]. The likelihoods are calculated using the expression as shown in (3.22), where C_k is the class, σ_{C_k} is the Standard Deviation and μ_{C_k} is the Mean of the feature distribution and x_i is the input feature.

$$P(\boldsymbol{X} \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_{C_k}{}^2}} \exp\left(-\frac{(x_i - \mu_{C_k})^2}{2\sigma_{C_k}{}^2}\right) \tag{3.22}$$

3.6.2.2 Multinomial naive bayes

The Multinomial model is suitable for classification of discrete features, which are typically integers. The likelihood of observing X given class C_k is shown in (3.23). This equation makes use of (3.24) and (3.25), where $N_{C_k i}$ is the amount of times a feature i appears in samples of class C_k in the training

© University of Pretoria

set T; N_{C_k} is the total count of all features for the class C_k; n is the amount of features in the feature space and α is a smoothing factor. An $\alpha \geq 0$ accounts for features not present in the training samples and prevents non-zero probabilities in calculations.

$$P(\boldsymbol{X} \mid C_k) = \frac{N_{C_k i} + \alpha}{N_y + \alpha n} \tag{3.23}$$

$$N_{C_k i} = \sum_{x \in T} x_i \tag{3.24}$$

$$N_{C_k} = \sum_{i=1}^{n} N_{C_k i} \tag{3.25}$$

3.6.2.3 Bernoulli naive bayes

The Bernoulli model is suitable for classification with features that are independent boolean inputs. The likelihood of observing X given class C_k is shown in (3.26), where p_{ki} is the probability of the class C_k generating the term x_i.

$$P(\boldsymbol{X} \mid C_k) = \prod_{i=1}^{N} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)} \tag{3.26}$$

3.6.3 Artificial neural network

An artificial neural network is loosely modelled on how a brain solves problems. The network is a formation of neurons and the interconnections between them called synapses. Neurons act as a summation point for data travelling through the synapses. When data moves through the neural network in only one direction to an output neuron, this is known as a feed-forward neural network.

A single layer perceptron is a neural network which consists of only one layer of output nodes. A multi-layer perceptron consists of at least three layers, a single input layer, a single output layer and one or more hidden layers. Every neuron in each layer, is connected through synapses, to every neuron in adjacent layers, with each synapse having its own corresponding weight. An example of a multi-layer perceptron is shown in Figure 3.12.

© University of Pretoria

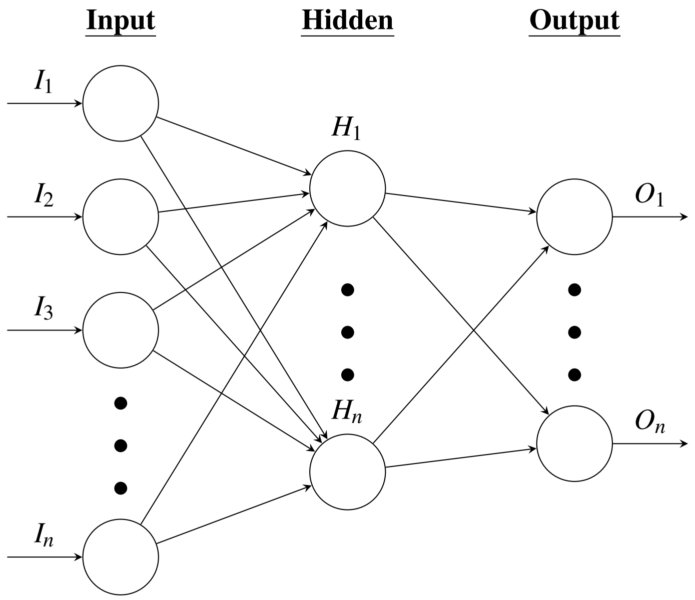


Figure 3.12. Structure of a fully connected multi-layer perceptron.

The output of a neuron is determined by an activation function and the weighted sum of inputs from its synapses. A step function, used as an activation function would act only as a binary classifier, however by using a non-linear "S" shaped function, the output becomes continuous and therefore has infinite possible outputs. A multi-layer perceptron makes use of these non-linear activation functions and is therefore used in deep learning [38].

3.6.3.1 Weight initialisation

Once the neural networks structure has been decided, the weights of each synapse are assigned during an initialisation process. Commonly used approaches for weight initialisation are discussed below:

- Zero Initialisation - This method sets all weights in the network to 0. However, it means that all neurons in each layer are performing the exact same calculation.
- Xavier Initialisation - This method sets all the weights in the network randomly according to a Gaussian distribution. The weights are generated with a zero-mean and the same variance for every layer [39].

© University of Pretoria

- He-et-al Initialisation - This method randomises the weights, however the weights are randomised based on the size of the previous layer of neurons. It is therefore a more controlled initialisation, and according to the study, leads to more efficient Gradient Descent [40].

3.6.3.2 Activation functions

The activation function of a node defines the output of that node given a set of inputs. There are several desirable properties of an activation function, which include it being nonlinear, smooth, continuously differentiable and has a defined range [41]. Listed below are several examples of activation functions commonly used.

- Rectifier - This activation function, also known is a ramp function was first introduced in a study of digital circuits [42]. It rose in popularity as the activation function of choice in deep learning, with other applications in computer vision and speech recognition [43]. It is represented in (3.27). Several problems exist using this activation function, with the largest being that it is not differentiable at 0, it is not centred around 0 and it is has a range of [0,∞).

$$f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \tag{3.27}$$

- Logistic Sigmoid - This function was first named and used in research that studied population growth [44]. It introduced non-linearity in neural networks and is capable of limiting the outputs within a specified range. This function is itself a derivative of the Softplus activation function, which is a logarithmic, smoothed representation of the Rectifier function. The Sigmoid function is represented in (3.28). In its standard form, this function is continuously differentiable, is nonlinear and has a range of (0,1).

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{3.28}$$

- Hyperbolic Tangent - This activation function has its roots in trigonometry. While visually similar to the ArcTan function, Tanh has properties that are different to ArcTan, but ideal for use an activation function [45]. It is continuously differentiable and has a range of (-1,1) and is

© University of Pretoria

represented in (3.29).

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.29}$$

- Identity - This activation function is essentially the most basic function there is and is represented in (3.30). It is similar to the Rectifier, however, it is continuously differentiable, is centred around 0, is linear and has a range of $(-\infty,\infty)$. This activation function is typically only used in output neurons [46].

$$f(x) = x \tag{3.30}$$

- Softmax - This activation function is a generalised form of the logistic function, it squashes a vector so that all terms in that vector are in a range of (0,1) and their sum is equal to 1. This function is exclusively used in output neurons and is commonly used for Multinomial Logistic Regression and is represented in (3.31) [46].

$$f(\mathbf{x}) = \frac{\exp(x_i)}{\sum\limits_{i=1}^{N} \exp(x_i)} \tag{3.31}$$

3.6.3.3 Loss functions

To generate an error between the predicted value of the neural network, compared to the actual value, a loss function must be defined. This loss function is ultimately used to determine how to update the weights of each synapse. Several mathematical expressions that are commonly used as loss functions are described below. In all functions listed, n is the number of outputs, y is the actual value, and $\hat{y}$ is the predicted value.

- Mean Squared Error (MSE) - This loss function is commonly used in linear regression; A method of minimising this error is called the Ordinary Least Squares (OLS) [47]. This loss function is represented in (3.32).

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.32}$$

© University of Pretoria

- Mean Squared Logarithmic Error (MLSE) - This loss function is a variant of MSE, it differs by taking the difference of log of the predicted and actual values. This aims to minimise the effect of a large difference between the values. This loss function has a tendency to underestimate the error. If both the predicted and actual values are small, then the MSE and MLSE will be almost identical. If either the predicted or actual value is large then the MSE will be much larger than MLSE. The loss function is represented in (3.33).

$$Loss = \frac{1}{n}\sum_{i=1}^{n}(log(y_i+1)-log(\hat{y}_i+1))^2 \tag{3.33}$$

- Mean Absolute Error (MAE) - This loss function is used to measure the average absolute distance between the predicted and actual values. Since this loss function negates the square, compared to MSE, MAE is more robust against outliers, meaning it does not create large errors. The loss function is represented in (3.34).

$$Loss = \frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{3.34}$$

- L1 - This loss function stands for Least Absolute Deviations (LAD), which is similar to MAE, but does not determine the average error. The loss function is shown in (3.35).

$$Loss = \sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{3.35}$$

- L2 - This loss function stands for Least Square Errors (LSE), is similar to MSE, but does not determine the average error. This function is represented in (3.36).

$$Loss = \sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{3.36}$$

3.6.3.4 Learning

Backpropagation is a means of updating the weights of the synapses between neurons in all layers of the neural network. Backpropagation is combined with what is called Gradient Descent, which makes use of the overall error gradient, to minimise the loss of the system. Equations 3.37 to 3.40 show how the weights from various layers are updated. In (3.37), $g'(in_k)$ is the derivative of activation function

© University of Pretoria

at in_k, where in_k is the weighted sum of inputs. Err_k is the error of the k'th component of the error vector, where the error vector is determined by the equations discussed in Section 3.6.3.3. In (3.38), (3.39) and (3.40), $w_{j,k}$ and $w_{i,j}$ are the weights of the different layers, where $w_{j,k}$ refers to the output layer and $w_{i,j}$ refers to the hidden and input layer weights; a_j and a_i refer to the output of the previous layers neurons, and α is the learning rate of the system.

$$\Delta_k = Err_k \times g'(in_k) \tag{3.37}$$

$$w_{j,k} \leftarrow w_{j,k} + \alpha \times a_j \times \Delta_k \tag{3.38}$$

$$\Delta_j = g'(in_j) \sum_k w_{j,k} \Delta_k \tag{3.39}$$

$$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta_j \tag{3.40}$$

3.6.3.5 Stopping criteria

An important consideration for neural networks is when to stop training the network. This is important because, if the network trains for too many iterations with the same configuration, overfitting may occur. Overfitting is where the neural network completely fits the training data, but generates errors when feeding the network with new data. As discussed previously one method of countering overfitting is by splitting the data up into training and validation sets. Another means of countering overfitting is by monitoring the error and stopping training once the error stops decreasing. One other method exists where the neural network is configured to only train over a pre-defined amount of Epochs. An Epoch is defined as the network having completed a full training cycle on training data.

3.6.4 Support vector machine

The aim of a Support Vector Machine (SVM) is to define the widest straight-line or linear separation between positive and negative samples of a target class. By definition a SVM is therefore a binary linear classifier. Since the SVM attempts to find a linear separation between the classes, a decision rule must be defined. These decision rules are defined in (3.41) to (3.43). Equation 3.41 represents the optimal hyperplane which separates the training data with a maximal margin. It determines the direction where the distance between training vectors of two different classes is at a maximum. To

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

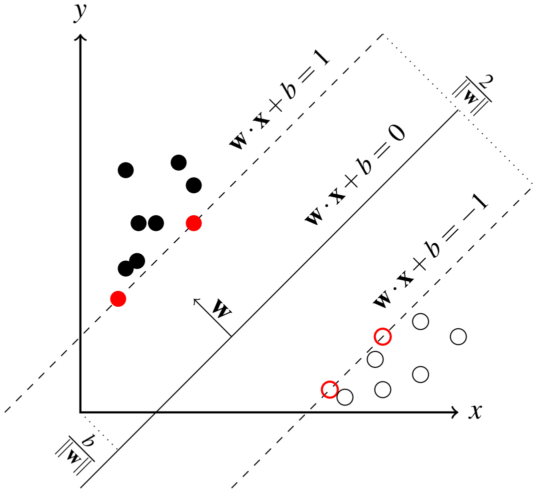

y
x
$\mathbf{w}\cdot\mathbf{x}+b=1$
$\mathbf{w}\cdot\mathbf{x}+b=0$
$\mathbf{w}\cdot\mathbf{x}+b=-1$
$\boldsymbol{w}$
$\frac{2}{\|\mathbf{w}\|}$
$\frac{b}{\|\mathbf{w}\|}$

Figure 3.13. Separation of positive and negative samples by maximum margin hyperplane.

further constrain the problem, (3.42) and (3.43) are satisfied by the vectors $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ which are termed support vectors and are positive and negative samples of the target class [48].

$$\boldsymbol{w}\cdot\boldsymbol{x}+b=0 \tag{3.41}$$

$$\boldsymbol{w}\cdot\boldsymbol{x}_+ +b=1 \tag{3.42}$$

$$\boldsymbol{w}\cdot\boldsymbol{x}_- +b=-1 \tag{3.43}$$

A visual representation of the maximum margin hyperplanes are shown in Figure 3.13. In the figure, the black circles represent positive samples while the white circles represent negative samples. The red circles indicate support vectors, or samples which meet the criteria in (3.42) and (3.43). The parameter $\frac{2}{||\boldsymbol{w}||}$, is the distance between the two hyperplanes and is called the Margin. The parameter $\frac{b}{||\boldsymbol{w}||}$ is the offset of the optimal hyperplane from the origin, along the normal vector $\boldsymbol{w}$.

In order to find the vector $\boldsymbol{w}$, the system is required to solve the optimisation problem shown in (3.44); The equation shows that the optimal hyperplane is a linear combination of training vectors. The term y_i is +1 for positive samples and -1 for negative samples where x_i is the training sample vector; The parameter α is > 0 for support vectors only.

$$\boldsymbol{w}=\sum \alpha_i y_i \boldsymbol{x_i} \tag{3.44}$$

© University of Pretoria

After determining the vector $\boldsymbol{w}$, a final classification is made by using the (3.45). A result that is ≥ 1 yields a positive classification whereas a result that is $\leq$ -1 yields a negative classification.

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{3.45}$$

3.6.4.1 Kernels

The largest problem with a SVM is that if the training samples are not linearly separable then the optimisation problem will never converge. To combat this, the use of a kernel function transforms the feature space into a separating space, where $\boldsymbol{x} \rightarrow \phi(\boldsymbol{x})$. The properties of the classifier with vector $\boldsymbol{w}$ means that (3.44) becomes (3.46).

$$\boldsymbol{w} = \sum \alpha_i y_i \phi(\boldsymbol{x_i}) \tag{3.46}$$

The kernel function K must satisfy (3.47):

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \phi(\boldsymbol{x_i}) \cdot \phi(\boldsymbol{x_j}) \tag{3.47}$$

The linearity of the dot product implies the classification of the input vector changes (3.45) into (3.48).

$$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}) + b = \sum y_i \alpha_i \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x_i}) + b \tag{3.48}$$

Several examples of kernel functions are presented in (3.49) to (3.52). Equation 3.49 is an example of a homogeneous polynomial, (3.50) is an example of an in-homogeneous polynomial, (3.51) is the Gaussian Radial Basis Function (RBF) and (3.52) uses a Hyperbolic tangent. Figure 3.14 shows an example of applying a kernel to a non-linearly separable feature space, transforming it into a linearly separable space. Each kernel function has its own advantages however, the RBF shown in (3.51) has been shown to converge even when thousands of features are in the feature space [49].

© University of Pretoria

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = \boldsymbol{x_i} \cdot \boldsymbol{x_j} \tag{3.49}$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = (\gamma(\boldsymbol{x_i} \cdot \boldsymbol{x_j}) + r)^d \tag{3.50}$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = \exp\left(-\gamma|\boldsymbol{x_i} - \boldsymbol{x_j}|^2\right) \tag{3.51}$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = \tanh(\gamma(\boldsymbol{x_i} \cdot \boldsymbol{x_j}) + r) \tag{3.52}$$

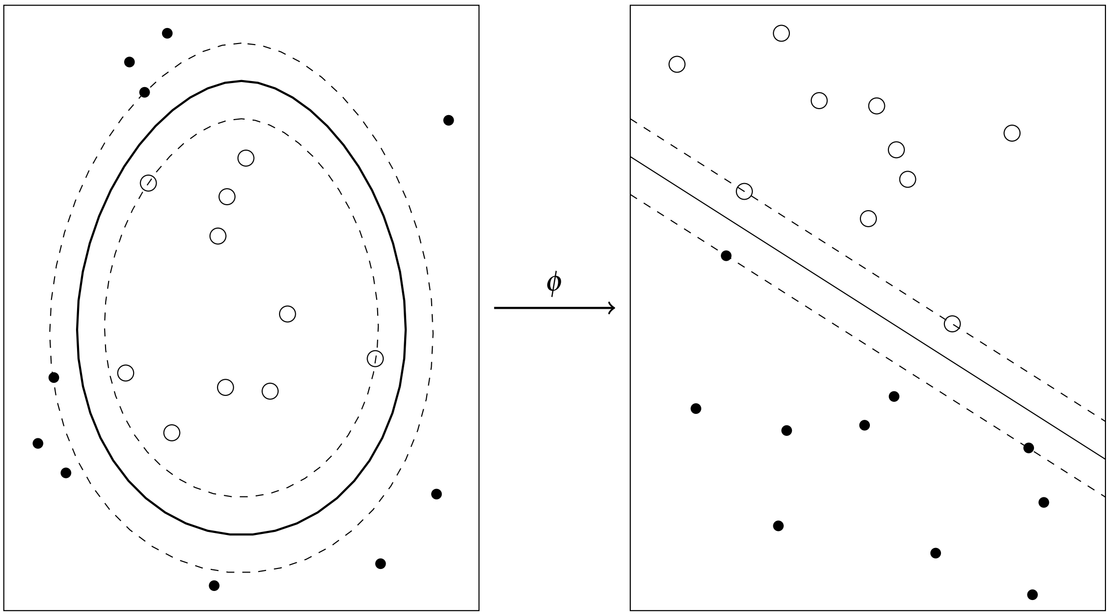


Figure 3.14. Transformation of the separation between samples of a non-linear feature space to linear kernel space.

3.6.4.2 Multiclass classification

Since the SVM is naturally a binary classifier, it is limited to classifying a maximum of 2 classes. The common approaches to convert the SVM into a multiclass classifier are, One-vs-Rest and One-vs-One. These strategies reduce a multiclass problem, into multiple binary problems and may also be called transformation techniques.

- One-vs-Rest (OvR) - This strategy involves training a classifier for each class, by using positive samples of the target class and all other classes as negative samples of the target class. A

© University of Pretoria

requirement for using this strategy is that the classifier does not produce only a label but some score that gives confidence in its decision. This strategy takes an input sample, classifies it using all the created classifiers and then assigns the class based on the maximum confidence score amongst all classifier outputs.

- One-vs-One (OvO) - This strategy creates $N \times (N-1)/2$ binary classifiers, where each classifier trains with samples of pairs of classes. Unlike OvR this strategy does not require a confidence score for each classifier, because it is capable of operating with discrete labels assigned to samples. Given a random input sample, all $N \times (N-1)/2$ classifiers classify the sample. Similar to the k-NN classifier, a voting scheme is used whereby a tally of the assigned class from each classifier is taken and the class with the highest amount of votes is ultimately assigned as the predicted class.

3.7 GENERATING RESULTS

In this section the configurations of the various stages in the machine learning models are presented. Finally a discussion on the approach of the overall experiment is included.

3.7.1 Configurations

Before being able to generate results the configurations of the various classifiers and feature selection algorithms need to be set. The feature selection algorithms require little to no configuration with mRMR needing only a discretisation threshold to be configured. As discussed previously, this threshold was set to 0.25. The amount of features chosen from the mRMR ranker, will be equal to the amount of features generated in the CFS feature subset. The CFS algorithm is configured to use the forward selection search strategy and CFS-MDL for measuring feature quality. The CFS generated dataset will be labeled CFS, the mRMR datasets generated by the Maximum Relevance ranker will be labelled MaxRel, mRMR-MID for the Mutual Information Difference ranker and mRMR-MIQ for the Mutual Information Quotient ranker.

Each classifier also requires various configurations as discussed in their respective sections above. The 1-NN, 3-NN and 5-NN classifiers use 1, 3 and 5 nearest neighbours respectively. Each of these k-NN classifiers were configured to use the Euclidean Distance metric between input and training

© University of Pretoria

vectors. Tie breaking was done by choosing randomly between tied classes. The Naive Bayes classifier was configured to be a Gaussian Naive Bayes classifier, also using a randomly chosen class for tie-breaking.

The Multi-Layer Perceptrons were configured to have 1 Hidden Layer containing 100 neurons and 1 Output Layer with an output neuron for each activity or location, while using the Xavier Initialisation method for weight assignment. Two activation functions will be compared, namely, the Logistic Sigmoid and Hyperbolic Tangent. The neural network using Logistic Sigmoid will be labelled as MLP-S and the neural network using the Hyperbolic Tangent will be labelled MLP-T. Both of these classifiers use the Mean Squared Error as their loss function and use a learning rate of 0.0001. Both MLPs also use a stopping criteria where the error stops decreasing. The SVM is configured to use a Radial Basis Function as its kernel and uses a One-vs-Rest approach for multiclass classification; Data generated by the SVM will be labelled as SVM.

3.7.2 Approach

To generate a robust set of results, it was decided that a brute-force or exhaustive search would be used. This means that for every input dataset, the data is evaluated using every possible combination of feature selection algorithm and classifier. To give an example, a diagram showing the interconnections between the various stages is shown in Figure 3.15. In the Figure the structure resembles that of the fully connected Neural Network shown in Figure 3.12. However, contrary to the Neural Network, every path from beginning to end is independent and the results from one path have no effect on the results of another.

3.8 CHAPTER SUMMARY

In this chapter, the structure of four machine learning models that solve the activity recognition problem are proposed. Each of the four models are designed to accommodate the on-body device location differently. The various facets of the implementations for each stage in the machine learning models are discussed. These stages include data collection, pre-processing, feature selection and classification. Furthermore, the configurations for each stage are set and the approach to generating comparable results among models is finalised.

© University of Pretoria

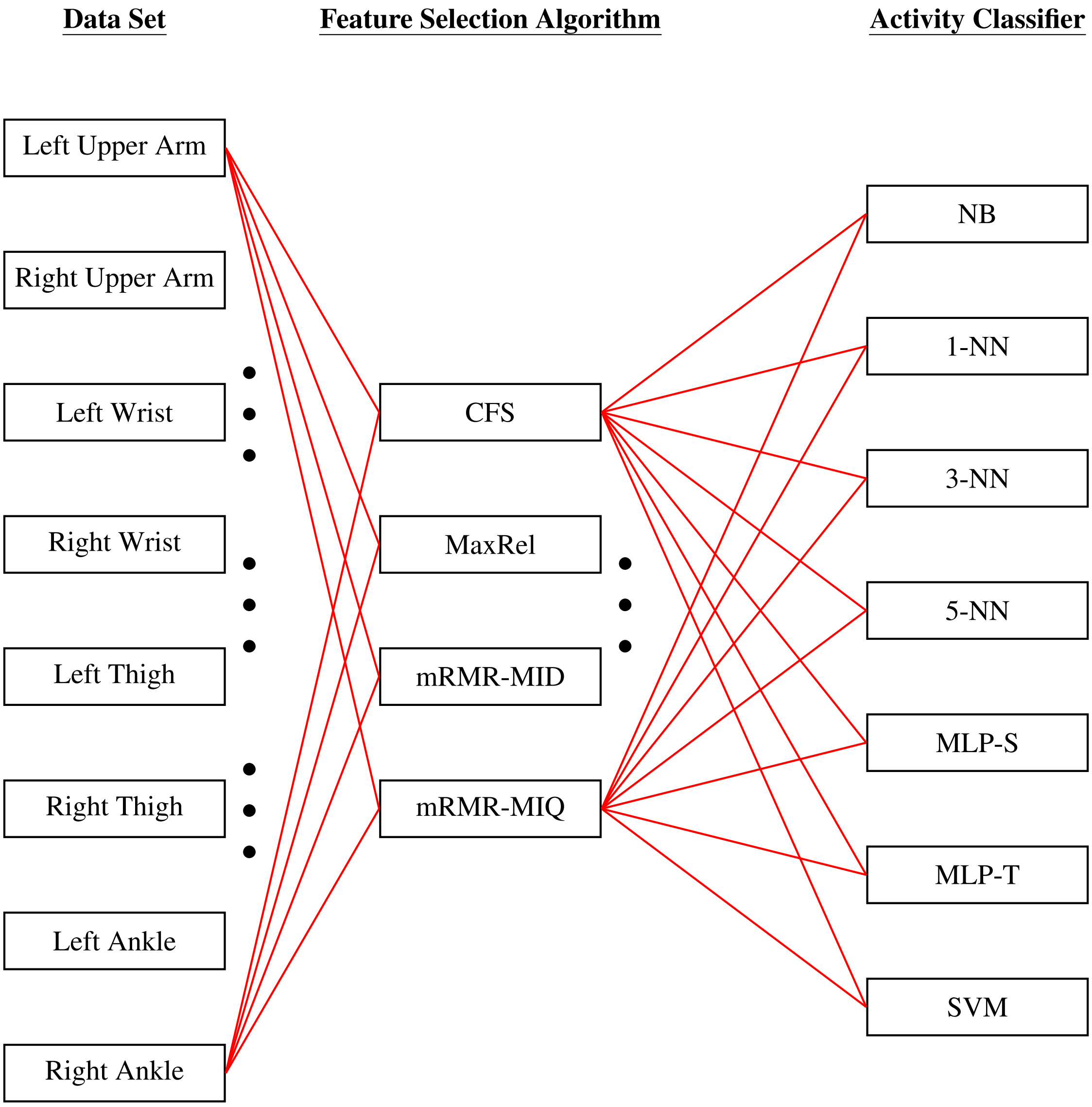


Figure 3.15. Diagram for Model 3 results collection.

© University of Pretoria

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

In this chapter the results of all machine learning models are presented, compared and analysed. The results for Model 1 and 2 are presented in Section 4.2 and 4.3 respectively. Furthermore, the results of each individual location, which make up Model 3, are presented in Section 4.4. A final model for Model 4 is then generated and its results are evaluated in Section 4.5. Finally the results for each model are summarised, compared and further analysed in Section 4.6.

4.2 MODEL 1

The following results were obtained for Model 1. The entire dataset is stored in a single file and each feature selection algorithm and classifier must generalise all activities regardless of the device location. The results for the classifier accuracies are shown in Figure 4.1.

From Figure 4.1 it is clear that the overall best performing datasets are the ones generated by CFS and mRMR-MIQ. The CFS dataset performs the best while using the 3 and 5 NN classifiers, while the mRMR-MIQ dataset performs better when using the NB, 1-NN, MLP-S, MLP-T and SVM classifiers. The maximum achieved accuracy across all classifiers was the 1-NN classifier using the mMRM-MIQ dataset with it achieving 89.0% accuracy. This was closely followed by the CFS dataset also using 1-NN with an 88.9% accuracy. The worst performing classifier and dataset combination was the MLP-S using the mRMR-MID dataset, which achieved a classification accuracy of 41.7%. For the 1-NN classifier and mMRM-MIQ dataset, the weighted average recall was 86.445%, the weighted average precision is 86.860% and the classification accuracy has a standard deviation of 0.47%.

© University of Pretoria

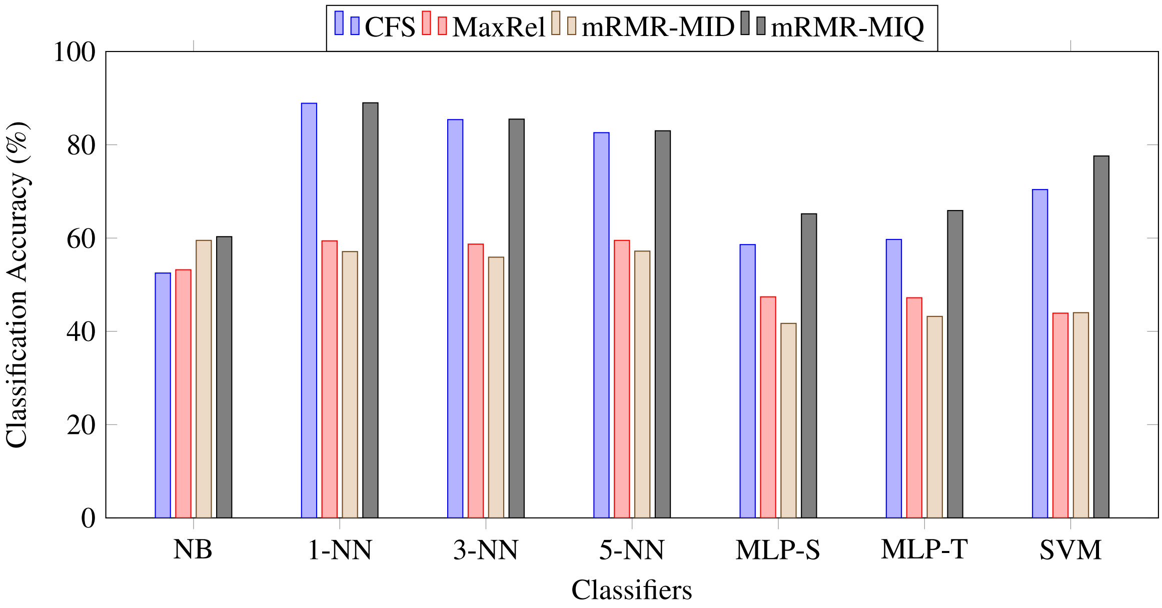


Figure 4.1. Classification accuracies for all feature subset and classifier combinations of Model 1.

By considering the confusion matrix in Table A.1 the right leg lunge, left leg lunge, squat and hinge were misclassified as each other as often as 6.1% of the time. A similar result is observed between walking, jogging and running with misclassifications occurring in 8.5% of the observations. Standing and lying down were accurately represented with a misclassification happening in only 0.5% of the observations.

Table 4.1. Model 1 Features from the mMRMR-MIQ dataset.

Sensor	Axis	Feature
Accelerometer	Y	Mean
Accelerometer	Y	Median
Accelerometer	Z	Mean
Magnetometer	Y	RMS
Magnetometer	Z	RMS
Gyroscope	Y	Mean
Gyroscope	Z	Fundamental Frequency

By considering the features in Table 4.1, a total of 7 features were chosen with the accelerometer appearing in 42.9%, the magnetometer in 28.6% and the gyroscope appearing in 28.6% of all features.

© University of Pretoria

The axis which was represented the most was the Y axis, appearing in 57.1% of features with the Z axis appearing in the remaining 42.9%.

4.3 MODEL 2

The following results were obtained for Model 2. This model is identical to that of Model 1, with the addition of the The known location of the device being appended to the dataset as a feature. The results for the classifier accuracies are shown in Figure 4.2.

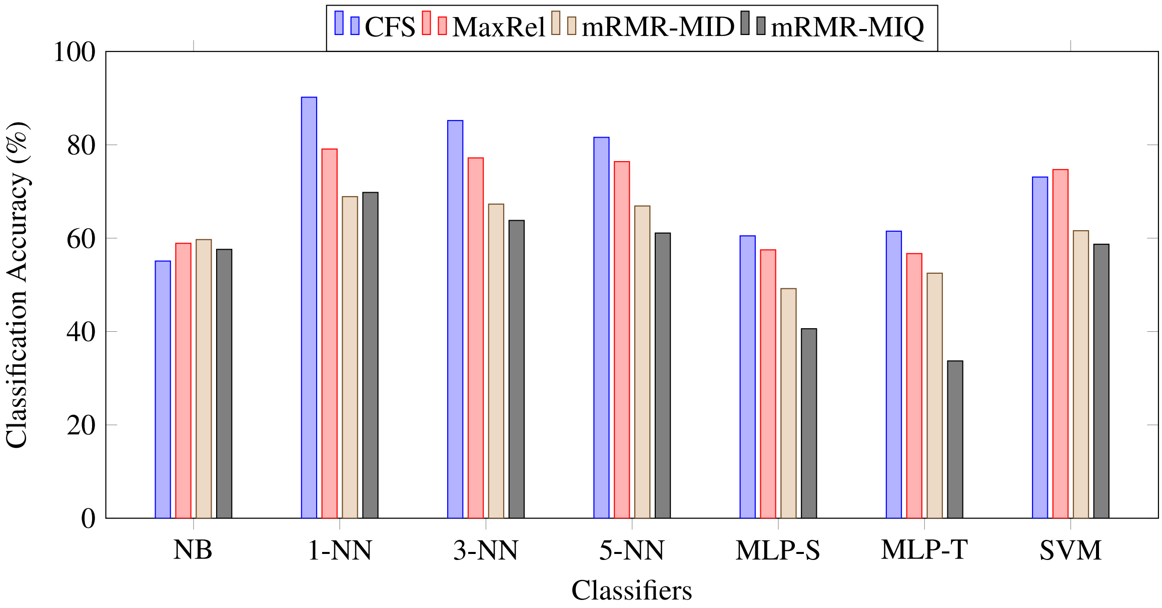


Figure 4.2. Classification accuracies for all feature subset and classifier combinations of Model 2.

As shown in Figure 4.2 the overall best performing dataset was the CFS dataset, which was the top performer using 5 of the 7 classifiers. The CFS dataset performed the best using the 1-NN, 3-NN, 5-NN, MLP-S and MLP-T classifiers, with the mRMR-MID dataset performing the best using the NB classifier and the MaxRel dataset performing the best using the SVM. The best overall performing dataset and classifier combination was the CFS dataset with the 1-NN classifier. It achieved a classification accuracy of 90.2%. The worst performing dataset and classifier combination was the MLP-T with the mRMR-MIQ dataset, reaching only a 40.6% classification accuracy. The CFS and 1-NN combination had a weighted average recall of 88.185%, a weighted average precision of 88.627% and has a standard deviation of 0.37%.

© University of Pretoria

By considering the confusion matrix in Table A.2, it is clear that all activities except for lying down were misclassified as eachother, with running being the most misclassified as jogging in 8.6% of the observations. Apart from the misclassifications between running, jogging and walking, no other misclassification occured more than 4.5% of the time. Lying down was misclassified as a left leg lunge in 0.1% of the observations.

Table 4.2. Model 2 Features from the CFS dataset.

Sensor	Axis	Feature
Accelerometer	Y	Mean
Accelerometer	Y	Median
Accelerometer	Z	Mean
Magnetometer	Y	RMS
Magnetometer	Z	RMS
Gyroscope	Y	Mean
Gyroscope	Z	Fundamental Frequency
N/A	N/A	Location

Considering the features listed in Table 4.2, the results are identical to those of Model 1 with the added exception of including the location of the device as a feature.

4.4 MODEL 3

The following results were obtained for Model 3, where the data for each on-body device location was stored separately and treated as an individual problem. Each feature selection algorithm and classifier has to generalise the data for each location individually. For this reason the results are split up by their location and an average performance for the entire model will be calculated after examining each location individually. Section 4.4.1 contains results for the upper arms, Section 4.4.2 contains results for the wrists, Section 4.4.3 contains results for the thighs and finally Section 4.4.4 contains results for the ankles.

© University of Pretoria

4.4.1 Upper arms

The results obtained from the various classifiers and datasets for the right and left upper arms are shown in Figures 4.3 and 4.4 respectively.

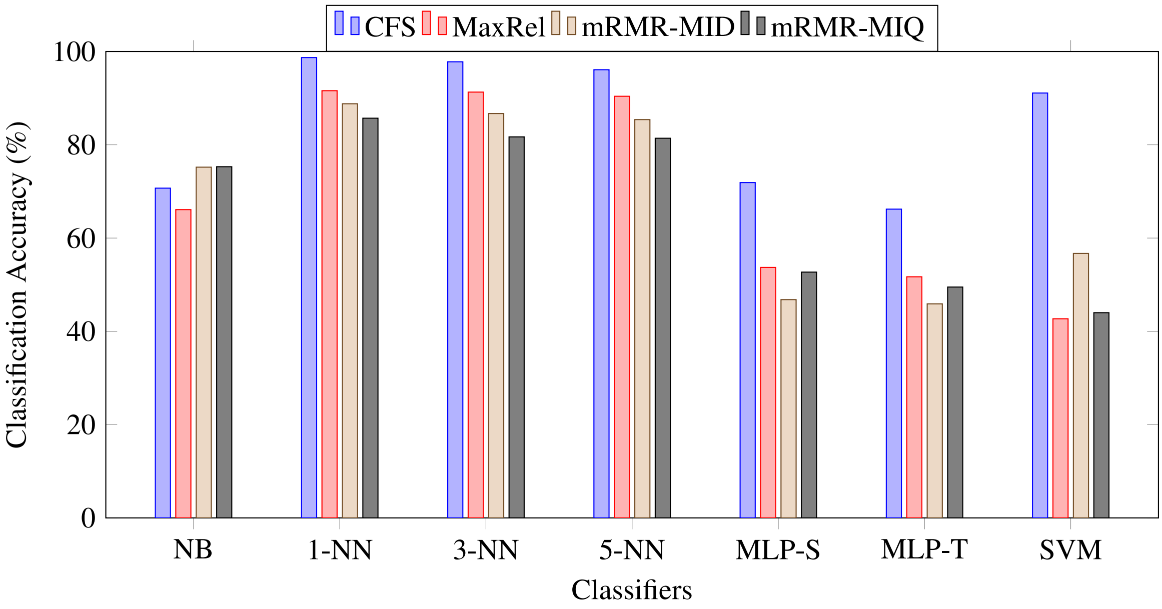


Figure 4.3. Classification accuracies for all feature subset and classifier combinations for the right upper arm of Model 3.

With reference to Figure 4.3 which presents the classification accuracy for the right upper arm, the overall best performing dataset is CFS using 6 of the 7 classifiers. The CFS dataset performed the best using the 1-NN, 3-NN, 5-NN, MLP-S, MLP-T and SVM classifiers, with the mRMR-MIQ dataset being the top performer using the NB classifier. The worst performing classifier and data set combination was the MaxRel dataset using the SVM, achieving only 42.7% accuracy. Overall the CFS dataset with the 1-NN classifier obtained the best result of 98.7% accuracy. It achieved a weighted average recall of 98.334% and a weighted average precision of 98.473%.

By considering the confusion matrix in Table A.3, only the right leg lunge, squat, jog and run were classified with 93.6%, 95.1%, 98.8% and 97.5% accuracy respectively while the remaining activities were all classified with 100% accuracy. The right leg lunge was misclassified as a left leg lunge or squat, a squat was misclassified as a left leg lunge, a hinge or a jog; A jog was misclassified as a run and a run was misclassified as a jog.

© University of Pretoria

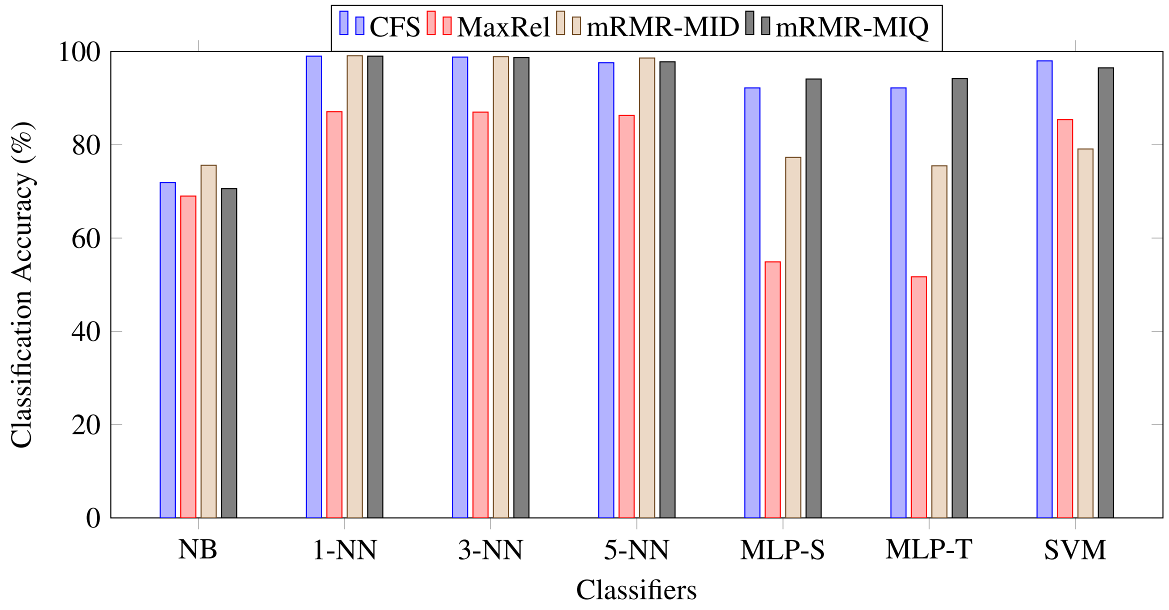


Figure 4.4. Classification accuracies for all feature subset and classifier combinations for the left upper arm of Model 3.

The results for the classification accuracies of the left upper arm are shown in Figure 4.4. The results show a highly contested top performance between the CFS, mRMR-MID and mRMR-MIQ datasets when using various classifiers. Overall the top performing dataset was the mRMR-MID dataset using the 1-NN, 3-NN and 5-NN classifiers. The mRMR-MIQ dataset was the top performer using the MLP-S and MLP-T classifiers and the CFS dataset was the top performer using the SVM. For the 1-NN classifier, CFS obtained 99.0% accuracy, mRMR-MID obtained 99.1% accuracy and mRMR-MIQ obtained 99.0% accuracy. Using the 3-NN classifier the CFS, mRMR-MID and mRMR-MIQ datasets obtained 98.8%, 98.9% and 98.7% classification accuracies respectively. The worst performing classifier and dataset combination was the MLP-T using the MaxRel dataset, obtaining a 51.7% classification accuracy. This of course means the top performing classifier and dataset combination was the 1-NN classifier with the mRMR-MID dataset. This combination managed to obtain a 98.476% weighted average recall and 98.668% weighted average precision.

By considering the confusion matrix in Table A.4, the right leg lunge, left leg lunge, hinge and run obtained a 93.6%, 97.1%, 96.7% and 98.9% classification accuracy respectively. The remaining activities obtained a 100% classification accuracy. Both lunges were in some cases misclassified as the lunge using the opposite leg or a hinge. In some cases a hinge was misclassified as a walk and running was in other cases misclassified as a walk.

© University of Pretoria

Table 4.3. Left and right upper arm features.

Left - mMRMR-MID			Right - CFS		
Sensor	Axis	Feature	Sensor	Axis	Feature
Accelerometer	X	Mean	Accelerometer	X	Mean
Accelerometer	Z	Median	Accelerometer	Z	Median
Accelerometer	X	Peak FFT Amplitude	Gyroscope	Z	Median
Gyroscope	X	Fund. Frequency	Gyroscope	X	Fund. Frequency
Gyroscope	Y	Peak FFT Amplitude	Gyroscope	Z	RMS
Gyroscope	Z	Fund. Frequency	Gyroscope	Z	Fund. Frequency
Magnetometer	Z	RMS			

Shares 61.5% of total features

Table 4.3 shows a combined list of features from their own respective best performing datasets, mRMR-MID for the left upper arm and CFS for the right upper arm. The dataset for the left upper arm contains 7 features, of which 3 are from the accelerometer (42.9%), 3 from from the gyroscope (42.9%) and the final feature being from the magnetometer (14.3%). The 7 features contained 3 along the X axis (42.9%), 1 along the Y axis (14.3%) and the remaining 3 along the Z axis (42.9%). When examining the right upper arm dataset, 6 features were present. In those features, 2 were taken from the accelerometer (33.3%) while the remaining 4 features were taken from the gyroscope (66.6%). Considering those 6 features, 2 features were along the X axis (33.3%) and the remaining 4 were along the Z axis (66.6%). When considering both datasets together there were a total of 13 combined features, of which 8 (61.5%) of them were shared.

4.4.2 Wrists

The results obtained from the various classifiers and datasets for the right and left wrists are shown in Figures 4.5 and 4.6 respectively.

As shown in Figure 4.5, the overall best performer was the MaxRel dataset which obtained the best results using 5 of the 7 classifiers. MaxRel obtained the best results using the NB, 3-NN, 5-NN, MLP-S and MLP-T classifiers, while the CFS dataset obtained the best result using the 1-NN classifier and

© University of Pretoria

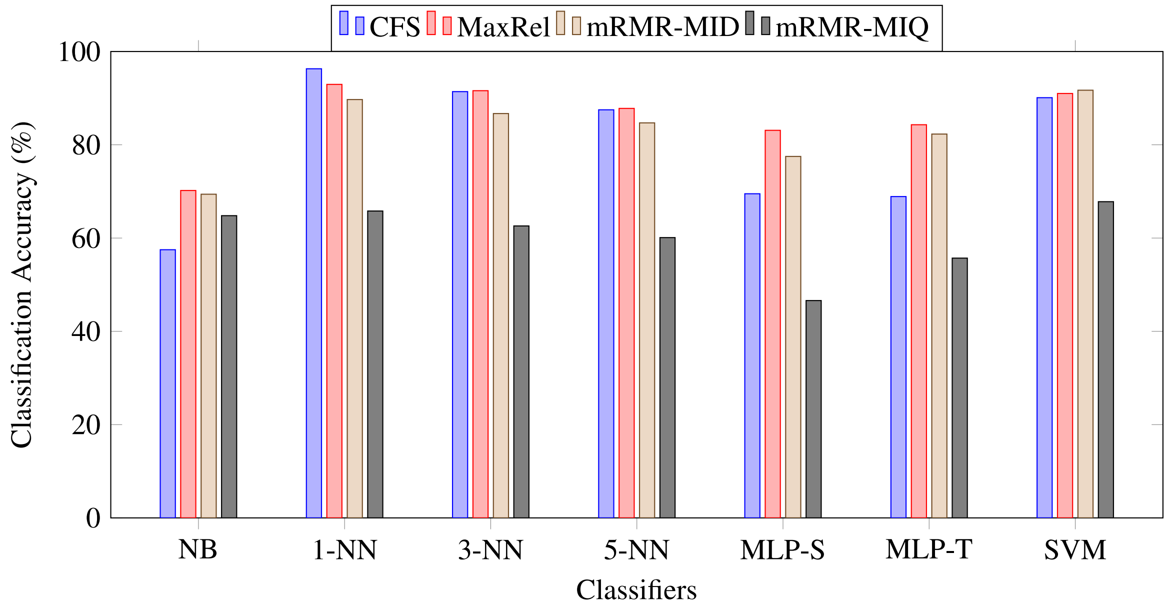


Figure 4.5. Classification accuracies for all feature subset and classifier combinations for the right wrist of Model 3.

mRMR-MID obtained the best result using the SVM. The worst performing classifier and dataset combination was the MLP-S using the mRMR-MIQ dataset which obtained a classification accuracy of 46.6%. Although MaxRel was the overall top performer across all classifiers, it did not achieve the highest classification accuracy. The CFS dataset using the 1-NN classifier managed to obtain a 96.3% classification accuracy with its next closest competitor being MaxRel using the 1-NN classifier with 92.95% classification accuracy. The CFS and 1-NN combination managed to obtain a weighted average recall of 96.092% and a weighted average precision of 96.345%.

By considering the confusion matrix in Table A.5, only lying down managed to achieve a 100% classification accuracy. The worst performing activity was a jog, which obtained 90.7% accuracy and was occasionally misclassified as a walk or a run.

With reference to Figure 4.6 the overall best performing dataset was the mRMR-MID dataset obtaining the best result using 4 of the 7 classifiers. The mRMR-MID dataset performed the best using the NB, 3-NN, 5-NN and MLP-S classifiers, with the CFS dataset being the top performer using the 1-NN and SVM classifiers and the MaxRel dataset performed the best using the MLP-T classifier. The worst performing classifier and dataset combination was the mRMR-MIQ dataset using the SVM classifier which obtained a classification accuracy of 48.6%. The combination with the highest average

© University of Pretoria

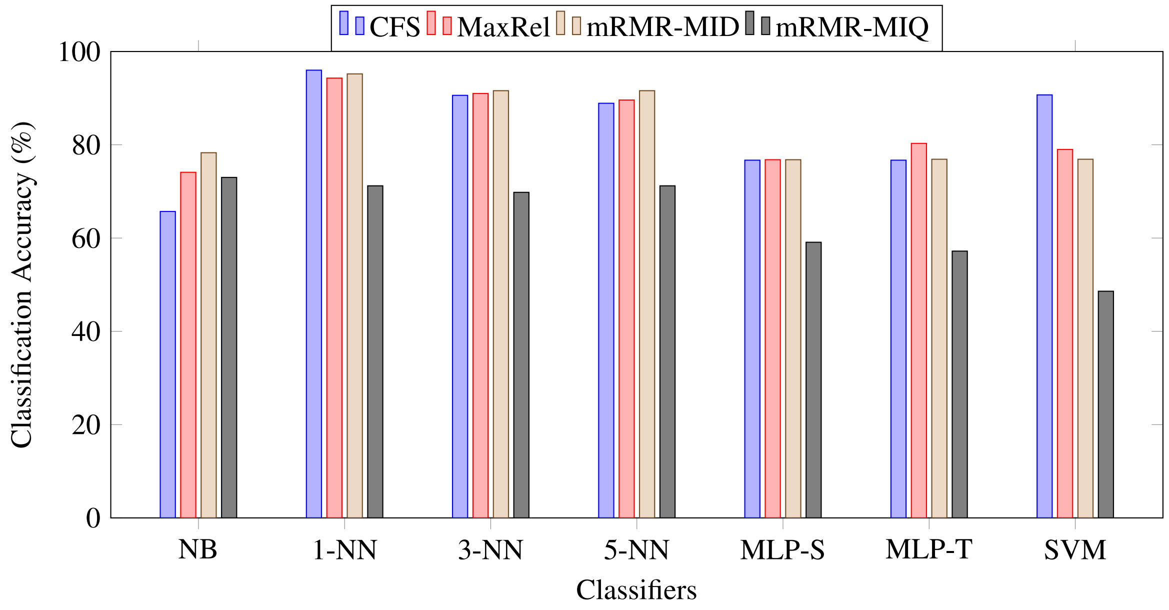


Figure 4.6. Classification accuracies for all feature subset and classifier combinations for the left wrist of Model 3.

accuracy was the CFS dataset with the 1-NN classifier obtaining a classification accuracy of 96.0%. The combination achieved a weighted average recall of 94.585% and a weighted average precision of 94.306%.

By looking at the confusion maxtrix in Table A.6 once again the only activity achieving a 100% classification accuracy was lying down. The worst performing activity was squatting which was misclassified as a left leg lunge, walk, jog or standing still. A similar result is seen for jogging, however this was misclassified as running in 12.1% of the observations.

Table 4.4 shows the features of the best performing datasets for the left and right wrists, where both datasets were generated by using the CFS feature selection algorithm. The left wrist dataset contained 9 features, 4 of which were from the accelerometer (44.44%), another 4 from the gyroscope (44.44%) and the last being from the magnetometer (11.11%). Of those 9 features, 3 of them were along the X axis (33.33%), 4 were along the Z axis (44.44%) and 2 were along the Y axis (22.22%). For the right wrist, the dataset contained 10 features. Of these features, 2 were from the accelerometer (20%), 6 were from the gyroscope (60%) and the remaining 2 were from the magnetometer (20%). The features comprised of 4 along the X axis (40%), 4 along the Y axis (40%) and the remaining 2 along the Z axis (20%). This results in a total of 19 features when considering both datasets, with 12 of the features

© University of Pretoria

Table 4.4. Left and right wrist features.

Left - CFS			**Right - CFS**		
Sensor	Axis	Feature	Sensor	Axis	Feature
Accelerometer	X	Fund. Frequency	Accelerometer	X	Fund. Frequency
Accelerometer	Z	Mean	Accelerometer	Y	Mean
Accelerometer	Z	RMS	Gyroscope	X	Variance
Accelerometer	Z	Fund. Frequency	Gyroscope	Y	Mean
Gyroscope	X	Fund. Frequency	Gyroscope	X	Fund. Frequency
Gyroscope	Y	Fund. Frequency	Gyroscope	Y	Fund. Frequency
Gyroscope	Z	Fund. Frequency	Gyroscope	Z	Fund. Frequency
Gyroscope	X	Mean	Gyroscope	X	Mean
Magnetometer	Y	Mean	Magnetometer	Y	Mean
			Magnetometer	Z	Fund. Frequency

Shares 63.2% of total features

being common between both wrists (63.2%).

4.4.3 Thighs

The results obtained from the various classifiers and datasets for the right and left thighs are shown in Figures 4.7 and 4.8 respectively.

Figure 4.7 shows the classification accuracies for the right thigh. The figure shows the overall top performing dataset is the CFS dataset, being the top performer using 6 of the 7 classifiers. It was the best performer using the 1-NN, 3-NN, 5-NN, MLP-S, MLP-T and SVM classifiers with the mRMR-MID dataset was the top performer using the NB classifier. The worst performing classifier and dataset combination was the mRMR-MIQ dataset using the MLP-S classifier, which achieved an average classification accuracy of 49.5%. The best performing combination was the CFS dataset using the 1-NN classifier, which obtained an average classification accuracy of 96.1%. This combination achieved a weighted average recall of 95.209% and a weighted average precision of 94.572%.

© University of Pretoria

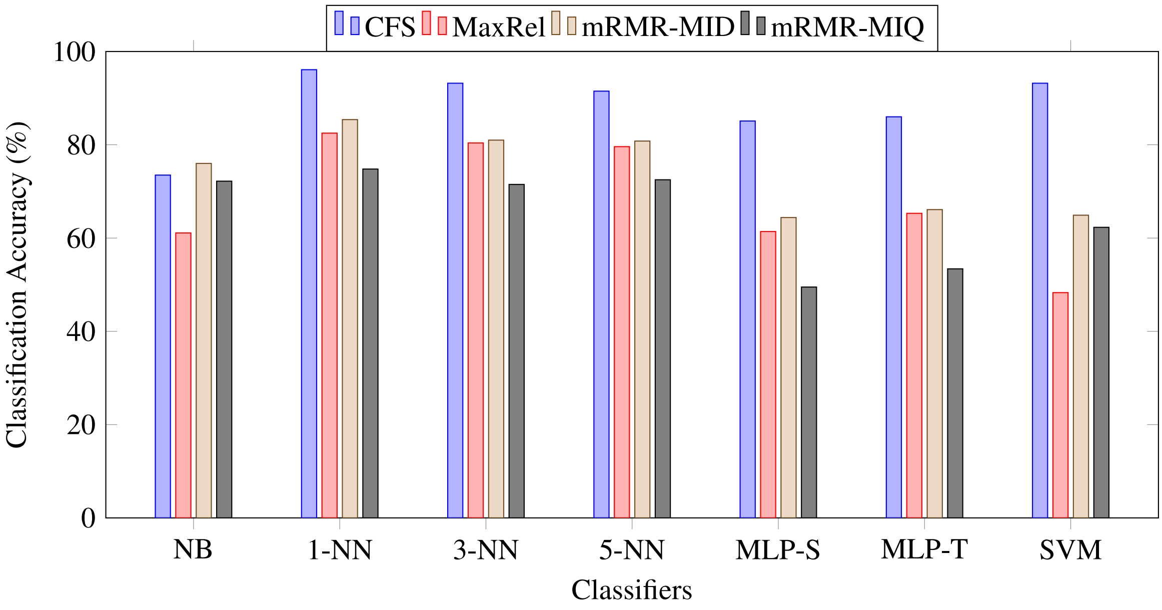


Figure 4.7. Classification accuracies for all feature subset and classifier combinations for the right thigh of Model 3.

By looking at the confusion matrix shown in Table A.7, it is clear to see that the left leg lunge and lying down obtained a classification accuracy of 100%. The worst performing activity was a hinge which was misclassified as a right leg lunge, a jog or standing still in 15.6% of the observations. The next worst performing activity was a right leg lunge which was misclassified as a squat, a hinge or a jog in 13.3% of the observations.

Figure 4.8 shows the classification accuracies for the left thigh. The figure shows the complete dominance of the MaxRel dataset which was the top performing dataset using all 7 classifier configurations. The mRMR-MIQ and CFS datasets alternated as the second best performer each being second in 3 of the 7 configurations. The worst performing dataset and classifier combination was the mRMR-MID dataset using the MLP-S classifier, with it obtaining a classification accuracy of 37%. The combination with the highest achieving classification accuracy was the 1-NN classifier with the MaxRel dataset, achieving a classification accuracy of 96.4%. This combination achieved a weighted average recall of 95.871% and a weighted average precision of 95.938%.

Table A.8 shows a confusion matrix of the best performing classifier and dataset combination. The table shows that the right leg lunge, walk, jog, run and lying down managed to achieve 100% classification accuracy. The worst performing activity was the hinge with an accuracy of 87.2%, being misclassified

© University of Pretoria

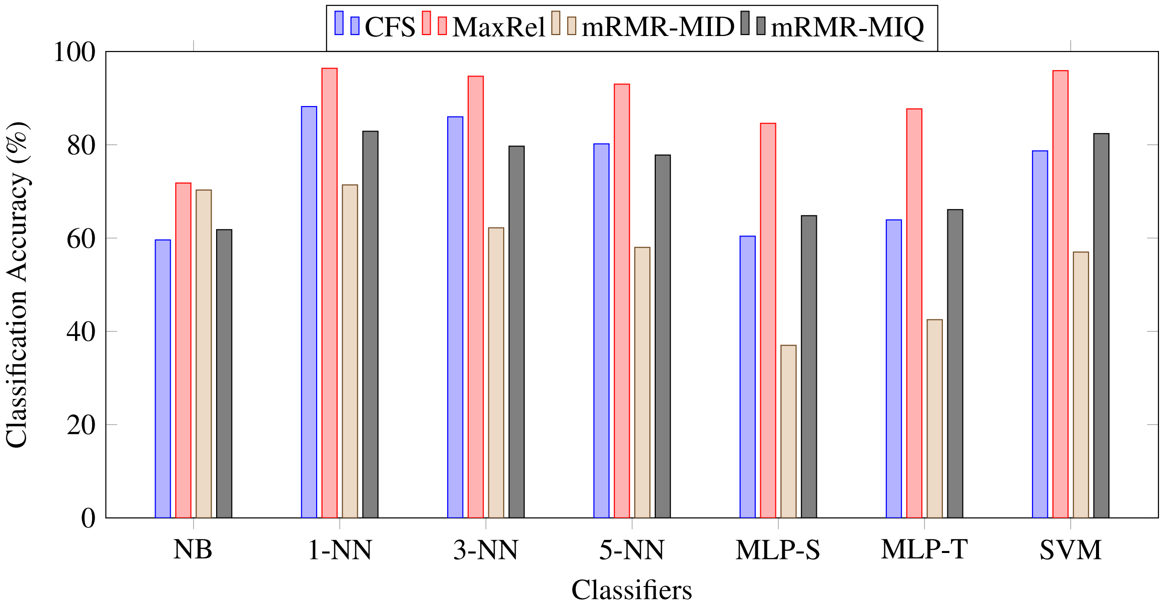


Figure 4.8. Classification accuracies for all feature subset and classifier combinations for the left thigh of Model 3.

as squatting 10.2% of the time or left leg lunges 2.6% of the time. The left leg lunge was misclassified as a squat or right leg lunge in 4.6% of the observations or a hinge in 1.5% of the observations.

Table 4.5. Left and right thigh features.

Left - MaxRel			**Right - CFS**		
Sensor	Axis	Feature	Sensor	Axis	Feature
Accelerometer	Y	Mean	Accelerometer	Y	Mean
Gyroscope	X	Fund. Frequency	Gyroscope	X	Fund. Frequency
Gyroscope	Z	Median	Gyroscope	X	Median
Gyroscope	Z	Fund. Frequency	Gyroscope	Z	Fund. Frequency
Gyroscope	X	Mean	Gyroscope	Z	Mean
Gyroscope	Z	Variance	Magnetometer	Y	Interquartile Range
Magnetometer	Z	RMS	Magnetometer	Z	RMS

Shares 57.1% of features

Table 4.5 shows the features obtained from the top performing feature selection algorithm created datasets for the left and right thighs. The results for the left thigh show that the CFS dataset contained 7 features, of which 1 of them was from the accelerometer (14.3%), 5 were from the gyroscope (71.4%)

© University of Pretoria

and the remaining 1 was from the magnetometer (14.3%). The results also show that 2 features were along the X axis (28.6%), 1 was along the Y axis (14.3%) and the remaining 4 were along the Z axis (57.1%). The results for the right thigh show that the CFS dataset contained 7 features aswell, 1 of which was from the accelerometer (14.3%), 4 were from the gyroscope (57.1%) and the remaining 2 were from the magnetometer (28.6%). The features consisted of 2 along the X axis (28.6%), 2 along the Y axis (28.6%) and the remaining 3 along the Z axis (42.9%). By considering the two feature sets together, there are a total of 14 combined features. Of those 14 features, 8 were shared between the two datasets (57.1%).

4.4.4 Ankles

The results obtained from the various classifiers and datasets for the right and left ankles are shown in Figures 4.9 and 4.10 respectively.

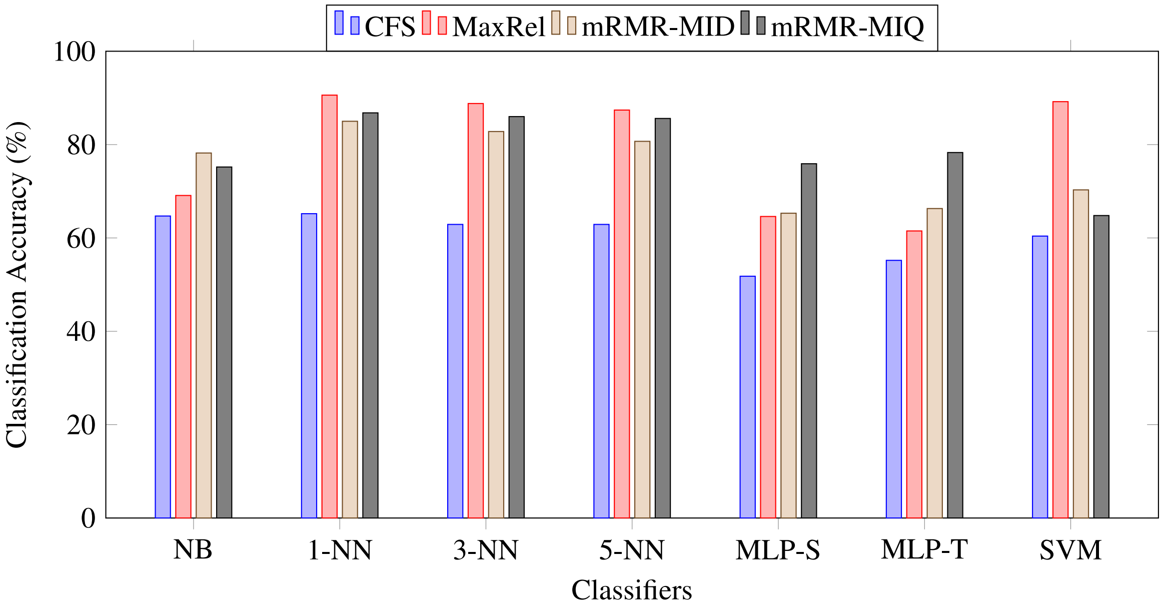


Figure 4.9. Classification accuracies for all feature subset and classifier combinations for the right ankle of Model 3.

Figure 4.9 shows the classification accuracies for the right ankle of the various datasets and classifiers. The figure shows that the MaxRel dataset was the top performer using the 1-NN, 3-NN, 5-NN and SVM classifiers; It also shows that the mRMR-MID dataset performed the best using the NB classifier and that the mRMR-MIQ dataset performed the best using the MLP-S and MLP-T classifiers. The worst performing combination was the CFS dataset using the MLP-S classifier, achieving an average

© University of Pretoria

classification accuracy of 51.8%. The top performing combination was the MaxRel dataset using the 1-NN classifier which achieved an average classification accuracy of 90.6%. This combination achieved a weighted average recall of 86.903% and a weighted average precision of 86.406%.

The confusion matrix in A.9 shows that only lying down managed to achieve 100% classification accuracy. The worst performing activities were running which was misclassified as walking, in 16.5% of the observations. A similar result was seen for squatting which was misclassified as right leg lunges, left leg lunges, hinges, running or standing. The only other two activities above a 90% accuracy were standing still and jogging.

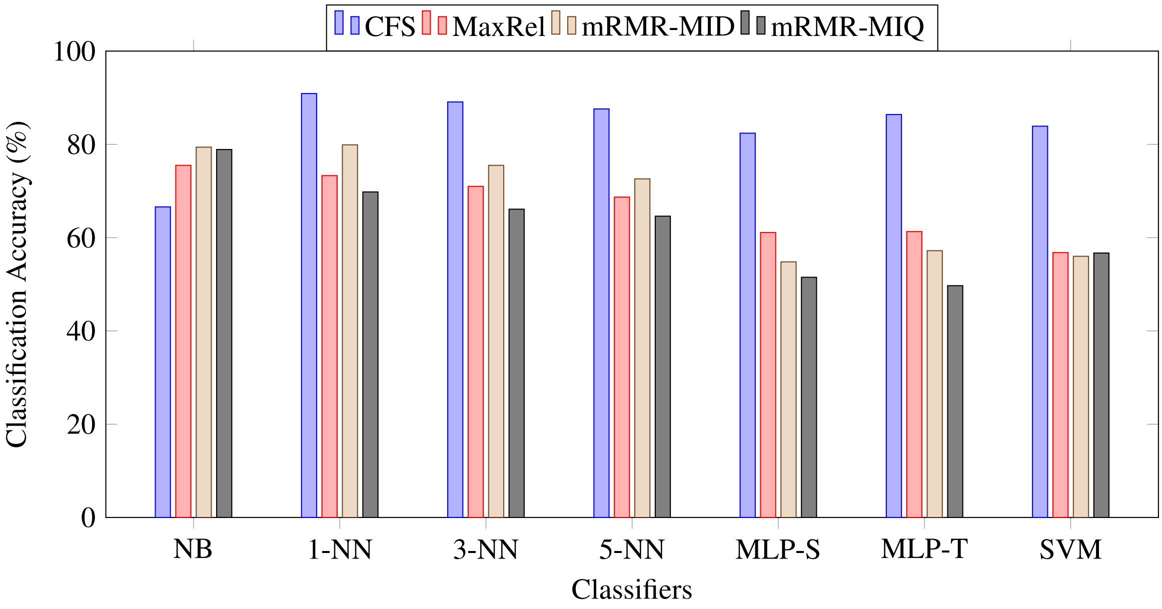


Figure 4.10. Classification accuracies for all feature subset and classifier combinations for the left ankle of Model 3.

By looking at Figure 4.10 it's possible to see that the CFS dataset was the best performing dataset using all classifiers except NB; Using the NB classifier, the mRMR-MID dataset was the best performer. The worst performing combination was the mRMR-MIQ dataset using the MLP-T which achieved a average classification accuracy of 49.7%. The best performing combination was the CFS dataset using the 1-NN classifier, which achieved an average classification accuracy of 90.9%. The combination achieved a weighted average recall of 87.168% and a weighted average precision of 87.361%.

By looking at the confusion matrix in Table A.10 the activity of lying down was once again at 100% classification accuracy. Three other activities were above 90%, standing, walking and a hinge. The

© University of Pretoria

worst performing activities were running (74.6%), which was misclassified as jogging in 14.1% of the observations and left leg lunges (79.8%) which were misclassified as right leg lunges, squatting, hinges, walking, jogging or running.

Table 4.6. Left and right ankle features.

Left - CFS			**Right - MaxRel**		
Sensor	Axis	Feature	Sensor	Axis	Feature
Accelerometer	X	Mean	Accelerometer	X	Mean
Accelerometer	Z	Median	Accelerometer	X	Median
Accelerometer	Z	Mean	Accelerometer	X	Power
Accelerometer	Y	Fund. Frequency	Accelerometer	Y	Fund. Frequency
Gyroscope	X	RMS	Gyroscope	X	RMS
Gyroscope	Y	Fund. Frequency	Gyroscope	Y	Fund. Frequency
Gyroscope	Z	RMS	Gyroscope	Z	RMS
Gyroscope	Z	Fund. Frequency	Gyroscope	Z	Power

Shares 75.0% of features

Table 4.6 shows a combined list of features from the left and right ankles. The dataset for the left ankle was generated using CFS and the dataset for the right ankle was generated using MaxRel. The left ankle's dataset contained 8 features, of which, 4 of them were from the accelerometer (50%) and the remaining 4 were from the gyroscope (50%). Of the 8 features, 2 of them were along the X axis (25%), 2 were along the Y axis (25%) and the remaining 4 were along the Z axis (50%). The right ankle's feature set also contained 8 features. Of the 8 features, 4 of them were from the accelerometer (50%) and the remaining 4 were from the gyroscope (50%). Of those 8 features, 4 were along the X axis (50%), 2 were along the Y axis (25%) and the remaining 2 were along the Z axis (25%). By considering the two datasets together there are a total of 16 features. Of those 16 features, 12 of them were shared by both datasets (75%).

Table 4.7 shows a summary of the results for all the on-body device locations, containing the best average performing dataset and classifier for each. Of the results, 5 of the 8 of the datasets were generated using CFS, 2 of the 8 were generated by MaxRel and only 1 was generated by mRMR-MID. With regards to the classifiers, the best performing classifier was the 1-NN classifier, being the best performer for all 8 locations. By considering both left and right components of the same type of

© University of Pretoria

Table 4.7. Summary of the maximum average classification accuracies for Model 3.

	Classification Accuracy	Feature Subset	Classifier
Right Upper Arm	98.7%	CFS	1-NN
Left Upper Arm	99.1%	mRMR-MID	1-NN
Right Wrist	96.3%	CFS	1-NN
Left Wrist	96.0%	CFS	1-NN
Right Thigh	96.1%	CFS	1-NN
Left Thigh	96.4%	MaxRel	1-NN
Right Ankle	90.6%	MaxRel	1-NN
Left Ankle	90.9%	CFS	1-NN
Average Accuracy	95.5%		
Standard Deviation	0.18%		
Average Recall	94.08%		
Average Precision	94.01%		

location, the upper arms performed the best with an average classification accuracy of 98.9%. The next best performer was the thighs, which obtained an average classification accuracy of 96.25%. Following the thighs, the wrists obtained an average classification accuracy of 96.15%. Finally, the ankles obtained an average classification accuracy of 90.75%. By considering all the data from all the locations, the weighted average classification accuracy is 95.5% with a standard deviation of 0.18%. The weighted average recall was 95.51% and the weighted average precision was 95.6%.

By considering the the results in this table, a less complex version of Figure 3.15 can be generated. This is shown in Figure 4.11, this figure shows the resultant path to obtaining the best results by considering the device locations independently.

4.5 MODEL 4

The following results were obtained for Model 4, which is a resultant model from the best performing model between Models 1, 2 or 3. Based on the top performing model, a single dataset or multiple datasets will be used, as well as single or multiple combinations of classifiers to perform activity

© University of Pretoria

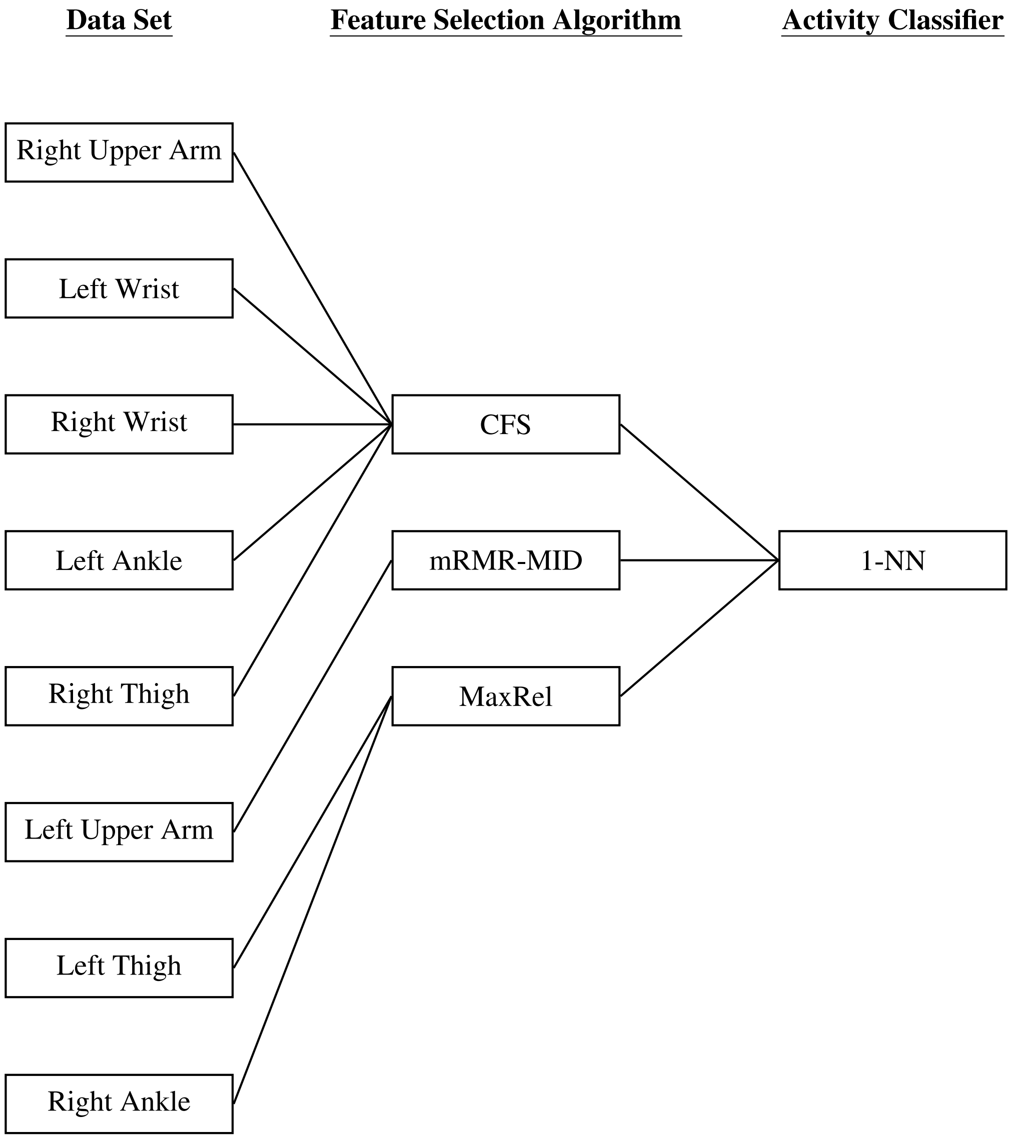


Figure 4.11. Block diagram for Model 3, post-result evaluation.

recognition. This section is broken into 2 subsections, namely, Localisation and Activity Recognition. The Localisation subsection covers the results of localising the on-body device before the location is used in the Activity Recognition section. Afterwards, the results are compared in the same fashion as all previous models.

4.5.1 Localisation

In this section the results for the localisation process are presented. All data is contained in a single dataset and feature selection algorithms and classifiers must generalise the dataset in a similar fashion

© University of Pretoria

to that of Model 1. The algorithms must localise the nodes to the right upper arm, left upper arm, right wrist, left wrist, right thigh, left thigh, right ankle and left ankle.

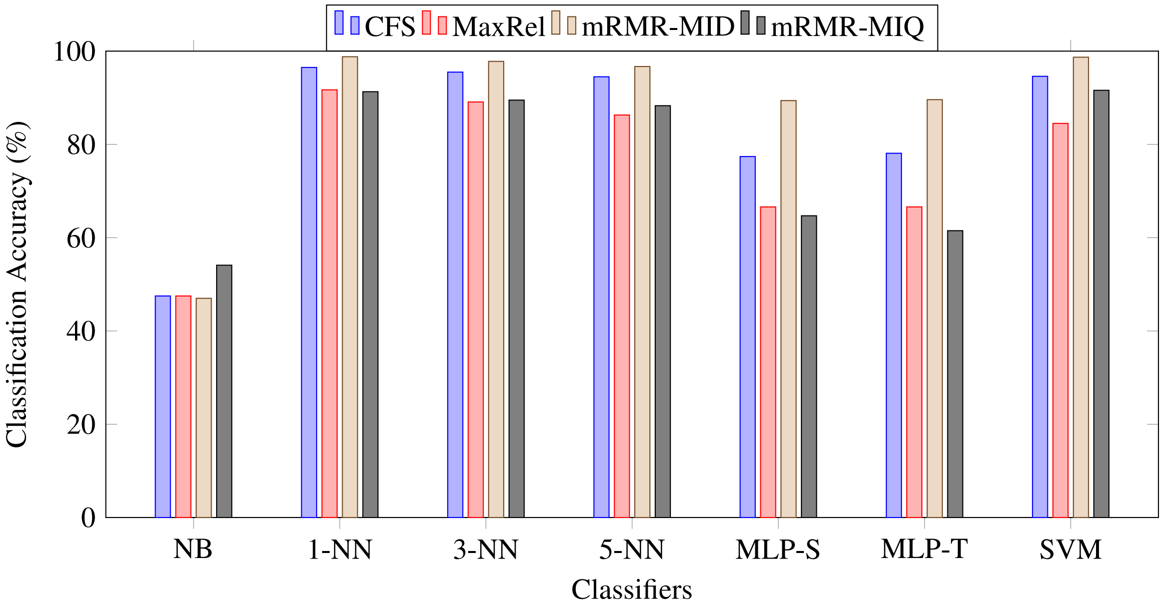


Figure 4.12. Classification accuracies for all feature subset and classifier combinations for the localisation of Model 4.

Figure 4.12 shows the results for the localisation of the nodes using the various feature selection algorithm generated datasets and classifiers. The results show that mRMR-MID is the top performing dataset using 6 of the 7 classifiers, 1-NN, 3-NN, 5-NN, MLP-S, MLP-T and SVM. The mRMR-MIQ dataset is the top performer using the NB classifier. The worst performing combination is the NB classifier using the mRMR-MID dataset, which obtained an average classification accuracy of 47.0%. The best performing combination is the mRMR-MID dataset using the 1-NN classifier which achieved an average classification accuracy of 98.8%. This was closely followed by the same dataset using the SVM which achieved 98.7% average classification accuracy. The top performer obtained a weighted average recall of 98.811% and a weighted average precision of 98.813%.

Table A.11 shows the confusion matrix of the localisation component of Model 4. To simplify the diagram, each locations label has been written as an abbreviation, where L and R stand for left and right respectively, UA stands for upper arm, W stands for wrist, T stands for thigh, and A stands for ankle. As an example, the label LUA would therefore represent the left upper arm. The results show all locations obtaining a classification accuracy of above 97%, with the right thigh and right ankle

© University of Pretoria

both obtaining 97.9% classification accuracy. The best performing location is the right wrist obtaining 99.8% accuracy.

Table 4.8. Model 4 localisation features (mRMR-MID).

Sensor	Axis	Feature
Accelerometer	X	Median
Accelerometer	Z	Median
Accelerometer	Z	Mean
Magnetometer	X	Mean
Magnetometer	Z	Mean
Gyroscope	X	Median
Gyroscope	X	Mean
Gyroscope	Y	Median
Gyroscope	Z	Mean

Table 4.8 contains a list of features that are present in the mRMR-MID dataset. The dataset contains a total of 9 features, 3 of which are from the accelerometer (33.33%), 2 of which are from the magnetometer (22.22%) and the remaining 4 are from the gyroscope (44.44%). Of the same 9 features, 4 are along the X axis (44.44%), 1 is along the Y axis (11.11%) and the remaining 4 are along the Z axis (44.44%).

4.5.2 Activity recognition

This section presents the results of the activity recognition portion of Model 4. Before the results could be generated, the model first needed to be developed, which as discussed, would be generated using the best performing activity recognition model between Models 1, 2 or 3. The results are briefly considered in Section 4.5.2.1.

4.5.2.1 Model generation

To generate a final model for Model 4, considerations of the previous 3 models need to be made. Figure 4.13 shows a comparison between the average classification accuracies of Models 1, 2 and 3.

© University of Pretoria

The results show marginal improvement from each model to the next, with Model 1 achieving 89.0% average classification accuracy, Model 2 achieving 90.2% and finally Model 3 achieving 95.5% average classification accuracy. Based on this result, Model 4's activity recognition stage will be implemented using the same method as in Model 3. The resultant model for Model 4 is shown in Figure 4.14.

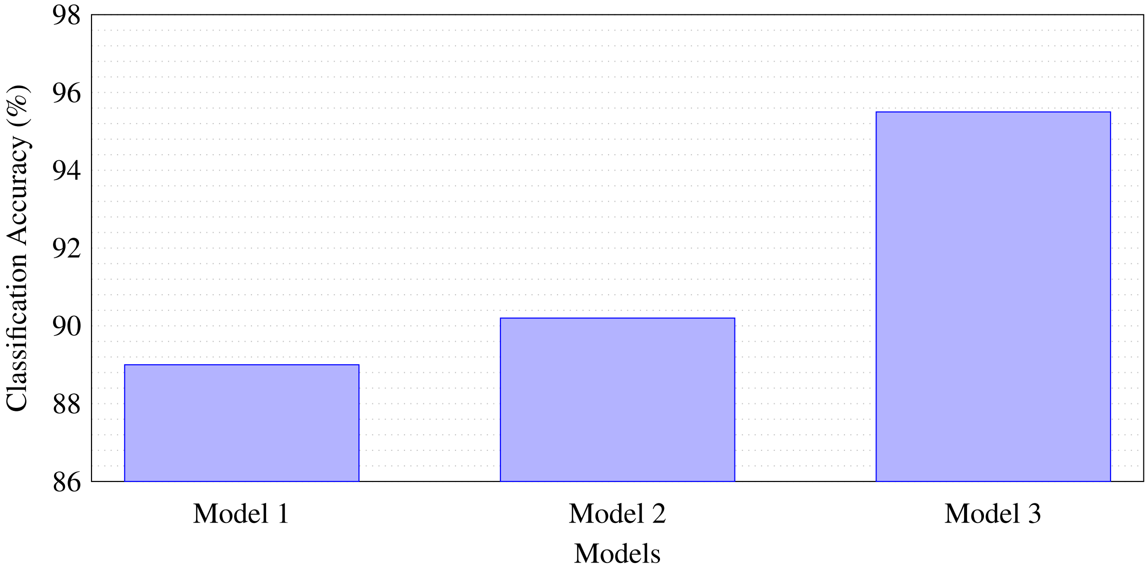


Figure 4.13. Comparison of average achieved accuracies for Models 1 to 3.

As shown in Figure 4.14 Model 4 is based on a similar approach to Model 3, where each location is considered individually with each dataset being stored separately and various classifiers and feature selection algorithms used on each dataset separately. However, instead of the known location being configured for the execution phase, localisation on the node is performed and the result is then used to choose a corresponding dataset and classifier combination.

4.5.2.2 Model results

The results for Model 4 are shown in Table 4.9. Since the model is based on Model 3, each location had to be considered individually. The results show that the upper arms achieved an average classification accuracy of 98.2%, the wrists obtained an average classification accuracy of 95.9%, the thighs obtained an average classification accuracy of 94.45% and the ankles obtained an average classification accuracy of 89.45%. Since the datasets and classifiers used were identical to that of Model 3, those results have remained the same. Considering the performance of the entire Model 4, it achieved an average

© University of Pretoria

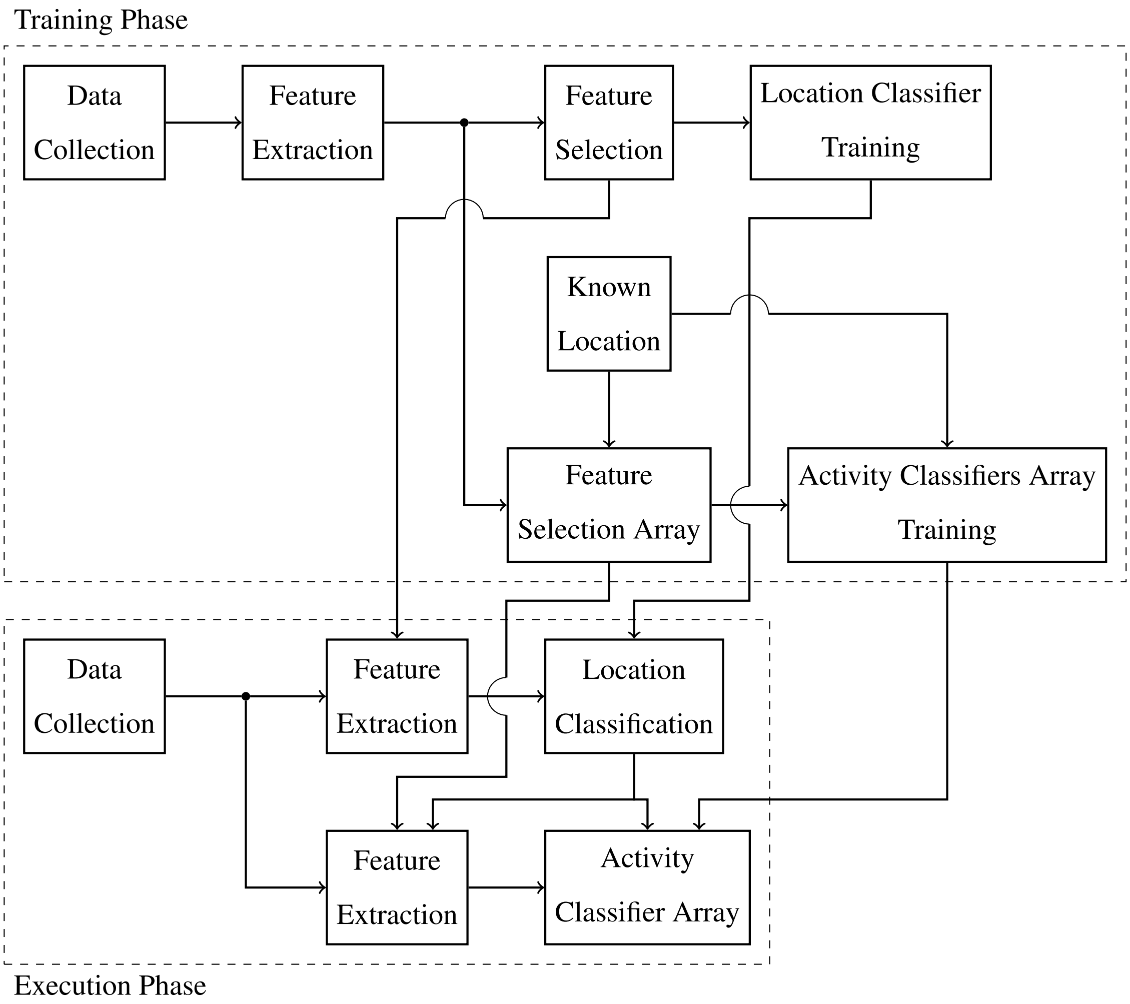


Figure 4.14. Block diagram for the final design of Model 4.

classification accuracy of 94.5%, a standard deviation of 0.27%, a weighted average recall of 93.08% and a weighted average precision of 93.01%.

4.6 MODEL COMPARISON

This section shows the results of all models in comparison with each other. It is used to highlight the differences in performances of the feature selection algorithms, classifiers, activities and a summary of the results presented in all previous sections.

Table 4.10 shows the performance of all feature selection algorithm generated datasets by averaging them across the classifiers with which they were used in each model. The results show improvements

© University of Pretoria

Table 4.9. Summary of maximum average classification accuracies for Model 4 activity recognition.

	Classification Accuracy	Feature Subset	Classifier
Right Upper Arm	98.3%	CFS	1-NN
Left Upper Arm	98.1%	mRMR-MID	1-NN
Right Wrist	96.0%	CFS	1-NN
Left Wrist	95.8%	CFS	1-NN
Right Thigh	94.5%	CFS	1-NN
Left Thigh	94.4%	MaxRel	1-NN
Right Ankle	88.7%	MaxRel	1-NN
Left Ankle	90.2%	CFS	1-NN
Average Accuracy	94.5%		
Standard Deviation	0.27%		
Average Recall	93.08%		
Average Precision	93.01%		

Table 4.10. Average performance of the feature selection algorithms for Models 1 to 4.

		Model 1	**Model 2**	**Model 3**	**Model 4**
CFS	**Average (%)**	71.2	72.5	81	80.1
	Std Dev (%)	14.6	13.7	13.8	13.9
MaxRel	**Average (%)**	52.8	68.6	77.0	76.3
	Std Dev (%)	6.6	10.3	14.3	14.2
mRMR-MID	**Average (%)**	51.2	60.9	74.6	73.9
	Std Dev (%)	7.8	7.6	14.3	14.4
mRMR-MIQ	**Average (%)**	75.2	55	70.6	69.9
	Std Dev (%)	11.3	13	14.3	14.2

in average accuracy with all FSAs except mRMR-MIQ from Model 1 to Model 2. The most significant increase from Model 1 to Model 2 is observed with MaxRel, increasing by 15.8%. A more significant increase in average accuracy is observed from Model 2 to Model 3, for all FSAs, with the most significant increase being 16.6% for mRMR-MIQ. Slight decreases are observed from Model 3 to Model 4, with a consistent standard deviation for both models. The best performing FSA was CFS for

© University of Pretoria

model 3 (81%), with the worst performing FSA being mRMR-MID for model 1 (51.2%). The FSA with the lowest standard deviation is MaxRel for model 1 (6.6%), with the highest standard deviation being mRMR-MID for Model 4 (14.4%).

Table 4.11. Average performance of the classifiers for Models 1 to 4.

		Model 1	Model 2	Model 3	Model 4
NB	**Average (%)**	56.4	57.8	70.7	69.9
	Std Dev (%)	4.1	2.0	5.4	5.3
1-NN	**Average (%)**	73.6	77	86.6	85.6
	Std Dev (%)	17.7	9.9	9.7	9.6
3-NN	**Average (%)**	71.4	73.4	84.0	83.1
	Std Dev (%)	16.3	9.7	10.2	10.1
5-NN	**Average (%)**	70.6	71.5	82.5	81.6
	Std Dev (%)	14.1	9.2	9.8	9.7
MLP-S	**Average (%)**	53.2	52	66.4	65.7
	Std Dev (%)	10.6	8.9	14.0	13.8
MLP-T	**Average (%)**	54	51.1	67.4	66.7
	Std Dev (%)	10.6	12.1	13.8	13.6
SVM	**Average (%)**	59.0	67	73.4	72.7
	Std Dev (%)	17.6	8.0	15.3	15.1

Table 4.11 shows the performance of the classifiers when averaged across all feature subsets introduced to them in each model. The results show minor improvements for all classifiers with the exception of MLP-S and MLP-T, between Model 1 and Model 2. The standard deviations improve as well with the exception of MLP-T. The largest difference in average accuracy of 8% is observed with the SVM. A significant increase in performance for all classifiers is observed between Model 2 and Model 3. Minor increases are also observed for each classifiers respective standard deviation. The largest increase in performance of 16.3% is observed for the MLP-T of Model 2 to Model 3. There are slight decreases in average accuracy from Model 3 to Model 4, with the largest difference being 1% for the 1-NN classifier.

Table 4.12 shows the average recall for the best performing classifier and FSA combination for all activities of each model. The average performance of the activities using Model 1 ranged from 76.9%

© University of Pretoria

Table 4.12. Performance of each activity for Models 1 to 4.

	Model 1	**Model 2**	**Model 3**	**Model 4**
Lunge R (%)	83.9	83.9	90.5	89.3
Lunge L (%)	83.0	83.4	92.8	91.6
Squat (%)	84.2	84.2	91.7	90.5
Hinge (%)	80.4	89.4	91.7	90.5
Walk (%)	91.8	91.7	96.9	95.6
Jog (%)	79.4	83.5	93.0	91.8
Run (%)	76.9	80.1	92.4	91.2
Stand (%)	98.4	97.5	97.8	97.6
Lie (%)	99.9	99.9	100	100

for running, to 99.9% for lying down. Two activities achieved below 80% accuracy, four achieved between 80% and 90% with the remaining three achieving above 90%. Model 2 performed similarly to Model 1 with increases for the left leg lunge, hinge, jogging and running. Slight decreases were observed for walking and standing still. The remaining activities performance remained the same. Significant increases were observed for Model 3, with increases in right leg lunges, left leg lunges, squats, hinges, walking, jogging, running, standing and lying down. The largest improvement observed was 12.3% for walking. Minor decreases are observed between Model 3 and 4. The performance of all models for standing and lying down remained very similar, with Model 1 being the best performer for standing.

Table 4.13. Average Accuracies for Models 1 to 4 including 1 and 5 standard deviations.

	Avg Accuracy (%)	**1 Std Dev (%)**	**5 Std Devs (%)**
Model 1	89.0	0.47	2.35
Model 2	90.2	0.37	1.85
Model 3	95.5	0.18	0.9
Model 4	94.4	0.27	1.35

Figure 4.15 and Figure 4.16 show the average performance of the activity recognition models with one and five standard deviations, these results are also reflected in Table 4.13. Figure 4.15 shows no overlaps between any of the four models, whereas Figure 4.16 shows a large overlap between Models

© University of Pretoria

1 and 2 and an overlap between Model 3 and 4, however, no overlap between Model 2 and 3, or 2 and 4 are observed.

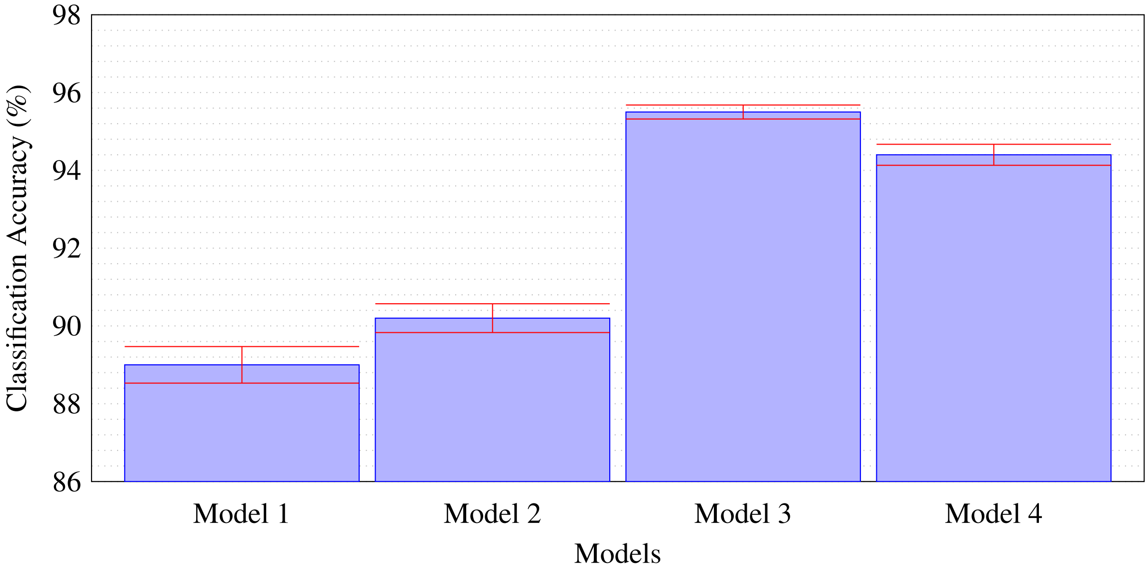


Figure 4.15. Comparison of average achieved accuracies for Models 1 to 4 within 1 standard deviation.

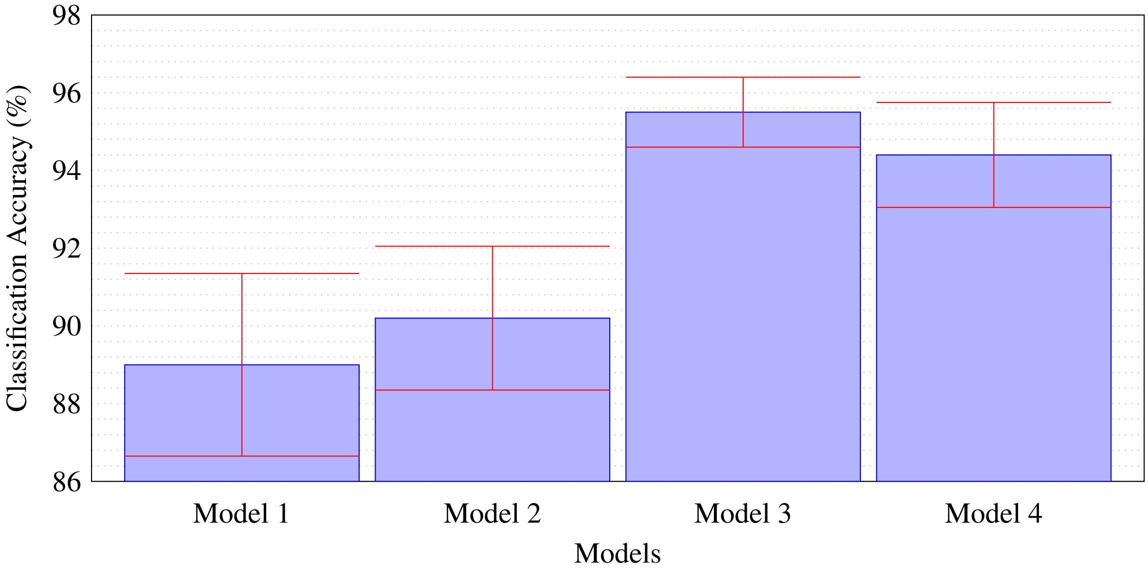


Figure 4.16. Comparison of average achieved accuracies for Models 1 to 4 within 5 standard deviations.

Table 4.14 shows a summary of the results for Models 1 to 4. It contains such details as the best performing classifier, best performing feature selection algorithm, various details about the feature sets including the sensor, axis, count and overlap. It also summarises the average classification accuracy, standard deviation, recall and precision for all models.

© University of Pretoria

4.7 CHAPTER SUMMARY

This chapter presents the results of the experiments performed, as discussed in Chapter 3. The classification accuracy, weighted average recall, weighted average precision and standard deviations of all classifiers using all feature selection algorithm generated datasets are presented for Models 1, 2 and 3. The best performing combination of classifier and feature selection algorithm for each model generates a confusion matrix, to determine where misclassifications are being made. Alongside the confusion matrix, a table containing the feature subset that yielded the best results for each model is presented. From these tables, data pertaining to the relevant sensors and axes, as well as an overlap between features are extracted.

Once the best performing of the first three models was determined, a final model for Model 4 was generated. Model 4 is then subsequently examined in the same manner as Models 1, 2 and 3, in order to generate comparable results amongst all models. A final comparison of all 4 models is made, by considering the average performance across features, feature selection algorithms and classifiers independently.

© University of Pretoria

Table 4.14. Summary of the best performing results of Models 1 to 4.

		Model 1	Model 2	Model 3								Model 4
				RUA	LUA	RW	LW	RT	LT	RA	LA	
Classifier		1-NN	1-NN	1-NN	1-NN	1-NN	1-NN	1-NN	1-NN	1-NN	1-NN	See Model 3
Feature Set		mRMR-MIQ	CFS	CFS	mRMR-MID	CFS	CFS	CFS	MaxRel	MaxRel	CFS	See Model 3
# of Features		7	8	6	7	10	9	7	7	8	8	See Model 3
Feature Axis (%)	X	0	See Model 1	33.3	42.9	40	33.3	28.6	28.6	50	25	See Model 3
	Y	57.1		0	14.3	40	22.2	28.6	14.3	25	25	
	Z	42.9		66.6	42.9	20	44.4	42.9	57.1	25	50	
Feature Sensor (%)	Acc	42.9	See Model 1	33.3	42.9	20	44.4	14.3	14.3	50	50	See Model 3
	Gyro	28.6		66.6	42.9	60	44.4	57.1	71.4	50	50	
	Mag	28.6		0	14.3	20	11.1	28.6	14.3	0	0	
Overlap (%)		N/A	N/A	61.5		63.2		57.1		75.0		See Model 3
Avg Accuracy (%)		89	90.2	95.5								94.5
Std Deviation (%)		0.47	0.37	0.18								0.27
Avg Recall (%)		86.45	88.19	94.08								93.08
Avg Precision (%)		86.86	88.63	94.01								93.01

© University of Pretoria

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

This chapter contains an interpretation and discussion of the significance of the results found in Chapter 4; This chapter also includes explanations of new insights into the research problem. In Section 5.2, various aspects of the feature subsets including feature axis and sensors used are discussed. A review of the performance of the various configurations of feature selection algorithms and classifiers are presented in Sections 5.3 and 5.4 respectively. Section 5.5 evaluates the activities based on the results of the machine learning models. A discussion on the approach to localising devices and then performing activity recognition in Model 4 is presented in Section 5.6. Finally, an evaluation on the impact of localising a wearable device on the activity recognition problem is presented in Section 5.7.

5.2 FEATURES

To gain an understanding of the activity recognition problem it is also important to examine the features that were chosen by the feature selection algorithms. To do this, reference must be made to all tables in the Results chapter which contain the best performing feature set for each model, as well as Table 4.14.

5.2.1 Feature axis

Model 1 contains 7 features in the dataset, of which 0% were along the X axis, 57% along the Y axis and 42.9% along the Z axis. Model 2 contains 8 features, of which the first 7 are identical to that

© University of Pretoria

of Model 1, with the 8th feature being the on-body device location. However, Model 3 on average contains 35.5% of its features along the X axis, 22.6% along the Y axis and 41.9% along the Z axis. The same would obviously apply to Model 4 since it uses the same datasets as Model 3. The result from Model 3 does however show that when considering each location individually there is a much more even distribution amongst the feature axes when compared to Model 1. However, the Z axis showed comparable results for both Model 3 and 1. This result indicates a strong relevance of information along that axis. When considering these results a deduction can be made that every axis contains relevant information for these specific activities. It is however possible, that given a different set of activities, the strong relevance of the Z axis may fall away and another axis may become more predominant or all the axes may become more equally relevant.

5.2.2 Feature sensor

A similar comparison to the feature axis can be made while considering the different sensors that were sampled. Model 1's feature subset consisted of 42.9% from the accelerometer, 28.6% from the gyroscope and 28.6% from the magnetometer. Model 3 however contained 33.7% from the accelerometer, 55.3% from the gyroscope and only 11.0% from the magnetometer. Once again, Model 2 and 4's results are the same as Model 1 and 3 respectively. By comparing the results of Models 1 and 3, a significant shift in the sensor which provides the most information can be observed. Model 3 places a much higher relevance on the gyroscope, indicating that more information is gained from the orientation of the device than its speed or direction. Model 3 also uses significantly less information from the magnetometer. These results do seem to line-up with the activities that were recorded, as much more focus was placed on the technique of the activities than the speed or direction in which they were performed. This is however not the case for walking, jogging and running as the difference between those activities are determined mostly by speed, with some changes in the mechanical aspects of the body, for example a longer stride length [50].

5.2.3 Feature count

Another aspect to examine is the amount of features in the various feature sets. Model 1's feature set contained 7 features, Model 2 contained 8, Model 3's upper arms contained 6 and 7, the wrists, 10 and 9, the thighs, 7 and 7 and finally the ankles contained 8 and 8. When considering the models training

© University of Pretoria

datasets, Model 1 contained the least prior knowledge and had one large dataset. This means the model needed to generalise 9 activities, with the devices placed at 8 locations, making it a complex problem to solve. As previously stated, Model 2's feature set contained 8 features, which included an encoded version of the on-body device location as a feature. This made the feature space more complex by introducing one extra dimension, however the accuracy of the model did increase in comparison to that of Model 1. Model 3 however, better highlights the complexities of the problem as it considers each location individually. On average the upper arms required the least features, the thighs then required the next least, the ankles required the second most with the wrists requiring the most features. Of those locations however, the ankles on average performed the worst, with the thighs and wrists performing almost identically while the upper arms performed the best.

The amount of features does however appear to have some correlation with the full range of motion of the limb to which the device is attached. The position and orientation of the wrist for example, is affected by the elbow and shoulder, however the upper arms are only affected by the shoulder [51, 52]. The reduced amount of degrees of freedom results in a less complex model for the device location, which would intuitively allow the limbs activities to be modelled by less features [53]. The ankles and the wrists are both affected by a similar amount of degrees of freedom, resulting in the largest amount of features to model them. The upper arms and thighs are also affected by a similar amount of degrees of freedom, resulting in a lower amount of features required to model them. Model 3 has a total of 62 features, which corresponds to an average of 7.75 features per device location. In comparison with Model 1 which uses 7 features, Model 3 better generalises the activities resulting in a 6.5% classification accuracy increase. This means that an increase in complexity of the problem, does not guarantee a decrease in accuracy. Therefore, it can be said that Model 3, on average has higher quality features than all other models.

5.2.4 Feature overlap

Feature overlap is something specifically affecting Model 3, as it examines each on-body device location individually. It is expected that the left and right versions of the same limb will produce a similar classification accuracy, because the motions when performing the activities will be somewhat mirrored by both limbs [54]. It is therefore also expected that the feature set will also be somewhat similar. The upper arms have a 61.5% overlap, the wrists have a 63.2% overlap, the thighs a 57.1%

© University of Pretoria

overlap and the ankles a 75% overlap. The fact that an overlap exists in the results is expected however the amount of overlap was unknown prior to the experiments.

This overlap is thought to be able to model a neutral version of a limb, e.g. not considering a left and right wrist, but rather just a wrist. This would reduce the amount of features in the feature set for each location, with the ankles containing 6 features, the thighs, 4 features, the wrists, 6 features and the upper arms, 4 features. It was expected that there would be a portion of the feature set that does not overlap between both sides and it is thought that this uniqueness is generated by unique movements performed by the specific side. For example, a right-handed person lifts up two bags, one with each hand. It is thought that some micro-movements may be made by the right arm that the person unknowingly does not do with the left arm. This difference in the movements between the left and right arms could therefore be viewed as noise, thus generating the unique portions of the feature sets. This noise could be contributing to overfitting of the classifiers, therefore reducing the overall accuracy.

5.3 FEATURE SELECTION ALGORITHMS

This section examines the performance of the various feature selection algorithms. The results of each of the feature selection algorithms are compared based on their performance as a result of the classifiers that use the feature subsets they each generate. This discussion is based on the results presented in Table 4.10.

The results in the table show an increase in performance for the CFS, MaxRel and mRMR-MID FSAs from Models 1 through 3. This was however not the case for mRMR-MIQ which best performed in Model 1. The increase in performance between the models indicates that the extra prior knowledge in the system was able to add to the quality of the features that the FSA was able to extract. A significant increase from MaxRel and mRMR-MID was noted from Model 1 to Model 2. This increase cannot solely be attributed to the FSAs because the same features were used for both Model 1 and 2, with the exception of the encoded location of the device included in Model 2's feature set. This indicates that although the features are almost identical, an added degree of complexity can greatly increase the average performance of the system, allowing better discrimination of the features from one another. A similar increase can be seen when comparing Model 2 to Model 3. This can be attributed to the FSAs

© University of Pretoria

being able to examine each location individually thus determining which features are most relevant for each location. Similar results can be seen when comparing Model 3 to Model 4, with the primary difference being introduced by the localisation error for each device.

These average results do agree with those shown in Table 4.14 as the best performing FSA for Model 1 was mRMR-MIQ, CFS for Model 2 and a distribution of CFS, MaxRel and mRMR-MID for Model 3. The results do appear poor in comparison to the results shown in Table 4.14 due to the fact each FSA is being compared using the average across all classifiers. This means that the poorly performing classifiers are greatly reducing the average performance of the FSA. This can be noted by the fairly high standard deviations for all FSAs. A high standard deviation indicates that the FSA was able to perform well with some classifiers and poorly with others. A low standard deviation however indicates that the FSA performed somewhat consistently across all classifiers, which can be seen for Model 1 MaxRel and mRMR-MID, as well as, Model 2's mRMR-MID. It is possible that by tuning the machine learning models, there may be moderate increases in the overall performances of each of the FSAs.

5.4 CLASSIFIERS

This section examines the performance of the classifiers using all feature selection algorithms. The discussion is based on the results presented in Table 4.11.

Similarly to the increases seen in the feature selection algorithms for every model, a distinct increase in average classification accuracy can be seen for all classifiers with the exception of both MLPs for Models 1, 2 and 3. The 1-NN classifier was the overall top performing classifier across all experiments and models. This was followed by the 3-NN and 5-NN classifiers respectively. An interesting observation however, is the gradual decrease in the average classification accuracy as k increases. This gradual decrease in performance for the same model is likely linked to a property of the k-NN classifier where the boundaries become more smooth as k increase; With $k = 1$ the boundaries are sharp and very distinct. It is possible that by using an even-numbered value for k the experiments may have produced different results, by forcing the tie-breaking strategies of the classifier to be used. The 1-NN classifier did manage to improve its performance and confidence from Models 1 through 3, which can

© University of Pretoria

be seen by its decreasing standard deviation. This indicates a strong performance from the classifier irrespective of the dataset being used.

The overall worst performing classifiers were undoubtedly the MLPs, which managed an average classification accuracy of only 66.4% and 67.4% each, in their best performing activity recognition model. This is almost certainly as a result of the many parameters that must be chosen for the MLP to function. This includes the number of hidden layers, number of neurons per layer, the learning rate, etc. It is assumed that given a better estimation of those parameters, the MLPs would have performed significantly better as neural networks do form part of deep learning. In contrast to the classifiers that require significant configuration, the classifiers which require almost none, performed admirably.

Increasing the overall performance of every classifier may be achieved using several methods. The first being the one already raised for the MLPs where tuning may have been an issue. A second method for improving performance may be the use of a different class of feature selection algorithm such as Wrapper FSAs. It appears that many of the classifiers are sensitive to the features that they are provided with, so it is possible that given a different set of features, the worst performing classifier may become the best performing classifier.

5.5 ACTIVITY RECOGNITION

This section examines the best performance of each model, which are made up of best combinations of feature selection algorithms and classifiers. The discussion is based on results from Table 4.14 and Table 4.12.

With particular reference to Table 4.14, the results show marginal increases from Models 1 to 3. These increases can be attributed to the manner in which the datasets are being treated, based on their known location information. The table shows a 1.2% increase from Model 1 to 2 and a 5.3% increase from Model 2 to 3. This jump in average classification accuracy can largely be attributed to the manner in which Model 3 performs feature selection, as it is able to consider each device location individually, allowing the system to generalise each location independently of others. Similar to previously discussed material, the standard deviation reveals information about the consistency of the results. Model 1 had

© University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

a standard deviation of 0.47%, Model 2 had 0.37% and Model 3 had 0.18%. This steady decrease shows that each model becomes more robust to noise and is able to produce a consistent result across multiple experiments. Based on these results, Model 3 has shown a very consistent performance, in which none of the experiments dropped below 95%. When choosing the top performing classifier, the 1-NN classifier was clearly dominant in the entire experiment. There were however times where other classifiers came close to being the top performer, but ultimately the 1-NN classifier produced the highest average classification accuracy.

Since Model 3 was the best performing model, reference can be made to Table 4.7 for a closer look at the which limbs were able to produce the best results. As can be seen, the upper arms did almost equally well achieving 98.7% and 99.1%. The wrists and thighs performed quite similarly achieving approximately 96%. A significant drop in performance is however observed when considering the ankles, which achieved 90.6% and 90.9%. There are several factors that may contribute to the decrease in performance in comparison to other limbs. One possible explanation is that the activities may be very similar from the perspective of the ankles. This would lead to the features being similar, thus not allowing for an acceptable generalisation of the data.

On the other hand if the activities are easily distinguishable from each other from the perspective of the ankle, then the feature selection algorithms simply did not extract high quality features. Another explanation may be to the fault of the classifiers themselves, where it would be a case of tuning. One other explanation may be that a completely different set of features may need to be extracted during the pre-processing stage, thus allowing all subsequent stages to perform better. One aspect to consider in this type of application is what is deemed to be an acceptable classification accuracy. If anything that falls below 95% is deemed unacceptable then the ankles do require improvement. However, in these experiments, the ankles from Model 3 do perform better than that of Model 1 and 2. This once again reinforces the notion that the contextual information is adding value.

To determine if there are particular activities which were difficult to generalise, reference should be made to Table 4.12. When considering Model 1, the activities that proved the most difficult to classify were running and jogging. As discussed previously, most of the focus during data collection was placed on the technique of the activity, rather than the speed at which it was performed. It is assumed that the performance of these activities can be attributed to two possible scenarios, or a combination of both. The first scenario is that the test subjects, when instructed to run and jog, performed activities

© University of Pretoria

that were in fact almost identical. This would lead to the data being very similar, making the activities difficult to distinguish. A second scenario, assuming the test subjects performed the activities perfectly, is that the activities are in fact very similar.

This would mean that a better set of features would be required to accurately describe the differences between running and jogging. The best performing activities were standing and lying down. The assumption is that those activities are the best performing because contrary to the others, they are completely static. Standing however does not achieve almost perfect classification accuracy and the assumption is that this is because every activity with the exception of lying down, are all performed while standing up. When considering Model 2, the performance of the activities is very similar for all, with the exception of a 9% improvement with hinges and slight improvements on running and jogging. This gives some indication that more, or better features are required to accurately describe those particular activities in general. The biggest improvement is observed in Model 3, with no activities falling below 90% classification accuracy. Similar to previous discussions it is assumed that this is entirely attributed to each location being considered individually, allowing for better generalisation of each activity. There is however little to no difference for standing and lying down, as those activities appear easily distinguishable regardless of the machine learning model.

5.6 MODEL 4

5.6.1 Localisation

The localisation of Model 4 is reminiscent of Model 1's activity recognition, where both models use a single dataset to classify the data. However, for activity recognition there are 9 classes but with the localisation there are only 8 classes. Because both models use a single dataset, the localisation classifier has to determine the location of the device regardless of the activity, inversely the activity recognition classifier has to determine the activity, regardless of the location. From Figure 4.12, one of the differences that can be observed between the classification accuracies of Model 1's activity recognition and Model 4's localisation is the difference in classification accuracy, which were 89.0% and 98.8% respectively. This difference can most likely be attributed to the fact that the activities contain data from the perspective of both sides of the body, whereas the locations of the device can be differentiated by the side of the body. So although the left and right wrist may produce directly

© University of Pretoria

opposite versions of the same movement, the activity recognition classifier has to know of both the positive and negative versions which represent the same activity, where as the localisation only has to know that the positive movement would represent the right wrist and the negative would represent the left wrist in a single cycle. This would in turn make the classification task much more trivial.

By considering the features shown in Table 4.8, results comparable to those of Model 3 are observed. Model 4 localisation used 33.3% of the total features from the accelerometer, 44.4% from the gyroscope and 22.2% from the magnetometer, whereas Model 3 activity recognition contained 33.7% of the total features from the accelerometer, 55.3% from the gyroscope and 11% from the magnetometer. These similar results reinforce the idea that the gyroscope provides the most information, followed by the accelerometer and lastly the magnetometer. The feature axes for the Model 4 localisation showed 44.4% of features along the X axis, 11.1% along the Y axis and 44.4% along the Z axis, whereas the Model 3 activity recognition feature set contained 35.5% along the X axis, 22.6% along the Y axis and 41.9% along the Z axis. This again reinforces the idea that the X and Z axes contain most of the information with the Y axis containing only some information. This gives the impression that the information gain from those sensors and directions are somewhat consistent, when using the same dataset for multiple applications. This is not to say that these proportions are universally applicable to all applications. When considering the feature count, Model 1's activity recognition required 7 features with Model 4's localisation requiring 9 features. The increase in feature count again shows that an increase in complexity does not necessarily reduce accuracy. It does however show that the performance of the model is very sensitive to the features in the feature subset.

With regards to the classifier performance shown in Figure 4.12, strong performance can be seen by multiple datasets using many classifiers. This highlights the importance of the dataset in the performance of the system in solving the problem, as the configurations of the classifiers are no different to that of Model 1. The NB classifier however still remained the worst performer by a large margin, in comparison to all other classifiers. The reason for this is unknown, however it is assumed that the various activities performed included noise which the NB classifier was not able to deal with.

© University of Pretoria

5.6.2 Activity recognition

Since the ultimate goal of Model 4 is to try replicate the result of Model 3 as closely as possible, consider the results shown in Table 4.9, Table 4.12, Table 4.10 and Table 4.11. A brief look at Table 4.9 shows that the classification accuracies for the activities at each limb are very comparable to those of Model 3. The right ankle however, suffered a 1.9% performance decrease and the left thigh saw a 2.0% performance decrease. It is assumed that all the decreases are caused by the error introduced by the localisation of each of the nodes. If the node is localised incorrectly, the incorrect training dataset would be chosen and the incorrect features would then be extracted which would likely lead to a misclassification of the activity as well.

If considering the performance of the classifiers or feature selection algorithms as shown in Table 4.10 and Table 4.11, once again a very comparable performance is noted. As expected, Model 4 does overall, perform slightly worse when compared to Model 3, however, there are notable improvements in comparison to Models 1 and 2. By looking at the individual activity performance shown in Table 4.12, once again the performance is comparable to that of Model 3, with a maximum of 1.3% performance decrease noted for walking. Standing and lying down performed roughly the same, indicating that even if the location of the device was classified incorrectly, the activity recognition classifier would still accurately determine the activity being performed. Once again, it can be said that these minor dips in performance can be attributed to the misclassification of the on-body device location, leading to the chain of events as described above.

5.7 EFFECT OF LOCALISATION

The research investigated four methods of using the on-body device location and its impact on solving the activity recognition problem. The first method makes no use of the location whatsoever, this was used as a benchmark or a control for direct comparison between models. The second method made use of the encoded location of the device and appended it as a feature into the dataset, which was shared with the first model. The third method used the location logically, to separate the training datasets, meaning that each location would solve the activity recognition problem individually. Finally, a model for autonomously detecting the location of the device was proposed, which then performed activity recognition based on the best performing of the first three models.

© University of Pretoria

With reference to Figure 4.15 and 4.16, the effects of localising the on-body device can somewhat be seen in terms of the standard deviations of each model. In particle physics, the certainty of the existence of the Higgs Boson particle was based on a 5 standard deviations criteria [55]. For this research the same criteria has been applied. When considering Model 1 and Model 2 only, there is an overlap between the model performance within 5 standard deviations. This suggests that using the location of the device as a feature in the subset did not prove to be statistically significant. However, considering Models 2 and 3, even across 5 standard deviations there are no overlaps observed, so it could be argued that the improvement between those models was statistically significant. Although the increase in average performance observed between Models 2 and 3 is significant, it is not to say that this is entirely due to the logical use of the location.

As discussed previously, with the system considering each location individually, the machine learning algorithms were able to better generalise the training data. So although the location did not implicitly impact the classifiers or feature selection algorithms themselves, the effect of limiting the feature space by means of the location are indeed observable. When considering Models 3 and 4, an overlap in performance can be observed within 5 standard deviations, which indicates that the difference in these models performance is not statistically significant. This means that although Model 4 is autonomously localising the node and the activity recognition performance is negatively affected by the error introduced by that localisation, its impact is not significant. When considering Models 2 and 4, there is no overlap across 5 standard deviations, again reinforcing the concept that the logical use of the location does make a significant impact on the activity recognition problem, even when including errors introduced by the nodes localisation.

5.8 CHAPTER SUMMARY

In this chapter, the results presented in Chapter 4 were discussed in terms of their effect on the activity recognition problem. This was done by examining the various aspects affecting the performance of the classifiers, feature selection algorithms, sensors, sensor axes, activities and ultimately the on-body device location. The effect of using the wearable devices' location, is evaluated based on statistically significant results generated by each model.

© University of Pretoria

CHAPTER 6 CONCLUSION

The research aimed to determine the effect of localising a node on the performance of an activity recognition system. Four activity recognition models were developed, each using the location of the device in a different and unique manner. The first model contained no information of the location of the device and kept all data in a single dataset. Using this method means that the feature selection algorithms and classifiers had to generalise the dataset for nine different activities, irrespective of the placement of the device. The second model inserted an encoded version of the location in the feature space.

Similarly to the first model, the feature selection algorithms had to generalise the single dataset, however the dataset contained some logical information of the location of the device. The third model treated each location as an independent problem and used the location of the device to determine which classifier and feature selection algorithm to use. This model made use of eight separate datasets, each containing data only relevant to the specific location of the device. A final model was proposed which autonomously localises the device and uses that on-body device location output in the same manner as the best performing of the first three models.

Each activity recognition model was evaluated using a brute-force search where all combinations of feature selection algorithm and classifier are evaluated. This method generated several results, measurable in terms of average classification accuracy, weighted average recall, weighted average precision and standard deviation. The results were evaluated in terms of feature selection algorithm, classifier and activity recognition performance, as well as an evaluation of metrics of the feature sets themselves including the sensor, axis, feature count and overlap of features between similar limbs.

6.1 FEATURES

By considering the feature axes of the feature sets for each model, it was revealed that the X and Z axes contained the most information. This became apparent because the performance of Model 3 was superior to that of Model 1 or 2; Model 3 contained most of its features along the X and Z axes. This could however change if other activities were being classified. Considering the feature sensors, it was revealed that the accelerometer and gyroscope contained the largest amount of information. Once again this was observed because of the performance of Model 3 and the strong representation of those sensors in the feature sets. The feature count showed that the limbs with the most freedom to move were the most complex to model, which was observable from the increased amount of features in the feature set for those limbs. The feature overlap affected only Model 3 because of its evaluation of the locations individually. The feature overlap percentage revealed that each location on the left and right side contained information common to both, which may mean that if they were considered together, those common features would possibly be able to model the same activities for those combined locations.

6.2 FEATURE SELECTION ALGORITHMS

By considering the feature selection algorithms, it was clear that the CFS feature selection algorithm was the overall best performer. The high standard deviations for each feature selection algorithm in each model indicates that the performance of the dataset depends heavily on the classifier used, as the difference in the performance of each classifier using the same dataset varies widely. It can therefore be concluded that no specific feature selection algorithm is the obvious choice for use in this application, as the performance depends on parameter tuning of the feature selection algorithm and the classifier combination. This gives the impression that using wrapper or embedded feature selection algorithms would increase the systems overall performance.

6.3 CLASSIFIERS

When considering the classifiers, the 1-NN classifier was the clear best performer for all models. It appears that the rigidness of the decision boundaries of the classifier aided its performance when compared to the other k-NN classifiers with larger k values. The worst overall performer was the

© University of Pretoria

NB classifier, with this observation stemming from its poor average performance coupled with a consistently low standard deviation. The multilayer perceptrons also performed poorly, however this can largely be attributed to poor choices and tuning of the parameters of the classifier, such as, learning rate, number of neurons, number of hidden layers etc.

Certain classifiers have a high standard deviation in performance, which indicates that the classifiers are sensitive to the quality of features being passed to it. With these and the above considerations it can be said that there appears that no set of features which will completely accurately model a users activity regardless of the location of the device. This conclusion can be drawn because the overlap between the paired locations is not 100%, meaning that some features help model specific movements for specific body parts on either side. The overall activity recognition performance can be improved by adjusting parameters of both the feature selection algorithms and classifiers. It could be assumed that the classifier may be able to make up for a poor set of features by being tuned properly, likewise a classifier could improve its performance with a properly tuned feature selection algorithm.

6.4 ACTIVITIES

When considering the activities themselves, it appears that the activities which overlap in terms of the motion required to perform them, are the ones whose average accuracy suffer the most; This can be seen when considering squatting and hinges, which contain some motions common to both. A similar observation can be made with jogging and running, which is largely a distinction determined by the speed of the subject. The activities which are static and require little to no movement are the ones that are most easily distinguishable, which was observed in the results with standing still achieving 97.8% accuracy and lying down achieving 100%. With this information, a conclusion can be made that a systems activity recognition performance is largely dependant on the activities chosen to be classified. Activities which contain overlaps in their movements, may negatively affect the performance of the system.

6.5 EFFECT OF LOCALISATION

The effect of localising a node in an activity recognition system, can be observed when considering the performance of each model. As stated in the Discussion, the 5 standard deviation criteria was chosen

© University of Pretoria

for this research. When referencing the relevant figures it is clear that the insertion of the location as a feature in the feature set did not prove to be statistically significant. However, by using the location to logically separate the datasets, then determining the activity based on data from the relevant location only, a statistically significant increase in average performance was observed. This indicates that the location has a definitive impact on the activity recognition problem, even if not implicitly used in the feature selection algorithms or classifiers. A conclusion can be made that considering each location individually makes an impact on the activity recognition system's performance, because the machine learning models could better generalise the activity without having to consider data from other limbs. Although only two methods of using the location were investigated, it is clear that there is a strong relevance in giving context to the activity recognition problem in the form of device location, however the exact amount of influence remains unknown. It can therefore be concluded that the use of the location in another manner may also contribute to the increase in performance of the machine learning model.

By considering the performance of Model 4, it is clear that the autonomous localisation of the device introduces some error. The results show that there is no statistically significant decrease in activity recognition performance between Models 3 and 4. There is however a statistically significant difference between Model 2 and 4s activity recognition performance, which further reinforces the concept that increasing the complexity of the system does not guarantee a decrease its performance. It can therefore be concluded that an autonomous system is capable of achieving results close to that of a system with pre-configured location information.

6.6 FUTURE WORK

Investigations should be made into determining the effects of other contextual information that may be relevant to the problem. Alternatively, other approaches in the application of the on-body device localisation or other contextual information may be investigated.

© University of Pretoria

REFERENCES

[1] K. Henricksen and J. Indulska, "Modelling and using imperfect context information," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, Mar 2004, pp. 33–37.

[2] J. Garcia, "Sensors: Types and characteristics," Illinois Institute of Technology, 2012, http://tinyurl.com/y4heds6v, [Accessed 2017-08-12].

[3] T. Mai, "What are passive and active sensors?" Aug 2017, https://tinyurl.com/y6v86k8k, [Accessed 2017-10-15].

[4] C. Schlenoff, T. Hong, C. Liu, R. Eastman, and S. Foufou, "A literature review of sensor ontologies for manufacturing applications," in *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, Oct 2013, pp. 96–101.

[5] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, May 2016.

[6] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031–2038, 2013.

[7] S. Hochreiter, A. S. Younger, and P. R. Conwell, "Learning to learn using gradient descent," in *International Conference on Artificial Neural Networks*. Springer, 2001, pp. 87–94.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Pretten-

© University of Pretoria

hofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[9] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, no. 4, p. 049901, 2007.

[10] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM computing surveys*, vol. 27, no. 3, pp. 326–327, 1995.

[11] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 1–33, Jan. 2014.

[12] I. Cleland, B. Kikhia, C. Nugent, A. Boytsov, J. Hallberg, K. Synnes, S. McClean, and D. Finlay, "Optimal placement of accelerometers for the detection of everyday activities," *Sensors*, vol. 13, no. 7, pp. 9183–9200, 2013.

[13] J. Wannenburg and R. Malekian, "Physical activity recognition from smartphone accelerometer data for user context awareness sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3142–3149, Dec 2017.

[14] A. Jain and V. Kanhangad, "Human activity classification in smartphones using accelerometer and gyroscope sensors," *IEEE Sensors Journal*, vol. 18, no. 3, pp. 1169–1177, Feb 2018.

[15] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A survey of wearable devices and challenges," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2573–2620, Fourthquarter 2017.

[16] "IDC forecasts worldwide shipments of wearables to surpass 200 million in 2019, driven by strong smartwatch growth and the emergence of smarter watches," December 2017, https://www.idc.com/getdoc.jsp?containerId=prUS43408517, [Accessed 2017-12-28].

[17] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*

© University of Pretoria

(Applications and Reviews), vol. 40, no. 1, pp. 1–12, Jan 2010.

[18] N. A. Alrajeh, M. Bashir, and B. Shams, "Localization techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 6, p. 304628, 2013.

[19] G. Lo, S. González-Valenzuela, and V. C. Leung, "Wireless body area network node localization using small-scale spatial information," *IEEE journal of biomedical and health informatics*, vol. 17, no. 3, pp. 715–726, 2013.

[20] D. de Arruda and G. P. Hancke, "Wearable device localisation using machine learning techniques," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, June 2016, pp. 1110–1115.

[21] B. E. Ainsworth, W. L. Haskell, M. C. Whitt, M. L. Irwin, A. M. Swartz, S. J. Strath, W. L. O Brien, D. R. Bassett, K. H. Schmitz, P. O. Emplaincourt, D. R. Jacobs, and A. S. Leon, "Compendium of physical activities: an update of activity codes and MET intensities," *Medicine and science in sports and exercise*, vol. 32, no. 9, pp. S498–S504, 2000.

[22] "Api reference," May 2015, https://developer.android.com/reference/, [Accessed 2015-09-30].

[23] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "On-body device localization for health and medical monitoring applications," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar 2011, pp. 37–44.

[24] W. Xu, M. Zhang, A. Sawchuk, and M. Sarrafzadeh, "Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks," in *2012 9th International Conference on Wearable and Implantable Body Sensor Networks BSN*, May 2012, pp. 124–129.

[25] R. Saeedi, J. Purath, K. Venkatasubramanian, and H. Ghasemzadeh, "Toward seamless wearable sensing: Automatic on-body sensor localization for physical activity monitoring," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2014, pp. 5385–5388.

© University of Pretoria

[26] K. Kunze, P. Lukowicz, H. Junker, and G. Tröster, “Where am i: Recognizing on-body positions of wearable sensors,” in *International Symposium on Location-and Context-Awareness*. Springer, 2005, pp. 264–275.

[27] J. Lester, B. Hannaford, and G. Borriello, “"are you with me?"–using accelerometers to determine if two devices are carried by the same person,” in *International Conference on Pervasive Computing*. Springer, 2004, pp. 33–50.

[28] T. M. Phuong, Z. Lin, and R. B. Altman, “Choosing SNPs using feature selection,” in *2005 IEEE Computational Systems Bioinformatics Conference (CSB’05)*, Aug 2005, pp. 301–309.

[29] M. A. Hall, “Correlation-based feature selection for machine learning,” Ph.D. dissertation, Department of Computer Science, The University of Waikato, Hamilton, New Zealand, Apr 1999.

[30] B. Paskaleva, M. Hayat, Z. Wang, J. Tyo, and S. Krishna, “Canonical correlation feature selection for sensors with overlapping bands: Theory and application,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 10, pp. 3346–3358, Oct 2008.

[31] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug 2005.

[32] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” in *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, Aug 2003, pp. 523–528.

[33] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy k-nearest neighbor algorithm,” *IEEE transactions on systems, man, and cybernetics*, no. 4, pp. 580–585, 1985.

[34] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, “Translated learning: Transfer learning across different feature spaces,” in *Advances in neural information processing systems*, 2009, pp. 353–360.

© University of Pretoria

[35] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Journal of Machine Learning Research*, vol. 10, no. Feb, pp. 207–244, 2009.

[36] S. Kulkarni, G. Lugosi, and S. Venkatesh, “Learning pattern classification - a survey,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2178–2206, Oct 1998.

[37] G. H. John and P. Langley, “Estimating continuous distributions in bayesian classifiers,” in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 338–345.

[38] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[39] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[40] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[41] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, 1996.

[42] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, no. 6789, p. 947, 2000.

[43] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

[44] P. Verhulst, *Recherches mathématiques sur la loi d’accroissement de la population*. M. Hayez, 1845. [Online]. Available: https://books.google.co.za/books?id=JZTHXwAACAAJ

© University of Pretoria

[45] A. C. Mathias and P. C. Rech, "Hopfield neural network: the hyperbolic tangent and the piecewise-linear activation functions," *Neural Networks*, vol. 34, pp. 42–45, 2012.

[46] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[47] F. Akdeniz and H. Erol, "Mean squared error matrix comparisons of some biased estimators in linear regression," *Communications in Statistics-Theory and Methods*, vol. 32, no. 12, pp. 2389–2413, 2003.

[48] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[49] Y. Lin and M. Yuan, "Convergence rates of compactly supported radial basis function regularization," *Statistica Sinica*, pp. 425–439, 2006.

[50] A. Minetti, L. Ardigo, and F. Saibene, "The transition between walking and running in humans: metabolic and mechanical aspects at different gradients," *Acta physiologica scandinavica*, vol. 150, no. 3, pp. 315–323, 1994.

[51] P. Morasso, "Spatial control of arm movements," *Experimental brain research*, vol. 42, no. 2, pp. 223–227, 1981.

[52] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien, "Animating human athletics," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 71–78.

[53] K. E. Zelik, K. Z. Takahashi, and G. S. Sawicki, "Six degree-of-freedom analysis of hip, knee, ankle and foot provides updated understanding of biomechanical work during human walking," *Journal of Experimental Biology*, vol. 218, no. 6, pp. 876–886, 2015.

© University of Pretoria

[54] Z. Li, K. Hauser, J. R. Roldan, D. Milutinovic, and J. Rosen, “A novel method for quantifying arm motion similarity,” in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 5716–5719.

[55] A. Franklin, *Shifting standards: Experiments in particle physics in the twentieth century*. University of Pittsburgh Press, 2014.

© University of Pretoria

ADDENDUM A CONFUSION MATRICES

Table A.1. Confusion matrix for the 1-NN classifier and mRMR-MIQ dataset of Model 1.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.839	0.041	0.039	0.027	0.014	0.008	0.014	0.016	0.004
Lunge L	0.042	0.830	0.027	0.035	0.022	0.020	0.015	0.005	0.004
Squat	0.034	0.017	0.842	0.055	0.019	0.011	0.008	0.015	0.000
Hinge	0.055	0.026	0.061	0.804	0.015	0.007	0.007	0.024	0.000
Walk	0.012	0.008	0.001	0.005	0.918	0.028	0.022	0.006	0.000
Jog	0.015	0.007	0.006	0.015	0.077	0.794	0.085	0.000	0.000
Run	0.013	0.016	0.016	0.003	0.068	0.111	0.769	0.003	0.000
Stand	0.005	0.002	0.003	0.003	0.001	0.001	0.001	0.984	0.000
Lie	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.999

Table A.2. Confusion matrix for the 1-NN classifier and CFS dataset of Model 2.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.839	0.032	0.036	0.013	0.023	0.008	0.019	0.032	0.000
Lunge L	0.045	0.834	0.023	0.014	0.020	0.020	0.016	0.029	0.000
Squat	0.024	0.009	0.842	0.059	0.009	0.018	0.009	0.029	0.000
Hinge	0.009	0.009	0.035	0.894	0.006	0.013	0.004	0.030	0.000
Walk	0.009	0.014	0.005	0.005	0.917	0.021	0.022	0.006	0.000
Jog	0.008	0.020	0.020	0.011	0.048	0.835	0.050	0.011	0.000
Run	0.005	0.020	0.011	0.012	0.063	0.086	0.801	0.003	0.000
Stand	0.003	0.003	0.006	0.004	0.000	0.007	0.003	0.975	0.000
Lie	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.999

© University of Pretoria

Table A.3. Confusion matrix for the 1-NN classifier and CFS dataset of Model 3 located at the right upper arm.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.936	0.043	0.021	0.000	0.000	0.000	0.000	0.000	0.000
Lunge L	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Squat	0.000	0.016	0.951	0.016	0.000	0.016	0.000	0.000	0.000
Hinge	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000
Walk	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Jog	0.000	0.000	0.000	0.000	0.000	0.988	0.012	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.000	0.025	0.975	0.000	0.000
Stand	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table A.4. Confusion matrix for the 1-NN classifier and mRMR-MID dataset of Model 3 located at the left upper arm.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.936	0.026	0.000	0.038	0.000	0.000	0.000	0.000	0.000
Lunge L	0.014	0.971	0.000	0.014	0.000	0.000	0.000	0.000	0.000
Squat	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
Hinge	0.000	0.000	0.000	0.967	0.033	0.000	0.000	0.000	0.000
Walk	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Jog	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.011	0.000	0.989	0.000	0.000
Stand	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

© University of Pretoria

Table A.5. Confusion matrix for the 1-NN classifier and CFS dataset of Model 3 located at the right wrist.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.925	0.000	0.000	0.000	0.000	0.000	0.015	0.060	0.000
Lunge L	0.000	0.977	0.023	0.000	0.000	0.000	0.000	0.000	0.000
Squat	0.018	0.000	0.982	0.000	0.000	0.000	0.000	0.000	0.000
Hinge	0.021	0.000	0.021	0.938	0.000	0.000	0.000	0.021	0.000
Walk	0.000	0.000	0.000	0.019	0.962	0.000	0.000	0.019	0.000
Jog	0.000	0.000	0.000	0.000	0.047	0.907	0.047	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.000	0.000	0.983	0.017	0.000
Stand	0.017	0.000	0.000	0.009	0.000	0.000	0.000	0.974	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table A.6. Confusion matrix for the 1-NN classifier and CFS dataset of Model 3 located at the left wrist.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.938	0.021	0.021	0.000	0.021	0.000	0.000	0.000	0.000
Lunge L	0.000	0.925	0.025	0.000	0.000	0.000	0.050	0.000	0.000
Squat	0.000	0.021	0.872	0.000	0.043	0.043	0.000	0.021	0.000
Hinge	0.000	0.000	0.000	0.979	0.000	0.000	0.021	0.000	0.000
Walk	0.010	0.000	0.000	0.000	0.981	0.010	0.000	0.000	0.000
Jog	0.000	0.000	0.000	0.000	0.000	0.879	0.121	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.000	0.026	0.974	0.000	0.000
Stand	0.000	0.000	0.018	0.000	0.000	0.000	0.018	0.965	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

© University of Pretoria

Table A.7. Confusion matrix for the 1-NN classifier and CFS dataset of Model 3 located at the right thigh.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.867	0.000	0.033	0.067	0.000	0.033	0.000	0.000	0.000
Lunge L	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Squat	0.000	0.000	0.971	0.029	0.000	0.000	0.000	0.000	0.000
Hinge	0.044	0.000	0.000	0.844	0.000	0.089	0.000	0.022	0.000
Walk	0.000	0.000	0.000	0.000	0.977	0.000	0.023	0.000	0.000
Jog	0.024	0.024	0.000	0.000	0.000	0.951	0.000	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.015	0.000	0.985	0.000	0.000
Stand	0.009	0.000	0.009	0.000	0.000	0.009	0.000	0.973	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table A.8. Confusion matrix for the 1-NN classifier and MaxRel-MID dataset of Model 3 located at the left thigh.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Lunge L	0.046	0.892	0.046	0.015	0.000	0.000	0.000	0.000	0.000
Squat	0.000	0.046	0.923	0.031	0.000	0.000	0.000	0.000	0.000
Hinge	0.000	0.026	0.102	0.872	0.000	0.000	0.000	0.000	0.000
Walk	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
Jog	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000
Run	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000
Stand	0.000	0.020	0.000	0.039	0.000	0.000	0.000	0.941	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

© University of Pretoria

Table A.9. Confusion matrix for the 1-NN classifier and MaxRel dataset of Model 3 located at the right ankle.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.813	0.009	0.065	0.028	0.047	0.009	0.019	0.009	0.000
Lunge L	0.050	0.861	0.040	0.050	0.000	0.000	0.000	0.000	0.000
Squat	0.067	0.086	0.790	0.038	0.000	0.000	0.010	0.010	0.000
Hinge	0.000	0.022	0.144	0.833	0.000	0.000	0.000	0.000	0.000
Walk	0.055	0.000	0.000	0.000	0.898	0.015	0.033	0.000	0.000
Jog	0.028	0.023	0.014	0.005	0.014	0.904	0.009	0.005	0.000
Run	0.025	0.008	0.025	0.033	0.165	0.000	0.736	0.008	0.000
Stand	0.000	0.000	0.004	0.011	0.000	0.000	0.000	0.986	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

Table A.10. Confusion Matrix for the 1-NN classifier and CFS dataset of Model 3 located at the left ankle.

	Lunge R	Lunge L	Squat	Hinge	Walk	Jog	Run	Stand	Lie
Lunge R	0.827	0.041	0.010	0.031	0.051	0.020	0.010	0.010	0.000
Lunge L	0.035	0.798	0.018	0.035	0.070	0.018	0.026	0.000	0.000
Squat	0.039	0.019	0.845	0.078	0.010	0.000	0.000	0.010	0.000
Hinge	0.009	0.018	0.044	0.904	0.000	0.009	0.009	0.009	0.000
Walk	0.008	0.020	0.004	0.000	0.932	0.012	0.012	0.012	0.000
Jog	0.006	0.024	0.018	0.000	0.018	0.807	0.127	0.000	0.000
Run	0.000	0.035	0.021	0.007	0.049	0.141	0.746	0.000	0.000
Stand	0.000	0.000	0.008	0.003	0.003	0.000	0.000	0.986	0.000
Lie	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000

© University of Pretoria

Table A.11. Confusion matrix for the 1-NN classifier and mRMR-MID dataset for the localisation of Model 4.

	LUA	RUA	LW	RW	LT	RT	LA	RA
LUA	0.996	0.004	0.000	0.000	0.000	0.000	0.000	0.000
RUA	0.007	0.990	0.001	0.000	0.000	0.000	0.000	0.001
LW	0.000	0.000	0.997	0.000	0.000	0.000	0.002	0.002
RW	0.000	0.000	0.000	0.998	0.000	0.000	0.000	0.002
LT	0.000	0.000	0.004	0.000	0.983	0.000	0.002	0.011
RT	0.000	0.000	0.002	0.000	0.005	0.979	0.000	0.014
LA	0.000	0.000	0.000	0.000	0.001	0.001	0.992	0.006
RA	0.000	0.001	0.002	0.000	0.004	0.007	0.007	0.979

© University of Pretoria