THE UNIVERSITY
of EDINBURGH

Detection of Unusual Fish Trajectories from Underwater Videos

Çigdem Beyan

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
2014

Abstract

Fish behaviour analysis is a fundamental research area in marine ecology as it is helpful for detecting environmental changes by observing unusual fish patterns or new fish behaviours. The traditional way of analysing fish behaviour is by visual inspection using human observers, which is very time consuming and also limits the amount of data that can be processed. Therefore, there is a need for automatic algorithms to identify fish behaviours by using computer vision and machine learning techniques. The aim of this thesis is to help marine biologists with their work. We focus on behaviour understanding and analysis of detected and tracked fish with unusual behaviour detection approaches. Normal fish trajectories exhibit frequently observed behaviours while unusual trajectories are outliers or rare trajectories.

This thesis proposes 3 approaches to detecting unusual trajectories: *i)* a filtering mechanism for normal fish trajectories, *ii)* an unusual fish trajectory classification method using clustered and labelled data and *iii)* an unusual fish trajectory classification approach using a clustering based hierarchical decomposition.

The rule based trajectory filtering mechanism is proposed to remove normal fish trajectories which potentially helps to increase the accuracy of the unusual fish behaviour detection system. The aim is to reject normal fish trajectories as much as possible while not rejecting unusual fish trajectories. The results show that this method successfully filters out normal trajectories with a low false negative rate. This method is useful to assist building a ground truth data set from a very large fish trajectory repository, especially when the amount of normal fish trajectories greatly dominates the unusual fish trajectories. Moreover, it successfully distinguishes true fish trajectories from false fish trajectories which result from errors by the fish detection and tracking algorithms.

A key contribution of this thesis is the proposed flat classifier, which uses an outlier detection method based on cluster cardinalities and a distance function to detect unusual fish trajectories. Clustered and labelled data are used to select feature sets which perform best on a training set. To describe fish trajectories 10 groups of trajectory descriptions are proposed which were not previously used for fish behaviour analysis. The proposed flat classifier improved the performance of unusual fish detection compared to the filtering approach.

The performance of the flat classifier is further improved by integrating it into a hierarchical decomposition. This hierarchical decomposition method selects more specific features for different trajectory clusters which is useful considering the trajectory

variety. Significantly improved results were obtained using this hierarchical decomposition in comparison to the flat classifier. This hierarchical framework is also applied to classification of more general imbalanced data sets which is a key current topic in machine learning. The experiments showed that the proposed hierarchical decomposition method is significantly better than the state of art classification methods, other outlier detection methods and unusual trajectory detection methods. Furthermore, it is successful at classifying imbalanced data sets even though the majority and minority classes contain varieties, and classes overlap which is frequently seen in real-world applications.

Finally, we explored the benefits of active learning in the context of the hierarchical decomposition method, where active learning query strategies choose the most informative training data. A substantial performance gain is possible by using less labelled training data compared to learning from larger labelled data sets. Additionally, active learning with feature selection is investigated. The results show that feature selection has a positive effect on the performance of active learning. However, we show that random selection can be as effective as popular active learning query strategies in combination with active learning and feature selection, especially for imbalanced set classification.

Lay Summary

Fish behaviour analysis is helpful for detecting environmental changes by observing unusual fish motions or new fish behaviours. The traditional way of analysing fish behaviour is by visual inspection using human observers. However, this is very time consuming and also limits the amount of data that can be processed. Fish behaviours can be identified by using computer vision and machine learning techniques which results in automatic analysis. The aim of this thesis is to help marine biologists with their work. We focus on detection of unusual fish trajectories from natural underwater videos. Unusual trajectories are rare trajectories while normal fish trajectories are frequently observed behaviours.

Firstly, the fish trajectories were defined in terms of primitive motions which consider the orientation of the fish. Using these primitive motions, we proposed a method which aims to reject as many normal fish trajectories as possible while not rejecting unusual fish trajectories. This method was used to obtain a data set which was used to test the performance of other proposed methods from a very large fish trajectory repository. Secondly, an unusual fish trajectory detection method was proposed. Fish trajectories were described by using 10 groups of descriptions such as location, velocity based features, etc. The features that performed the best were used to group the trajectories. Then, unusual trajectories were detected using a method based on the size of the groups and the distance between trajectories in a group. After selecting the best features for different trajectory groups, significantly improved results were obtained compared to previously proposed methods. This method was also applied to data sets which have two types of data where one type has much less data compared to other type. In such a case, a problem usually occurs because traditional methods tend to be biased towards the more common data. The results showed that the proposed approach's performance is better than the state of art methods. Finally, this method was integrated with a methodology called active learning which tries to maximise the performance while decreasing the amount of data used. By combining active learning with just the best performing features we showed how to obtain better results with less training.

Acknowledgements

I am deeply grateful to my supervisor Bob Fisher, who has provided inspiring supervision, continues guidance, unconditional support and encouragement throughout this research. I have greatly appreciated his wisdom. This thesis would not be possible without his trust. I would like to thank Vittorio Ferrari and Amos Storkey for their insightful feedback and suggestions. I am also grateful to my examiners Victor Lavrenko and Graeme Jones for reviewing this work. Their comments and suggestions helped me a lot to improve this thesis.

I have greatly appreciated my colleagues in the Vision Lab and Fish4Knowledge Project and would especially like to convey my special thanks to Bas Boom, Steven McDonagh and Xuan Huang for their valuable advice on my research.

I gratefully acknowledge the funding for this study provided by the Principal of University of Edinburgh and School of Informatics.

I would like to thank to my friends in Turkey; especially Oya, Öykü, Burçak and Vaizoğlu ailesi for their guidance, support and motivation throughout this thesis. I also would like to thank Alptekin Temizel who encouraged me to apply to the PhD study in University of Edinburgh.

I have been very lucky to have great friends in Edinburgh. I am thankful to their trust and support. I have had a fantastic time in this inspiring capital.

Last but not least, I would like to thank my family. Without their endless support, patience, love and encouragement this thesis simply would not have been possible. I am proud to be your daughter and grandchild. This thesis is dedicated to them.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Çigdem Beyan*)

To my family: Esin, Cengiz, Süreyya and Sadık

.

Table of Contents

1 Introduction **1**

1.1 Underwater Video Surveillance Approaches and the Fish4Knowledge Project 1

1.2 Problem Description 2

1.2.1 Definitions 3

1.3 Challenges 4

1.4 Thesis Statement and Claims 5

1.5 Organisation of the Thesis 6

1.6 Original Contributions 9

2 Literature Review **11**

2.1 Fish Behaviour Understanding 11

2.2 Related Works on Unusual Trajectory Detection 14

2.2.1 Trajectory Representation Methods for Unusual Trajectory Detection 15

2.2.2 Learning Methods for Unusual Trajectory Detection 17

2.3 Related Works on Imbalanced Data Classification 22

2.3.1 Evaluation Metrics for Imbalanced Data Classification 25

2.4 Related Works on Hierarchical Classifiers and Hierarchical Decomposition 27

2.5 Related Works on Active Learning 29

2.6 Summary 32

3 A Filtering Mechanism for Normal Fish Trajectories **34**

3.1 Methodology 35

3.1.1 Trajectory Description 35

3.1.2 Proposed Filtering Mechanism 35

3.1.3 Definition of Filters . . . 35
3.2 Data Set . . . 42
3.3 Experimental Work . . . 42
3.4 Results . . . 43
3.4.1 Comparison with a State of Art Method . . . 45
3.5 Conclusions . . . 46

4 Detecting Unusual Fish Trajectories Using Clustered and Labelled Data (Flat Classifier) 47
4.1 Methodology . . . 48
4.1.1 Feature Extraction . . . 49
4.1.2 Clustering . . . 56
4.1.3 Outlier Detection . . . 58
4.1.4 Feature Selection . . . 58
4.2 Data Set . . . 60
4.3 Results . . . 60
4.4 Conclusions . . . 63

5 Detection of Unusual Fish Trajectories Using a Clustering Based Hierarchical Decomposition 66
5.1 Methodology . . . 67
5.1.1 Hierarchy Decomposition . . . 67
5.1.2 New Trajectory Classification Using the Constructed Hierarchy 68
5.2 Experimental Work . . . 74
5.2.1 Data Sets . . . 76
5.2.2 Results . . . 78
5.3 Conclusions . . . 86

6 Classifying Imbalanced Data Sets Using Similarity Based Hierarchical Decomposition 88
6.1 Experimental Works and Results . . . 89
6.1.1 Experiments and Results using Public Imbalanced Data Sets . 90
6.1.2 Experiments and Results with Synthetic Data Sets . . . 104
6.2 Conclusions . . . 110

7 Active Learning with Imbalanced Data Sets **111**
7.1 Active Learning 112
7.2 Active Learning with Feature Selection 114
7.2.1 Methodology 115
7.2.2 Experimental Work 116
7.2.3 Results when Naive Bayes is used as the classifier 118
7.2.4 Results when a Support Vector Machine is used as the classifier 135
7.2.5 Summary and Discussion for Active Learning with Feature Selection 146
7.3 Hierarchical Decomposition Method Integrated with Active Learning 150
7.3.1 Proposed Setting 150
7.3.2 Data Sets and Experimental Design 152
7.3.3 Results 153
7.3.4 Further Analysis 159
7.3.5 Summary and Discussion for Hierarchical Decomposition Method integrated with Active Learning 162
7.4 Conclusions 163

8 Conclusions **164**
8.1 Main Contributions, Limitations and Future Work 165
8.1.1 A Filtering Mechanism for Normal Fish Trajectories 165
8.1.2 The Flat Classifier 166
8.1.3 Clustering Based Hierarchical Decomposition 167
8.1.4 Active Learning with Imbalanced Data Sets 168

Bibliography **171**

List of Figures

3.1 The block diagram of the rule based normal fish trajectory filtering method 36

3.2 The descriptions of 21 rules. 1 represent the first fish detection, a is the ath fish detection (the end of first segment that is also the beginning of the second trajectory segment), b refers to bth fish detection (the end of second segment that is also the beginning of the third trajectory segment), N is the last fish detection in the whole trajectory and p is the parameter that is used to define the search area for straight and/or cross motions and being stationary (see text for definition). 37

3.3 Example illustrations for straight and/or cross movements: a) Type 1 for left to right, b) Type 2 for right to left, c) Type 3 for up to down and d) being stationary. From the first to the last detection the bounding boxes are shown with black, green, pink and orange, respectively. The shaded areas are the search areas which is rectangular for Type 1 (shown as black shaded), Type 2 (shown with the same colour of shading with the corresponding bounding box), Type 3 (shown as black shaded) and circular for being stationary (shown as black shaded). . . 40

3.4 Illustration of the search area (red shaded area) for each type of the straight and/or cross movements. p is a parameter that defines the search area. a) For Type 1 (x_{f_a}, y_{f_a}) is the centre of the first detection and for Type 2 it is the centre of every detection from $i = 2$ to b where b is the end of that trajectory segment. b) For Type 3, (x_{f_a}, y_{f_a}) is the centre of the first detection (bounding box is shown with blue) and (x_{f_b}, y_{f_b}) is the centre of the last detection (bounding box is shown with green) of a given trajectory segment. 41

3.5 (a-b) Examples of normal fish trajectories which are classified by the proposed method, (c-d) Examples of unusual fish trajectories. 43

3.6 Example unusual (red) and normal (blue) trajectories for 4 different camera locations 44

4.1 Overview of the flat classifier 48
4.2 Example trajectories (left) and corresponding CSS images (right) . . . 50
4.3 Properties of the vicinity (adapted from [1]) 52
4.4 A trajectory with a loop. 54
4.5 Example fish trajectories with loops. 54
4.6 Segmented regions of the underwater image; black for open sea, red for above the coral and green for under coral 55
4.7 A representation of clustered data. For small clusters, boundaries are shown with thick lines and dense clusters' boundaries are shown with dashed lines. Outlier detection in dense clusters: samples which are inside of the inner circle are classified as the normal trajectories whereas the rest of the samples are classified as the unusual trajectories, given threshold τ. 59
4.8 Sequential Forward Feature Selection (adapted from [2]) 59
4.9 Mean of $TPrate$ and $TNrate$ after the feature is added at each iteration of $SFFS$ (Training). Best feature selection criterion value is emphasised as bold. 61
4.10 Examples of misclassified unusual trajectories (top) and misclassified normal trajectories (bottom). Trajectories are shown with blue while the last detections of the fish are shown with a red bounding box. . . . 64

5.1 Hierarchy Construction 69
5.2 The pseudo-code for hierarchy construction 70
5.3 Cluster types: perfectly classified clusters, misclassified clusters. A perfectly classified cluster can be *a)* perfectly classified mixed, *b)* perfectly classified pure normal, *c)* perfectly classified pure unusual. A misclassified cluster can be *d)* misclassified mixed, *e)* misclassified pure normal when the cluster is a dense cluster, *f)* misclassified pure normal when the cluster is a small cluster, *g)* misclassified pure unusual. Diamonds represent the unusual trajectories while circles represent the normal trajectories. The outlier detection thresholds for dense clusters are shown with dashed red circles. 71

5.4 New trajectory classification using the hierarchy. 73
5.5 New trajectory classification when the decisions of all levels are **no effect on decision**. 74
5.6 The flow chart of classification of a new trajectory using a previously (during training) constructed hierarchy. Decisions are all shown with rounded rectangles either with single or double line. Rounded rectangles with double lines represent the final class of the new trajectory whereas single line rounded rectangles indicate provisional decisions. 75
5.7 (a-b) Normal fish trajectory examples, (c-d) Unusual fish trajectory examples. 76
5.8 Data set which belongs to 1st of September 2009 in the Forum Pedestrian database [3]. Examples of normal (blue) and abnormal trajectories (red). 77

6.1 Comparing methods with the proposed method using Wilcoxon's Signed rank test using the *GeoMean*, *AGeoMean* and *AUC*. R^+ (Eq. 6.1) represents the rank of the proposed method and R^- (Eq. 6.2) represents the rank of the compared method while α is taken as 0.05. Significance (Sig.) is shown as "yes" if there is a significant difference, otherwise it is shown as "no". 99
6.2 The average ranks used in the computation of the Friedman test for the metrics *GeoMean*, *AGeoMean* and *AUC* respectively. Lower rank means better performance. The best performance is shown in bold. . . 100
6.3 Holm test results for the comparison between proposed method and the other methods using the *a)* GeoMean, *b)* AGeoMean *c)* AUC results. 101
6.4 Examples of train-test pairs when the number of features is 2. Samples belong to majority class is shown with blue while samples belong to minority class is shown with red. Data sets having *a)* $\alpha_{minority}$=1 and $\alpha_{majority}$=1, *b)* $\alpha_{minority}$=4 and $\alpha_{majority}$=128, *c)* $\alpha_{minority}$=256 and $\alpha_{majority}$=8 and *d)* $\alpha_{minority}$=512 and $\alpha_{majority}$=4 for the same set of class centres. The class centres are varied on each cross validation fold. 105

6.5 The best results of methods in terms of the average of *GeoMean* for 16 different data sets created by different values of α for the minority and majority classes. The error bars show the standard deviations of the performance considering the 30 data folds for each data set. The number of features is *a)* 2, *b)* 5, *c)* 10, and the number of samples for the majority class is 300 while the number of samples in the minority class is 50. The data sets given on the horizontal axis refer to the index number of data set given in the legend which uses different combinations of $\alpha_{minority}$ and $\alpha_{majority}$ and are sorted by the performance of *GMM_OC* by decreasing order. 107

6.6 Summary of the performance in terms of the *GeoMean* for different feature sets having 100 different data sets for each, grouped by $\alpha_{minority}$ and $\alpha_{majority}$ ratios. Orange columns show when the proposed method performed worse, green columns show when the proposed method performed better, pink columns show when the proposed method performed about equal (see text for more detail). 108

6.7 Best classification performance of methods in terms the average *GeoMean* using data sets with different imbalance ratios ($N_{minority}/N_{majority}$) when $\alpha_{minority}$ is $\{256, 256, 512\}$ and $\alpha_{majority}$ is $\{8, 4, 4\}$ which makes the ratios equal to 32, 64 and 128 respectively. Number of features is *a)* 2, *b)* 5, *c)* 10. ALL means that all classifiers had essentially equal performance. 109

7.1 Pool Based Active Learning 113

7.2 Active Learning with Feature Selection 116

7.3 Results in terms of the *GeoMean* for the Pima data set [4, 5] using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density. 119

7.4 Results in terms of the *GeoMean* for the Oil data set [6] using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density. 120

7.5 Results in terms of the *GeoMean* for the Satimage data set [7] using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density. 121
7.6 Results in terms of the *GeoMean* for the Forum Pedestrian Database using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density. 122
7.7 Results in terms of the *GeoMean* for the fish trajectory data set using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density. 123
7.8 Results in terms of the *GeoMean* for the Satimage_2 data set [7] using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, black for random selection and green for information density. 124
7.9 Results in terms of the *GeoMean* for the Yeast data set [7] using *NB*. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, black for random selection and green for information density. 125
7.10 The mean plus error bars for the Pima [4, 5] data set using random selection. With feature selection (top) and without feature selection (bottom). 132
7.11 The mean plus error bars for the Oil [6] data set using random selection. With feature selection (top) and without feature selection (bottom). 132
7.12 The mean plus error bars for the Satimage [7] data set using random selection. With feature selection (top) and without feature selection (bottom). 133
7.13 The mean plus error bars for the Forum Pedestrian Database data set using random selection. With feature selection (top) and without feature selection (bottom). 133
7.14 The mean plus error bars for the fish trajectory data set using random selection. With feature selection (top) and without feature selection (bottom). 134

7.15 The mean plus error bars for the Satimage_2 [7] data set using random selection. With feature selection (top) and without feature selection (bottom). 134
7.16 The mean plus error bars for the Yeast [7] data set using random selection. With feature selection (top) and without feature selection (bottom). 135
7.17 Results in terms of the *GeoMean* for the Pima data set [4, 5] using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 136
7.18 Results in terms of the *GeoMean* for the Oil data set [6] using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 137
7.19 Results in terms of the *GeoMean* for the Satimage data set [7] using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 138
7.20 Results in terms of the *GeoMean* for the Forum Pedestrian Database using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 139
7.21 Results in terms of the *GeoMean* for the fish trajectory data set using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 140
7.22 Results in terms of the *GeoMean* for the Satimage_2 data set [7] using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 141
7.23 Results in terms of the *GeoMean* for the Yeast data set [7] using *SVM*. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density. 142
7.24 Hierarchical Decomposition Method Integrated with Active Learning 152

7.25 Active learning with hierarchical decomposition results in terms of the *GeoMean* for the Pima data set [4, 5]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail). 154
7.26 Active learning with hierarchical decomposition results in terms of the *GeoMean* for the Oil data set [6]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan and magenta lines show the 95% of performance that was obtained when all training data is used and best performance, respectively (see text for more detail). 154
7.27 Active learning with hierarchical decomposition results in terms of the *GeoMean* for the Satimage data set [7]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail). 155
7.28 Active learning with hierarchical decomposition results in terms of the *GeoMean* for the Forum pedestrian data set. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail). . . 155
7.29 Active learning with hierarchical decomposition results in terms of the *GeoMean* for the fish trajectory data set. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail). . . 156

List of Tables

2.1 Comparison of the most popular trajectory representation methods for unusual trajectory detection with example references. 17
2.2 Most popular learning methods for unusual trajectory detection 22
2.3 Confusion matrix for a two-class problem 25

3.1 Performance of the rule based normal fish trajectory filtering method . 45

4.1 The number of extracted features before and after *PCA* for the fish trajectory data set given in Section 4.2. 56
4.2 Methods that are used for comparison. 62
4.3 Best results of each method in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$. The best results are emphasised in bold-face. The standard deviations considering the cross-validation folds are also given after the $\pm$ sign. 63

5.1 Methods that are used for comparison. 79
5.2 Best results of each method in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$ when fish trajectory data set is used. $Approximated$ $(Approx.)$ $TPrate$ is calculated for fixed $TNrate = 0.88$. The standard deviations (considering cross validation folds) of the methods are also given after $\pm$ sign. The best results are emphasised in bold-face. 80
5.3 Definition of alternative hierarchical decomposition methods. 82

5.4 Best results for the alternative hierarchy decomposition methods in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$ when the fish trajectory data set is used. $Approximated$ $(Approx.)$ $TPrate$ is calculated for fixed $TNrate = 0.88$. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face. 83
5.5 Applied methods using pre-processed (outlier removed and normalised) fish trajectory features. 84
5.6 Best results of the methods in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$ for the fish trajectory data set with and without pre-processed features. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face. 84
5.7 Methods that are used for comparison when using the pedestrian data set [3]. 85
5.8 Best results of each method in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$ when the pedestrian trajectory data set is used. $Approximated$ $(Approx.)$ $TPrate$ is calculated for fixed $TNrate = 0.87$. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face. 86

6.1 Pre-processing algorithms that are used. 89
6.2 Summary of used imbalanced data sets. 91
6.3 State or art methods and their combinations with pre-processing algorithms that are used for comparison. 93
6.4 Best results of each method in terms of the average *GeoMean*. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign. 94
6.5 Best results of each method in terms of the average *AGeoMean*. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign. 95

6.6 Best results of each method in terms of the average AUC. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign. 96
6.7 Paired t-test results between each method and the proposed method using the *GeoMean* results. 103

7.1 Used Balanced and Imbalanced Data Sets 117
7.2 The number of samples selected at each iteration of active learning for different data sets and classifiers. 118
7.3 Paired t-test of the *AL* query strategies and random selection with/without feature selection when *NB* is the classifier (see text for more detail) . . 127
7.4 Number and variety of selected features for *AL* and random selection with feature selection 129
7.5 The comparison in computation time (average and standard deviation (after the $\pm$ sign) over different cross validation folds) between *AL* with/without feature selection (using Random Selection) 130
7.6 Paired t-test of the query strategies and random selection when feature selection is integrated and *NB* is used as the classifier (see text for more detail) 131
7.7 Paired t-test of the information density and random selection with/without feature selection when SVM is used as the classifier 145
7.8 Paired t-test of information density and random selection when feature selection is integrated and SVM is used as the classifier 147
7.9 The minimum percentage of training data that is needed to obtain substantial performance and the percentage of training data that the best performance is reached with the corresponding *AL* strategy and random selection. *Random* is for random selection and $InfoDen$ is for information density. 157
7.10 Paired t-test results of the query strategies and random selection for hierarchical decomposition integrated with *AL* (see text for more detail) 158
7.11 Number of selected features for hierarchical decomposition integrated with *AL* and random selection 160
7.12 Number of hierarchy levels for hierarchical decomposition integrated with *AL* 161

Chapter 1

Introduction

The study of marine ecosystems such as observing coral reefs is important for understanding environmental effects caused by global warming, pollution, etc. Analysing fish behaviour is one approach to detecting environmental changes. This analysis may consist of detecting changes in behaviour pattern of fish or by finding unusual behaviours. For instance, by analysing the behaviour of fish hovering over coral, the health of the coral can be determined. However, investigating underwater environments is very challenging since it needs long-term monitoring and automatic analysis, whereas the traditional approach requires manual processing, which is very labour intensive and time consuming.

Underwater video surveillance systems can help marine biologists monitor marine life while computer vision and pattern recognition techniques can help them to automatically analyse the output of these systems, which is a huge amount of (tera-scale) underwater videos. Using these large-scale data, higher level interpretations can be extracted by automatically detecting, tracking and recognising fish to collate knowledge. Marine biologists can benefit from the data to analyse species abundance and distributions, assess environmental changes, understand predator-prey relationships, etc.

1.1 Underwater Video Surveillance Approaches and the Fish4Knowledge Project

There are many approaches to observing fish behaviour in their natural environment. Net casting with acoustic sensors [8] is a popular method to observe fish and determine their abundance [9]. Diving to observe underwater using photography, hand-held

video devices and optical systems to investigate fish behaviour are popular as well. Additionally, acoustic systems, echo-systems and sonar have been used for fish monitoring [10]. The main disadvantage of these systems is disturbing fish as they are very sensitive to their environment which results in unusual fish behaviour. Moreover, it is hard to capture large amounts of data which makes a comprehensive analysis difficult e.g. long-term monitoring is also impossible and the captured data might not include substantial information [9].

In recent years, as digital video recording systems become cheaper, collecting data in natural underwater environments with a fixed camera set up which continuously records underwater videos has become possible. For instance, in the **Fish4Knowledge project** embedded video cameras were used to capture underwater videos at different locations of the Taiwanese Coral Reef such as Third Taiwanese Power Station [11]. The Fish4Knowledge project includes methods to capture, store, analyse and query underwater videos. The aim is to analyse very large amounts of long-term video using computer vision, pattern recognition, database management, semantic web and work flow technologies [12]. The computer vision part of the Fish4Knowledge system covers components for fish detection, fish tracking and fish species recognition. Using the results of these components, it is possible to analyse fish behaviour. All the fish trajectory data sets used in this thesis are from the Fish4Knowledge project repository. The fish detection and tracking [9] and fish recognition [13] components were utilised to obtain fish trajectories while all the trajectories were manually inspected to be sure that there are no false detections, false tracking or false recognitions.

1.2 Problem Description

In computer vision research, behaviour understanding studies have commonly presented research on human behaviour analysis, traffic surveillance, and nursing home surveillance. These approaches can be classified in two categories:

- Prominent activity recognition,
- Unusual event detection [14].

In the first category, the system has a definite knowledge of the activities and when activity is detected, it can be classified in terms of the known activity description [14]. However, this may cause a rapid growth in the number of behaviour models in the real-world use [14, 15]. On the other hand, for unusual event detection, the system usually

does not have any prior knowledge about the behaviours and data is usually analysed by clustering the behaviours to detect normal (usual) or rare behaviours or modelling possible normal behaviours.

The aim of our work is to present **an unusual fish trajectory detection system** that analyses natural underwater environment videos. The methods proposed in this thesis classify the trajectories as **normal** and **unusual**. *Normal fish trajectories are defined as the trajectories which contain frequently observed trajectories while unusual trajectories are defined as the trajectories that are rare or outliers.* By using the proposed methods, we want to help marine biologists with their work. For instance, by detecting rare behaviours, an unknown behaviour of a fish which might be due to an environmental change can be detected. Furthermore, the proposed methods can be seen as a preliminary work to understanding specific behaviours of fish species such as feeding, predator-prey, reproduction, etc. In this thesis, we analyse previously detected and tracked fish [9], hence fish detection and tracking are beyond the scope of this thesis.

1.2.1 Definitions

- **Trajectory:** The displacements of objects that are typically considered as positions in 2 dimensions over time.
- **Action:** Simple motion patterns which happen in a short time by a single object [16].
- **Activity:** Complex sequence of actions that last a longer time and may include more than one agent [16].
- **Event:** Occurrence of an activity in a specific place and time [17].
- **Behaviour:** Activities and events in a specific context.

In the literature, trajectory can be used interchangeably with the terms action, activity, event and behaviour. Similarly, in this thesis, even though we analyse trajectories, we sometimes use words behaviour or event to refer them.

Additionally, the definition of **unusual** is a bit ambiguous in the literature. The words: **abnormal, rare, outlier, suspicious, anomaly** can be used interchangeably with unusual (see Chapter 2 for examples). In this thesis, we use **unusual** meaning not

common, not frequently observed, rare and outlier (see Section 4.1.3). On the other hand, **normal** means frequently observed and common.

1.3 Challenges

The difficulties for fish trajectory analysis using underwater videos are mainly two-fold:

- The challenges which directly affect the detection of unusual fish trajectories: When we compare fish trajectories in underwater videos with the other unusual behaviour detection systems (for instance traffic surveillance, human unusual trajectory detection and home surveillance), there are certain differences:
  - Fish in the open sea can freely move in 3 dimensions, there are no defined rules or roads such as exist in a traffic surveillance scenario.
  - Fish are usually not goal-oriented which produces highly complex trajectories in contrast to people or vehicles.
  - Fish usually make erratic movements due to currents in the sea which increases the complexity of the trajectories and also makes the encoding of the behaviours more difficult than is in human or animal behaviour recognition [18]
  - The huge amount of data: The Fish4Knowledge repository has tera-scale video data. Due to the failures in fish detection, tracking and recognition components, it is hard to automatically obtain ground-truth data for fish trajectory analysis. A method which quickly scans huge amounts of data and filters out tracking failures was needed. The proposed method in Chapter 3 was used with this purpose as well.
- The challenges which indirectly affect the detection of unusual fish trajectories: As fish trajectory analysis depends on fish detection and tracking any difficulty that affect these components affects detection of unusual fish trajectories.
  - Complex background, foreground objects and low quality of video: Due to sudden light changes in underwater scenes, bad weather conditions (such as storms, typhoons), murky water and swaying plants, etc. [19].

- Multiple fish occlusions due to the third dimension in the scene while all the processed images are in 2 dimensions [19].

These difficulties make gaps and noise in the trajectories. Furthermore, collection of the ground-truth data for fish trajectory analysis also becomes more difficult due to the number of tracking failures which happen due to above challenges (based on the manual examination, we estimate that only 75% of the trajectories are fully correct). These resulted in a manual sanity check of the output of the fish tracking and recognition components to obtain the fish trajectory data sets.

1.4 Thesis Statement and Claims

The central goal of this research project can be stated as follows:

Using multiple features extracted from fish trajectories in underwater videos and methods based on clustering, feature selection and outlier detection, unusual fish trajectories can be detected (which is also an imbalanced data classification problem).

The underlying claims to realise this goal can be defined as follows:

1. *Given that there is a huge amount of fish trajectories (considering the Fish4Knowledge repository especially) and the number of normal fish trajectories is much bigger than the number of unusual trajectories, a rule based method could be used to extract normal trajectories while keeping unusual trajectories which should result in a less imbalanced data set.*

2. *Individual fish of the same species behave similarly (meaning that the spatio-temporal characteristics and the shape of the trajectories are similar) in the same underwater locations (such as open sea, above the coral, below the coral) at the same time period of the day (such as morning) otherwise their behaviour could be unusual. This problem could be solved with:*

    - *A clustering based method, where clustered and labelled data are used together to select best feature set and unusual trajectories are determined by using an outlier detection method.*

- *An automatically generated hierarchical decomposition method (based on clustering, feature selection and outlier detection) which allows selecting more specific features for different trajectory clusters.*

To verify the first claim, fish trajectories from different distances, species, locations and time of the day are involved. To verify the following two claims, fish trajectories belonging to a single fish species, the same camera fields of view (which varies slightly due to repositioning after typhoons or camera lens cleaning) and time of the day are considered. All used data sets involve sub-varieties for normal and unusual trajectories and are highly imbalanced.

The other claims of this research project can be stated as follows:

3. *Given that the classification with imbalanced data sets is an important problem in machine learning (since the real-world data sets are generally imbalanced), the proposed hierarchical decomposition method to detect unusual fish trajectories could be used more generally as a solution of this problem.*

4. *The proposed hierarchical decomposition method can be integrated with active learning. This can result in equivalent performance with less training data by using a proper query strategy which determines the most informative unlabelled training data samples and better feature subsets to build a hierarchy.*

To verify the third claim, different imbalanced data sets from different fields such as biology, physics, etc. and synthetic data sets were used. To classify the test samples different heuristics were applied and evaluated using the constructed hierarchy. Similarly, to support the last claim, different imbalanced data sets including fish trajectory and pedestrian trajectory sets were used.

1.5 Organisation of the Thesis

This thesis is mainly about unusual fish trajectory detection which is performed in a supervised way using fish previously detected and tracked from the underwater videos. One aim is to help marine biologists with their work. The proposed methods allow the biologist to focus on data that is potentially unusual which is valuable especially considering the amount of data that they have to analyse. Additionally, since the proposed

methods help to detect rare trajectories this might help marine biologist to detect more interesting behaviours, maybe even behaviour changes for a specific species.

The remainder of that document is structured as follows:

Chapter 2 gives a comprehensive overview and comparison of existing works in the area of fish behaviour understanding, unusual trajectory detection, imbalanced data classification, hierarchical methods and active learning.

Chapter 3 presents a rule based method for filtering normal fish trajectories. Normal fish trajectories were defined in terms of primitive motions where the aim is to filter out normal trajectories as much as possible while not filtering out any unusual fish trajectories. This novel approach is useful to quickly scan the large fish trajectory repository to determine normal and unusual trajectories which can also be used to build a ground-truth data set. Additionally, its unusual fish trajectory detection performance is better than many other algorithms. The work presented in this chapter has been published or accepted to be published as follows:

- Beyan C., Fisher R. B. (2012), A Filtering Mechanism for Normal Fish Trajectories, *In Proceedings of International Conference on Pattern Recognition (ICPR)*, pp. 2286-2289, Tsukuba Science City, Japan.
- Beyan C., Fish Behavior Analysis, In Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data, *Chen-Burger* et al. *(Editors), Springer*, in preparation.

Chapter 4 focuses on extracting novel multiple features from fish trajectories. The proposed method is based on clustering where clustered and labelled data are used to select the best features to detect unusual trajectories as outliers in the clusters. This novel method has improved performance compared to the method given in Chapter 3 and it is also the foundation for the method presented in Chapter 5. The work presented in this chapter has been published or accepted to be published as follows:

- Beyan C., Fisher R. B. (2013), Detecting Abnormal Fish Trajectories using Clustered and Labelled Data, *In Proceedings of International Conference on Image Processing (ICIP)*, pp. 1476-1480, Melbourne, Australia.

- Beyan C., Fish Behavior Analysis, In Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data, *Chen-Burger* et al. *(Editors), Springer*, in preparation.

Chapter 5 illustrates a novel hierarchical decomposition method for mainly fish trajectory detection. The hierarchy construction is based on clustering, outlier detection and feature selection. Different feature sets and different data samples are used at different levels of the hierarchy. The advantage of this method is to allow selecting more specific features once the data focuses onto specific subclasses. The proposed method was tested with pedestrian data set as well and different aspects of the proposed method such as the heuristics that it uses to classify test trajectories were examined. The work presented in this chapter has been published or accepted to be published as follows:

- Beyan C., Fisher R. B. (2013), Detection of Abnormal Fish Trajectories Using a Clustering Based Hierarchical Classifier, *In British Machine Vision Conference (BMVC)*, Bristol, UK.
- Beyan C., Fish Behavior Analysis, In Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data, *Chen-Burger* et al. *(Editors), Springer*, in preparation.
- Beyan C., and Fisher R. B., Hierarchical Decomposition for Unusual Fish Trajectory Detection, In Computer Vision and Pattern Recognition in Environmental Informatics, *Zhou* et al. *(Editors), IGI Global*, in preparation.

Chapter 6 addresses classification with imbalanced data sets. The method proposed in Chapter 5 is applied to various imbalanced data sets and also synthetic data sets. It is compared with the state of art classification methods for imbalanced data. The results and various statistical analysis show that the proposed method generally performs significantly better than the state of art methods. It performs well even when classes have sub-varieties, classes are overlapping and more imbalanced. The work presented in this chapter has been accepted to be published as follows:

- Beyan C., Fisher R. B., Classifying Imbalanced Data Sets Using Similarity Based Hierarchical Decomposition, *Pattern Recognition*, to appear.

Chapter 7 focuses on active learning with feature selection and integration of the proposed hierarchical decomposition method with active learning. Various active learning query strategies and random selection are compared with each other. It is observed that active learning with feature selection achieved better performance than without feature selection although random selection is generally as good as active learning query strategies. Additionally, we integrated the hierarchical decomposition method with active learning. The results show that with a proper active learning query strategy, the desired performance (such as the performance which is very close to the obtained performance with passive learning) could be obtained with less training data.

Chapter 8 discusses the results and the contributions of this thesis and outlines future works.

1.6 Original Contributions

The next chapters present the following original contributions:

1. *A novel normal fish trajectory filtering method:* This is the first algorithm for filtering normal fish trajectories in data from an unconstrained open sea environment.

2. *A novel unusual fish trajectory detection method based on clustering, outlier detection and feature selection:* Both labelled and clustered data are used together for classification of trajectories which makes this method novel. Additionally, novel trajectory descriptions which were not previously used for fish behaviour analysis were proposed. The best classification performance (until the following method) was obtained for unusual fish trajectory detection in natural underwater videos.

3. *A novel hierarchical decomposition method which uses similarity of the data to build the hierarchy:* This method is different from previously proposed solutions for unusual fish trajectory detection and the class imbalance problem. It does not require any data pre-processing step as many other imbalanced data classification approaches do. This hierarchical decomposition method is novel as the hierarchy is constructed using the similarity of labelled data subsets at each level of the hierarchy with different levels being built by different data and feature subsets. This is in contrast to previous research that uses the same feature set for

every level of the hierarchy or a flat classifier. This method is the most accurate approach to detect unusual fish trajectories in underwater videos.

4. Fish behaviour analysis in natural underwater scenes is very limited and the proposed methods are the only methods that aim to detect unusual fish trajectories using natural underwater videos.

5. Throughout this thesis, the largest fish trajectory data set which is the largest labelled trajectory data set as well was collected and used in our experiments.

6. *A comprehensive investigation for active learning with feature selection:* The literature is very limited in terms of active learning with feature selection. The existing studies about active learning with feature selection all belong to the natural language processing field where the feature space is implicitly changing and the features were manually determined. We investigated active learning with feature selection without aiming at any specific application area where the feature space is automatically determined by the feature selection algorithm. Many popular active learning query strategies and random selection were utilised for the analysis.

7. *A novel approach to integrate the proposed hierarchical decomposition method with active learning:* Using this approach, the best performance was obtained with less training data. This method is possibly the simplest method conceptually while some other approaches need a learning stage to estimate the probabilities from the distance between data samples.

Chapter 2

Literature Review

In this chapter, overviews of previous research about fish behaviour understanding (Section 2.1), unusual trajectory detection (Section 2.2), imbalanced data classification (Section 2.3), hierarchical classifiers and hierarchical decomposition (Section 2.4) and active learning (Section 2.5) are given.

Studies on fish behaviour analysis are investigated based on the application area. Unusual trajectory detection methods are summarised in terms of trajectory representations and the learning methods used. Studies about imbalanced data classification are presented by their methods and evaluation metrics. Additionally, hierarchical decomposition is distinguished from hierarchical classifiers with example studies from the literature. Lastly, active learning particularly for imbalanced data classification and active learning with feature selection, is addressed.

2.1 Fish Behaviour Understanding

Fish behaviour monitoring and understanding studies using computer vision and machine learning techniques are becoming popular not only in marine biology but also in artificial intelligence. However, the number of studies in this field is still limited compared to the number of approaches for fish detection, tracking and recognition. Studies for fish behaviour analysis can be categorised as:

- Studies considering fish as individuals or as a school,
- The number of fish or fish species that are examined,
- The video capturing environment (fish tank, aquarium, aquaculture sea cage, natural environment, etc.),

- The application area of the study (detecting unusual fish trajectories, water quality monitoring using fish behaviour, fish stress factor identification, quantification of fish behaviour, video classification using fish motion patterns and so forth).

In this section, we review the fish behaviour understanding methods in terms of their application area.

Existing studies generally focus on water quality monitoring and chemical contaminant detection based on the behavioural stress responses of fish [20, 21, 22, 23, 24, 25]. For instance, Thida *et al*. [20] used trajectory shape features with a signed-distance function. Using incremental spectral clustering, fish trajectories were grouped and the location of fish and the swimming directions were determined in the clean water. Those trajectories were used to determine the abnormal trajectories in the toxic water where a threshold defines the normality score and any trajectory having a score under that threshold is classified as abnormal. Similarly, recurrence plots were used to analyse the swimming pattern of fish in the presence of chemicals in the water [24]. It is assumed that the behaviour of a fish in polluted water should be changed over time compared to the same fish in the clean water. The fish trajectories were represented as no movement, up, down, right and left movement. A string representing each trajectory in terms of those movements was obtained for each trajectory. Strings were compared with Levenshtein and Hamming distances and used to build the recurrence plots to detect unusual swimming patterns. A real time automatic fish school behaviour monitoring system was presented by Chew *et al*. [25]. In that study, two equal sized tanks (one for clean water one for contaminated water) having 20 fish in each with the same environmental conditions (such as illumination) were used. Position and the size of the fish school were determined using a simple background subtraction algorithm. The activity level of the fish school was determined using the overall speed of fish and the complexity of the path of the school. Additionally, the school distribution in different parts of the tank was found and the distribution in the clean water and the contaminated water were compared. The results showed that the fish school in the contaminated tank tended to swim away from the central area of the tank. Other studies which considered different stress factors also exist and it is known that high stocking density is one of the major factors that causes fish stress [26]. For instance, Papadakis *et al*. [23] considered the stocking density and proposed a computer vision system to observe the behaviour variability of *Sparus aurata*. This species was monitored before and after feeding time during the day. The time that fish spent to inspect the net and the number of bites on the new surface were determined. The results in that study [23] showed that there is a

connection between fish behaviour, stocking density, and net condition. Fish feeding is influenced by stocking density and by the social interactions of fish.

A different application area in this field is automatically monitoring fish behaviour in aquaculture sea cages and detecting anomalies to help the farm operators. For instance, Pinkiewicz *et al*. [27] presented a system which monitors 30 random fish in aquaculture sea cage throughout the day. With the features such as coordinates of fish bounding box, the size of the bounding box and eccentricity ratio, fish were tracked using a Kalman filter. Fish trajectories were represented in terms of average swimming speed and the direction. Using a threshold for calculated trajectory features, normal and abnormal behaviours were distinguished.

For quantification of zebrafish behaviour Kato *et al*. [28] presented an image processing based system. Multiple fish were tracked individually in a single aquarium. The interaction of two zebrafish were investigated in terms of chasing behaviour. Chasing was defined by the distance, approach and the angle of the two fish. This study is useful as it can be used as a tool to detect disorganised schooling in larger areas with many fish which is important for fish ecology.

A recent problem in this area and also the problem that we are interested in is automatic fish motion pattern analysis in underwater environments [18, 29]. Spampinato *et al*. [18] proposed an Adaptive Gaussian Mixture Model with the Adaptive Mean Shift algorithm to detect and track fish in underwater videos. Fish species were recognised using affine invariant descriptors to describe texture and shape based features (third moment, fourth moment, Fourier descriptors, and curvature scale space and so forth). Fish trajectories were sub-sampled using Douglass-Peucker algorithm and then clustered using I-kMeans. Fish species were associated with the extracted trajectories. This study can be seen as a preliminary work since it did not include any evaluation of the trajectory analysis. However, it is still important as it uses underwater videos and shows the importance of fish behaviour analysis in that field. On the other hand, using fish motion patterns, the underwater videos were classified [29]. Fish trajectory was modelled in terms of fish swimming speed, direction, periodicity and escape response time. Using three sea depths, six behaviour patterns of fish were defined. A new video was identified in terms of sea depth using the motion pattern. To do that: optical flow was used to detect the fish and optical flow vectors were clustered using an agglomerative hierarchical clustering method. Then, a tracking step were applied. Histograms of fish displacements were extracted and by using a Random Forest classifier different fish motion patterns were identified for video classification. In a different study,

Spampinato and Palazzo [9] applied an HMM to detect tracking faults (wrong trajectories) which occur due to background plant movements, object occlusions, and tracker mis-associations. Correct and false trajectories were first scaled by multi-dimensional scaling (MDS) and then clustered using k-means. For the k clusters, k HMMs were trained. If the maximum likelihood of a new trajectory using the k HMMs is lower than a threshold, then the new trajectory was classified as incorrect. This study is different in terms of it's aim but important as it presents an anomaly detection framework using fish trajectories.

In summary, the majority of works analysed the fish trajectories in a fish tank [25], aquarium [20] or an aquaculture sea cage [27] which actually makes the analysis simpler as it decreases the number of fish behaviours, the variety of fish behaviours and most importantly eliminates the effects of habitat on the behaviour of fish. A few studies analysed videos of natural habitat underwater environments [18, 29, 9]. Some studies focused on behaviour of individual fish [18, 21, 22] while other studies considered fish schools [20, 25]. Most of the studies analysed only one species like [27, 25, 28, 30] while some of them considered more species [18].

2.2 Related Works on Unusual Trajectory Detection

Trajectories describe the displacements of objects and are typically considered as positions in 2 dimensions over time. Unusual trajectory detection studies can be categorised based on:

- The trajectory representation methods that they utilised,
  - Using raw trajectory positions, reproducing trajectory positions such as by polynomial fitting, etc.
  - Extracting multiple features from trajectories such as velocity, acceleration and shape based features, etc.
  - Combinations of these.
- The learning method that they used
  - Unsupervised,
  - Supervised,
  - Semi-supervised.

2.2.1 Trajectory Representation Methods for Unusual Trajectory Detection

To reproduce trajectories from the original tracks, Morris and Trivedi [17] categorised possible methods as:

- Vector quantization,
- Polynomial fitting,
- Multi-Resolution Decomposition,
- Hidden Markov Model (HMM),
- Subspace Methods,
- Spectral Methods, and
- Kernel Methods.

Those methods were used to represent trajectories for trajectory clustering, path modelling, unusual trajectory detection, automatic activity analysis, and activity recognition. Here, we discussed only the methods that were used for unusual trajectory detection.

Polynomial fitting such as Least Square Polynomials, Chebyshev Polynomials and Cubic B-spline curves try to fit simple 2D curves to trajectories. For instance, Sillito and Fisher [31] proposed a semi-supervised anomalous trajectory detection method using cubic B-spline fitting. Li *et al*. [32] adapted the proposed B-spline approach in [31] to represent trajectories. Similarly, Makris and Ellis [33] applied spline fitting to extract common pathways from a set of pedestrians' trajectories. Spline fitting does not need machine learning methods but the accuracy depends on the chosen number of control points. If an incorrect number of points is chosen some trajectory dynamics might not be represented correctly. This might result in ignoring sharp changes in trajectory which may influence the discrimination of trajectories.

Haar representations and the Discrete Fourier Transform (DFT) are the most frequently used multi-resolution techniques. For instance, Naftel and Khalid [34] used the DFT to map trajectory time series to the frequency domain. Using Self Organising Map (SOM), trajectories were clustered in the chosen feature space. Anomalous trajectories were found as the trajectories sufficiently distant from all identified trajectory

clusters. Although the DFT representation is simple, it is unable to represent the temporal occurrence of frequency changes in a signal as it only represents the frequency content. Additionally, it was not successful at representing complex trajectories [34].

An alternative representation approach is using Hidden Markov Models (HMM) [35, 36, 37]. Suspicious human activities in a scene were identified as a part of a surveillance system which is capable of detecting and tracking people as well [37]. In that study [37], all possible normal activities were represented using HMMs. Any activity with low likelihoods from all the HMMs was classified as anomalous. The HMM is useful if the trajectory length is fixed for all trajectories. However, usually the lengths of trajectories are not equal. Therefore, to use HMM trajectory interpolation might be needed. Moreover, HMM based representations need training data to define the states and transition matrices.

Principal Component Analysis (PCA) is a well known subspace method which uses eigenvectors to project data into a lower dimension space. Using PCA was used by Bashir *et al.* [38] to represent segmented trajectories. The trajectories were segmented into atomic actions by perceptual discontinuities (which possibly occur due to occlusions and noise) in the trajectory using velocity and acceleration. All similar activities were then used to form a single data matrix and the principal components of this matrix were used to find a compact representation. Lastly, trajectories were classified using a HMM where an unusual trajectory can be found with the same approach as given above [37]. PCA is useful as it provides a compact representation but the number of components should be determined carefully as it is possible to lose a part of the trajectory information.

To compare popular trajectory representation methods Sillito and Fisher [39] used a fixed arc-length vector representation and applied Haar wavelet coefficients, DFT, Chebyshev polynomial coefficients and cubic B-spline to pedestrian trajectories, vehicle trajectories, hand trajectories and pen trajectories. These techniques were evaluated in terms of class separability while this metric is useful to evaluate an unusual trajectory detection method. The Haar representation was found to be better than the DFT while the highest separability values were obtained by Chebyshev or B-spline representations.

The most popular trajectory representation methods for unusual trajectory detection are summarised with their advantages, disadvantages and example references in Table 2.1.

Table 2.1: Comparison of the most popular trajectory representation methods for unusual trajectory detection with example references.

Method	Advantage	Shortcoming	Reference
Cubic B-spline	Does not need learning.	Accuracy depends on choosing the correct number of control points.	[31, 33, 32]
HMM	Successful if the trajectory length is constant.	Needs training data to define the states and transition matrix.	[35, 36, 37]
PCA	Provides a compact representation using eigenvectors.	Accuracy depends on correct number of PCA components.	[38]
Haar	Does not need learning.	Not able to represent complex trajectories.	[34]
DFT	Very simple.	Does not represent the temporal occurrence of trajectories.	[39]

As mentioned above, rather than explicitly reproducing the trajectories, the trajectories can be represented by the multiple features derived from the trajectories [40, 41, 42, 43, 44, 45, 46, 47, 48, 49]. For example, Zhong *et al*. [40] used colour and texture histograms. Behaviour patterns were classified as normal and unusual using the co-occurrence of these features. Porikli and Haga [41] proposed to use object based and frame based features together to detect abnormal behaviours. In that study, object based features includes the histogram of aspect ratio, orientation, speed, colour size of the object, the HMM trajectory representation, duration, length, displacement and global direction of the trajectory. As frame based features histogram of orientations, location, speed, size of objects were used. In the literature, extracting such multiple features from raw trajectories is very common for unusual trajectory detection. Some of interesting studies are addressed in Section 2.2.2 while discussing their proposed learning methods.

2.2.2 Learning Methods for Unusual Trajectory Detection

Unusual trajectory detection algorithms are commonly unsupervised. Unusual trajectories are those that are not similar (close) to any known clusters using a pre-defined distance threshold or are trajectories that are similar to clusters that have few trajectories. An earlier work in this category is [50] which used Self Organising Maps (SOM) to detect unusual trajectories. In that study, the trajectories were translated into a feature vector in terms of time smoothed positions and instantaneous velocity. The Euclidean distance between trajectories and clusters and a pre-defined distance threshold were used to find the unusual trajectories. A trajectory having a distance larger than a threshold was classified as unusual. Differently, Hu *et al*. [42] presented a hierarchical

trajectory clustering method to detect abnormal trajectories and make behaviour predictions. The position, the velocity and the size of the object were used to describe trajectories. At the first level of the hierarchy, trajectories were clustered using spatial information. At the second level, clustered trajectories were grouped according to temporal information. Abnormal trajectories were defined as the trajectories that belong to clusters having few samples. Another unsupervised unusual trajectory detection method was proposed by Izo and Grimson [51]. Normal and unusual trajectories were individually clustered using the Normalised Cuts Spectral Clustering algorithm. To represent the trajectories, a feature vector composed of the area of the object's bounding box, the speed, the direction of motion and the object position in the image were used. To classify a new trajectory, it was projected into the spectral embedding space of the obtained clusters and matched with the clusters. 4-D histograms in terms of 2-D trajectory position and the instantaneous velocity were used for unusual event detection in [52]. First trajectories were clustered using a GMM with an outlier removal which considers the direction of motions and then 4-D histograms were analysed to examine the local characteristics of the trajectory. The number of clusters were determined by the finite mixture models [53]. Outlier removal was presented with a split and merge procedure which was based on the Bhattacharyya distance. A new trajectory was classified as normal or unusual by comparing its features with the histograms that were obtained using training data and the thresholds that are specific to each cluster. A 3-stage unsupervised hierarchical trajectory and activity learning process with an abnormal trajectory detection method was presented in [15]. The trajectory points and the velocity extracted from the trajectory were used. In the first stage, interesting nodes were learned by a GMM. In the second stage, the routes which represent each trajectory cluster were extracted using Longest Common Subsequence (LCSS) distance and spectral clustering. Following this, the dynamics of activities were encoded using HMMs. The abnormal trajectories were determined by comparing the trajectory's log-likelihood with a threshold. This study is important as it gives a very nice flow for the trajectory learning process and unusual trajectory detection system as a sub-problem of trajectory learning. In [54], multiple features such as velocity, directional distance, target trajectory mean, initial target position, speed, acceleration, PCA transformed trajectory points and trajectory turns were used to cluster the trajectories and then to detect anomalies. The Mean-shift algorithm was applied to normalised trajectory features to obtain trajectory clusters. The abnormal trajectories were defined as outliers to the clusters that are different from the other trajectories in the same cluster

or the trajectories that belong to a cluster which has few samples. In a recent work [55], trajectories were examined at three levels as spatial, directional and object type (vehicles and pedestrians) and different clustering methods for each level were applied for unusual traffic behaviour detection. For the spatial level, a trajectory similarity matrix using the Hausdorff distance was extracted and spectral clustering was applied. For the directional level, start and end points of the trajectories were used and a GMM was applied. For the type level, based on the object's class, k-means clustering was applied. The output of those levels were combined as multi-level motion patterns. Abnormalities were detected as the trajectories that do not fit to any motion patterns that were found during training.

As an alternative to clustering, topic models can also be used to detect unusual trajectories. For instance, Probabilistic Latent Semantic Analysis (pLSA) was used in [56]. By using pLSA the co-occurrence of motion paths was determined and unusual paths were found. An unusual activity occurs if the pLSA predicts it as being very rare or if it has a log-likelihood below a threshold. Similarly, Varadarajan and Odobez [57] also used pLSA with the location, the direction and the shape features extracted from trajectories. Abnormality detection was performed by different metrics such as log-likelihood, Kullback-Leibler divergence and Bhattacharyya distance. For instance, when the log-likelihood is used, normal trajectories have a high log-likelihood while an abnormal trajectory does not fit any learned topic. As a different topic model Latent Dirichlet Allocation (LDA), which was applied in an unsupervised way for unusual trajectory detection, can be used [58, 59]. In those studies, trajectories were grouped by LDA and represented by HMMs.

In contrast to the studies detecting unusual trajectories with an unsupervised approach, there are other studies that utilise semi-supervised or supervised methods. Support Vector Machines (SVM) [60], HMM [43, 44, 45], and Dynamic Bayesian Networks (DBN) [46, 47, 48, 49] are the popular classifiers which were applied in a supervised or semi-supervised way using the trajectories either fully labelled as normal and unusual or only as normal.

As an example, Ivanov *et al*. [60] used velocity and acceleration features extracted from trajectories to detect unusual activities such as running or careless driving. In that study, a SVM was used and a model was trained using typical normal and unusual trajectories. The learned model was used to detect new unusual activities. Xiang and Gong [46] found natural groupings of trajectories using the eigenvectors of the behaviours' affinity matrix. They presented a time accumulative reliability measure to

detect abnormalities. Once a sufficient number of trajectories that belong to the same behaviour class is observed (which is determined by the reliability measure) the normal trajectories were determined *on-the-fly* without manual labelling in order to detect the abnormalities. In detail, a trajectory was defined in terms of the centre of bounding box of detection, the width and the height of the bounding box, its shape (filling ratio of foreground pixels within the bounding box associated with the blob) and the first order moment. The number of behaviour classes and the behaviour patterns were automatically determined using Bayesian Information Criterion (BIC) with a GMM. These behaviour patterns were used to find the natural groupings and each group was represented by a DBN with Multi-Observation Hidden Markov Model (MOHMM) topology. For each detection of a new trajectory, the log-likelihood of it was determined by the MOHMM model. Then, all log-likelihoods were used to determine the abnormality of the trajectory by comparing the reliability measure which is based on a threshold. The same authors explored that method more deeply in [47] by comparing the performance of a behaviour model trained using an unlabelled data set with a behaviour model trained using the same but labelled data set. The results showed that the trained model using an unlabelled data set is better than the trained model using the same but labelled data set for detecting abnormalities from an unseen video. The proposed method in [46, 47] was adapted in [48] to use incremental learning to detect anomalies in the video. In that study [48], instead of using a reliability measure, the likelihood ratio test (LRT) was used to detect anomalies. The advantage of that study is learning the model incrementally (based on expectation maximisation) and online with a small initial training set. Another study [49] aimed to detect and discriminate different types of anomalies based on their temporal duration and order using a Cascade Dynamic Bayesian Network (CasDBN). Behaviours were modelled by defining atomic actions in terms of the actions' order and temporal duration. The features: the blob centre, the width, the height of the bounding box, the occupancy, the ratio of the dimension, the mean optical flow of the bounding box and the scaled optical flow were used. It was assumed that normal behaviour should follow a typical order of atomic actions with certain durations while a deviation in temporal order or temporal duration causes an anomaly. Three kinds of anomalies: *i)* the behaviour patterns which are visually different from what have been observed from the training set, *ii)* the behaviours that are ambiguous since they have rare occurrence and *iii)* the behaviours supported by only very weak visual evidence were considered. Different DBN models were constructed to model the behaviours with a two-stage CasDBN. In the first

stage, a first-order HMM was used to model temporal order while in the second stage, a MOHMM was used to model the temporal duration. To detect and discriminate different classes of anomalies, thresholds specific to each stage were used for comparing a sample with the normalised log-likelihood of the behaviour. A different and uncommon supervised learning based unusual trajectory detection method was presented in [32]. Using trajectory sparse reconstruction linear reconstruction coefficients were found from labelled data. Normal trajectories were defined as the trajectories produced by the people walking from one exit to another while unusual behaviours represents activities such as fighting, falling down, leaving packages, etc. Although this study is different in terms of the supervised learning method, it was very sensitive to chosen thresholds.

As an example of semi-supervised unusual trajectory detection method, Sillito and Fisher [31] proposed using a GMM to learn normal and unusual trajectories. When a trajectory is classified as unusual by the model, the human operator decided whether or not the trajectory is normal and based on this, the model was incrementally updated. The trajectories which were classified as normal never went to the human operator to be labelled. A trajectory was classified as unusual by the model if its Mahalonobis distance to the closest component of the GMM exceeds a pre-defined threshold. The advantage of this system is the capability of classifying new trajectories at any time during the training. Differently, Luhr *et al*. [44] and Duong *et al*. [45] presented semi-supervised methods for nursing home and smart home systems respectively. They both utilised variations of HMMs such as a fully connected explicit state duration HMM (ESD-HMM) and Switching Semi-HMM (S-HMM). Both of these studies detected anomalies based on the order of the activities and the durations of the activities. As a different approach, a weakly-supervised joint topic model (WS-JTM) which is based on LDA was used to find rare and subtle behaviours (defined as the sparse behaviours among typical behaviours and not repeated enough to be modelled precisely) [61]. The advantage of that work [61], is being able to detect rare behaviours even with a few training data and being able to detect the anomalies which have very small spatio-temporal deviations.

In summary, here we reviewed the most interesting research on the unusual trajectory detection problem. Most of the works are unsupervised and usually based on clustering, although there is not any single clustering method that was particularly successful. For those methods, using the similarity between the trajectories and the known clusters with a pre-defined threshold is the most common way to detect unusual tra-

Table 2.2: Most popular learning methods for unusual trajectory detection

Method	Reference
Clustering	[50, 62, 36, 41, 63, 64, 42, 51, 65, 31, 52, 66, 54, 67, 68, 69, 70, 71, 72, 15, 73, 74, 75, 55]
HMM and Variants	[44, 45, 43, 76, 77, 78, 79, 15, 80]
DBN	[46, 47, 48, 49]
SVM	[81, 14, 82, 60, 83]
Topic Models	[56, 57, 84, 58, 59, 61, 85, 75]

jectories. On the other hand, first applying clustering and then modelling each cluster with a HMM to detect the unusual trajectories as the trajectories having low-likelihood is a frequent approach as well. The popular learning methods with the corresponding references are summarised in Table 2.2.

2.3 Related Works on Imbalanced Data Classification

Applications utilising imbalanced data sets are diverse such as text categorisation, medical diagnosis, fault detection, fraud detection, video surveillance, image annotations, anomaly detection [86, 87, 88, 89]. The diversity in applications has led to different solutions over the years. Approaches are traditionally divided into four categories: *i)* algorithmic level, *ii)* data level, *iii)* cost-sensitive methods and *iv)* ensembles of classifiers.

- **Algorithmic Level:** In this approach, the classifier is forced to converge to a decision threshold biased to an accurate classification of the minority class such as by adjusting the weights for each class. For instance, in [90] a weighted Euclidean distance function was used to classify the samples. Similarly, a SVM with a kernel function biased to the minority class was presented in [91, 92, 93, 94] to improve the minority class prediction.

- **Data Level:** Re-sampling the data in order to handle the problems cause by the imbalanced nature of data is the data level approach. This approach does not modify the existing classifier and is applied as a pre-processing technique. The training data set can be re-sampled by over-sampling the minority class samples [95, 96] and/or under-sampling the majority class samples [97, 98, 99].

One of the most popular re-sampling approaches is SMOTE [95] which synthesised new minority class instances. In this approach, for each minority class sample a new sample was created on the line joining it to the nearest minority class neighbour. As it did not replicate minority examples but created new samples, it overcame over-fitting. Previously, it has been combined with many classifiers such as SVM [94, 100], Naive Bayes [95], C4.5 [87, 95], Random Forest [101, 102]. Even though it is popular and works better than only under-sampling the majority class, it does not always achieve better classification performance compared to the original classifiers as observed in [102]. A possible reason for this can be that the newly generated samples might cause class overlapping or because it gives each minority class sample equal importance and does not pay more attention to the samples for which classification is harder. There are some improved versions of SMOTE to overcome its shortcomings such as Borderline-SMOTE [103], SMOTEBoost [104], and modified SMOTE [105]. For instance, Borderline-SMOTE [103] assumed that the data samples close to the decision boundary are more important as they might cause misclassification and use those samples to create new samples.

Even though re-sampling techniques are independent of the classifier, it is usually hard to determine the optimal re-sampling ratio automatically. It might be problematic to over-sample minority classes yet keep the distribution the same, especially in real-world applications where overlaps between minority and majority classes are highly likely. Therefore, over-sampling potentially results in over-fitting [106]. Moreover, when over-sampling is applied, the computational cost is also increasing which is not well suited to very large data sets. On the other hand, while under-sampling the majority class, it is usually difficult to keep the new distribution of the majority class as similar as the distribution that it is sub-sampled from. Additionally, it is possible to throw away some useful samples and thus increase variance in estimating model parameters [107].

- **Cost-sensitive Methods:** This approach was usually applied by earlier studies where different costs are assigned to training examples of the majority and the minority classes [108, 109]. The classifier is biased toward the minority class with a higher mis-classification costs while it tries to minimise the total classification error for both classes. However, it is difficult to set the cost properly and it may depend on the characteristics of the data sets. The standard public classi-

fication data sets do not contain the costs [86, 87] and over-training is possible when searching to find the most appropriate cost. Different cost functions were combined with the classifiers such as k-nearest neighbours (kNN) [110], SVM [111], decision trees [112], logistic regression [113]. For instance, in [113] the classification performance of logistic regression for mine classification was improved compared to pure logistic regression, although it was still not successful with the data sets that have very few minority class samples.

- **Ensembles of Classifiers:** This category has been popular in the last decade. In general there are two main approaches: bagging and boosting. Bagging contains different classifiers which are applied to subsets of the data [114]. Alternatively, in boosting, the whole set is used to train classifiers in each iteration while more attention is given to the classification of the samples that are misclassified in the previous iteration. This is done by adjusting the weights toward their correct classification. The most well known boosting method is AdaBoost [115].

  Even though ensembles are frequently used for classification of imbalanced data sets, they are not able to handle the imbalanced data sets by themselves. And they require one or a combination of the approaches that are mentioned above such as re-sampling data (SMOTEBoost [104], EUSBoost [87] etc.). For instance, Radivojac *et al*. [116] presented bagging with over-sampling for a bioinformatics application. Liu *et al*. [117] proposed a double ensemble classifier by combining bagging and boosting. In that study, EasyEnsemble and BalanceCascade were used for bagging in the first ensemble and also for each bag AdaBoost [115] was used. Sampling and ensemble techniques were again combined in [118]. This method is similar to SMOTEBoost [104] as being simpler, faster and performing better. It removes the majority class samples until the training set become balanced, assuming that classification of balanced data sets are better. However, the results showed that making the data set completely balanced can sometimes result in lower performance. Pure SVM was compared with ensembles of SVM in [119]. With ensembles of SVM the minority class prediction was increased. In that study [119], Boosting-SVM with Asymmetric Cost found as the best compared to methods such as SMOTEBoost [104], random undersampling with SVM and SVM-SMOTE [95].

In summary, the number of proposed approaches in this field is very large, and the studies are interesting, as imbalanced data sets implies a significant challenge for

Table 2.3: Confusion matrix for a two-class problem

	Prediction as Minority Class *(Positive Class)*	Prediction as Majority Class *(Negative Class)*
Minority Class *(Positive Class)*	True Positive (TP)	False Negative (FN)
Majority Class *(Negative Class)*	False Positive (FP)	True Negative (TN)

machine learning and data mining applications. The evaluation criteria used is also as important as the methods to make a good and fair evaluation to determine the successful and/or unsuccessful methods. Therefore in Section 2.3.1, we discuss the suitable evaluation metrics for imbalanced data classification.

2.3.1 Evaluation Metrics for Imbalanced Data Classification

The choice of appropriate evaluation criteria (such as feature selection criterion to lead the training process and/or the metric to evaluate the performance of the classifiers) is very important when dealing with imbalanced data sets since it might cause to ignore classification of minority class examples due to processing them as noise [86, 87]. For a two-class problem, the confusion matrix shown in Table 2.3 is used to define evaluation metrics.

The most common metric is the accuracy (Eq. 2.1) which is calculated as the sum of correctly predicted minority and majority samples over the total amount of samples. However, for imbalanced data sets, accuracy is not suitable and is not used as it misguides the classifier and ignores the importance of minority class since it is underrepresented [86, 87, 94, 100]. Using accuracy might even lead to total misclassification of the minority class if the imbalance ratio (the number of the minority class samples over the number of the majority class samples) is very low and the data is highly overlapping.

$$ACC = \frac{TP+TN}{TP+FN+TN+FP} \tag{2.1}$$

For this reason, many alternative metrics have been proposed for evaluation of imbalanced classification. Those metrics are based on *True Positive Rate (TPrate)* which represents the percentage of positive samples (the minority class samples) correctly

classified (Eq. 2.2), *True Negative Rate (TNrate)* which represents the percentage of negative samples (the majority class samples) correctly classified (Eq. 2.3), *False Positive Rate (FPrate)* which represents the percentage of negative samples misclassified (Eq. 2.4), *False Negative Rate (FNrate)* which represents the percentage of negative samples misclassified (Eq. 2.5) and *Positive Predictive Value (PPValue, Precision)* which represents the percentage of predicted positive samples (Eq. 2.6).

$$TPrate = \frac{TP}{TP+FN} \tag{2.2}$$

$$TNrate = \frac{TN}{TN+FP} \tag{2.3}$$

$$FPrate = \frac{FP}{FP+TN} \tag{2.4}$$

$$FNrate = \frac{FN}{FN+TP} \tag{2.5}$$

$$PPValue = \frac{TP}{TP+FP} \tag{2.6}$$

The geometric mean of $TPrate$ and $TNrate$ (GeoMean) [97] (Eq. 2.7) which encourages equal classification for both classes, the adjusted geometric mean (A-GeoMean) [120] (Eq. 2.8) which increases the $TPrate$ as high as possible by keeping the reduction of $TNrate$ as low as possible, the Area Under Receiver Operating Characteristic curve (AUC) [121] (Eq. 2.9), which corresponds to the area that is created by the probability of $TPrate$ and $FPrate$ and the F-measure (F-1 score) [100] (Eq. 2.10) which uses the $TPrate$ and $PPvalue$ are the most common and effective metrics for imbalanced data set classification [86, 87, 100, 122, 123]. Among these metrics, it is hard to distinguish the best one but the most common two metrics are $GeoMean$ and AUC.

In our work, we used $TPrate$, $TNrate$, $GeoMean$, $AGeoMean$ and AUC as they are suitable and popular metrics for the work we deal with.

$$GeoMean = \sqrt{TPrate \times TNrate} \tag{2.7}$$

$$AGeoMean = \begin{cases} \frac{GeoMean+TNrate\times N_n}{1+N_n}, & \text{TPrate } >0 \\ 0, & \text{TPrate = 0} \end{cases} \tag{2.8}$$

where N_n refers to proportion of the negative class (majority examples).

$$AUC = \frac{1 + TPrate - FPrate}{2} \tag{2.9}$$

$$F - Measure = \frac{2 \times TPrate \times PPValue}{TPrate + PPValue} \tag{2.10}$$

2.4 Related Works on Hierarchical Classifiers and Hierarchical Decomposition

Classification using hierarchical methods can be divided into two categories [124]:

- **Hierarchical Classifiers:** A pre-defined hierarchy such as a taxonomy exists and the classes are organised using this taxonomy as a tree or a graph.
- **Hierarchical Decomposition:** There is no pre-defined hierarchy and the hierarchy is created during training using factors such as similarity of data.

Hierarchical classifiers have been addressed in many studies [125, 126, 127, 128]. For instance, a tree shaped class taxonomy was used for a multi-class problem where any existing learning method can be adapted for single learning tasks [126]. Binarised and split-based structured label learning approach were described and a loss function for evaluation of the resulting structured classifiers were defined. Li *et al*. [127] presented a method which uses taxonomy for automatic music genre classification. In that paper [127], the taxonomy is in terms of the relationship between the genres. Additionally, automatic taxonomies were built using the similarity matrix from linear discrimination. Classification in large taxonomies was re-visited with improved results in [128]. In that work [128], solutions for error propagation which affects the classification of the lower levels of the hierarchy and the complex decision boundaries occurring in the higher levels of the hierarchy were studied. For hierarchical protein function prediction Silla *et al*. [129] proposed a method which uses a fixed taxonomy. Given that taxonomy, selecting the best classifier, selecting the best feature representation given a fixed classifier and selecting the best classifier and the best feature representation together were compared.

Hierarchical decomposition is as popular as hierarchical classifiers and in this section we pay more attention to them as we also propose a hierarchical decomposition technique. The most common approach for hierarchical decomposition is dividing

a multi-class problem in a hierarchical way to obtain binary hierarchical problems [124]. In this technique, a hierarchy can be created using the similarity of the classes. For instance, classes were divided in a hierarchical way in [130] where similar classes are grouped together and the multi-class classification problem turned to binary classification problem for hyper-spectral data analysis. In that study, a set of classes recursively partitioned into two groups and the best feature set that distinguishes those two group was found at the same time. In a similar study [131], hierarchical max-cut unsupervised decomposition was presented for multi-class classification. In that method, classes were partitioned into two subsets until one class label was obtained at the leaf node based on class similarities. As the classifier, SVM was applied at each node to find the best discrimination function for binary meta-classes. Therefore, that method was called as hierarchical SVM. The comparisons with the state of art methods on hyper-spectral data showed that, that method [131] performed accurately. In a different application area but a similar concept, the SVM based hierarchical method which is based on clustering was used for text mining using the similarities between features [132]. Using the hierarchy, the problem was divided into smaller problems and therefore a smaller and more specific set of features were selected for each sub-problem. This increased the accuracy and efficiency [132]. Freitas *et al.* [133] proposed generation of meta-classes *on-the-fly* without using a fixed taxonomy for handwriting recognition. Using the disagreement of the characters and Euclidean distance between the confusion matrices, a two level hierarchy was built where the leaf node were constructed by the similarity of the meta-class level. Epshtein and Ullman [134] used the relationship between features to construct an automatic hierarchy. The same feature extraction procedure was applied at all levels of the hierarchy. The top-level features were broken into their smaller components and for all levels of the hierarchy different features, sub-features and their specific parameters were learned using the training samples. The results showed that dividing features into a hierarchy performs better than using features as a whole. Entropy based feature selection combined with hierarchical clustering was applied to construct binary hierarchy in [135]. Different feature subsets were used at different levels of the hierarchy. At each level of the hierarchy SVM was used. The results showed that, that method [135] is better than one-against-one hierarchical decomposition for multi-class audio event classification for health-care applications.

Studies such as [13, 132, 134, 136] showed that hierarchical methods can have better classification performance compared to flat classification techniques. In summary,

hierarchical methods have been used for classification in different application areas. They are preferred especially if the problem is a multi-class problem, high-dimensional data exist and the data set is large.

2.5 Related Works on Active Learning

Active learning is a field of data mining and machine learning which considers the cost of data labelling [137]. The goal of active learning is to achieve better learning performance with fewer training instances [138, 139]. When unlabelled data is abundant, labelled data is limited and labelling is expensive, active learning is very useful. Active learning seek to choose the most informative unlabelled training instances with a query strategy. This requires labelling only selected instances which decreases the labelling cost in contrast to passive learning where the labels of all training examples are required.

Based on how the queried instances are sampled, active learning has 3 subtypes: *i)* membership query synthesis, *ii)* stream based selective learning, *iii)* pool based sampling [138]. In membership query synthesis, queried instances are artificially created therefore they might not have appropriate labels [140]. On the other side, with stream based selective learning, and pool based sampling, the queried instances are always real examples which means their labels can be provided by the expert. In stream based selective learning, the learner decides to query or discard the instances while in pool based sampling queries are selected from a pool of unlabelled samples. The main difference from stream based selective learning is the large amount of unlabelled samples during the query time [140]. As the most popular type of active learning is pool based sampling and it is the most relevant technique for imbalanced data classification, in this thesis (including the experiments and the literature review) we consider pool based active learning and refer to it as active learning. The details of pool based active learning are given in Section 7.1.

In this section, an overview of related works on pool based active learning in terms of imbalanced data set classification is given. Additionally, active learning with feature selection is also investigated.

Active learning has been examined in different domains and several studies have addressed this problem. The majority of the studies in this field specifically focused on imbalanced data set classification. In this thesis, we also focus on active learning for imbalanced data sets, although we made some experiments on balanced data sets

as well.

For imbalanced data sets, there are two main techniques for using active learning [139]:

- Balance the training set and then apply one of the standard query strategies (instance selection approach which defines the rule that is applied to select informative instances at each iteration of the active learning) such as uncertainty,
- Propose a novel query strategy which is specific to imbalanced data classification.

For instance, Haines *et al*. [141] preferred the first technique where minority class samples were synthetically over-sampled to make the data set balanced and then, uncertainty sampling was applied to the balanced data set. Similarly, Bootstrap based over-sampling is combined with uncertainty based sampling and used to eliminate class imbalance for word sense disambiguation [142]. Doyle *et al*. [143] selected informative instances considering the class balance for histopathology annotation. They first determined the most uncertain samples, labelled them, then randomly selected samples which make the training set balanced. Holub *et al*. [144] advocated using uncertainty based selection using different uncertainty schemes such as least confident instance, margin and entropy.

The most popular query strategy is uncertainty [145]. Alternatively, a method based on a SVM classifier was proposed in [146]. In this work [146], informative instances were determined by the distance to the SVM hyperplane where samples close to hyperplane are more informative compared to the rest of the samples. Li *et al*. [147] proposed a new query strategies called co-testing and self-selecting for imbalanced sentiment classification. The proposed strategies were compared with random selection, margin based selection [146], uncertainty and certainty. In that work [147], co-testing was defined as the strategy which selects the informative samples those have a low confidence score where the confidence score is based on the class prediction of different classifiers. Self-selecting selects k uncertain sample sets and then randomly select samples from each class inside of the uncertain sample sets to make the training set balanced as Doyle *et al*. [143] were applied. The results showed that margin-based selection [146], uncertainty and self-selecting did not perform better than random selection while co-selecting was better than random selection in 2 domains out of 4 domains with a slight improvement.

Attenberg *et al*. [148] showed that for the data sets which have very few minority class samples with much class overlapping, it is hard to get reliable posterior probabilities especially at the first stages of the active learning. To handle this, Uguroglu [139] proposed a new strategy which is called maximum probability. In that study [139], instances were selected with higher probability to be from the minority class in order to keep the training set balanced. The location of the samples was not considered and in case of wrong probability estimation it was assumed that the samples which were mistaken as minority class are highly informative majority class samples.

Another very popular query strategy is expected error reduction, which was first proposed by Roy and McCallum [149]. This query strategy tries to estimate the future error of a learning model when unlabelled instances are combined with the current labelled training data and uses the remaining unlabelled instances as the validation set. The aim is to select the instances with minimal expected error. Each instance is tried with all possible labels using the current trained model. Therefore, even though this method is successful, it is one of the most computationally expensive query strategies. To the best of our knowledge, the expected error reduction query strategy in active learning has not been specifically used for imbalanced data classification. However, it was combined with many learning methods such as Naive Bayes [149], Gaussian Random Fields [150] and SVM [151].

Information density based selection was proposed in [152] and became a popular query strategy. In this strategy, informative instances are not only those which have high uncertainty score but also those which have high similarity score when similarity is calculated in terms of the distance between each unlabelled samples. Similar to expected error reduction, this strategy has not been specifically used for imbalanced data classification yet. For instance, in [152], information density was compared with uncertainty, query-by-committee [153] and random selection using 6 evaluation corpora. The results showed that information density is usually more successful than others and can be recommended. In this thesis, we applied this strategy to compare its performance with other strategies for active learning with feature selection and also combined it with different classifiers.

As seen, none of these works focused on active learning in combination with feature selection. Additionally, there is not much work on active learning with feature selection, although in the literature, there is a vast amount of research on active learning and feature selection individually.

To the best of our knowledge, previous studies about active learning with feature

selection all belong to the natural language processing field (especially text classification) such as [154, 155, 156]. In those studies, the feature space is implicitly changing since features are based on word frequency and any new training data means changes in the feature space. Moreover, those studies usually did not compare active learning performance with and without feature selection (using all features in the training set) but instead tried to find a way to determine the best features manually using a human expert. The increase in active learning effectiveness by determining the best features was addressed in [154] for text categorisation. However, in [154] features were selected by human annotators while selected instances were being labelled. Similarly, in [155], features were ranked using cluster based feature selection and then best features were selected by users for document clustering. This study also showed that selecting effective features guides active learning positively. Differently, Bilgic [157] proposed an adaptive dimensionality reduction technique that determines the proper number of dimensions for each active learning step. The results of active learning with/without dimensionality reduction showed that active learning with dimensionality reduction performed significantly better than without dimensionality reduction. Moreover, even though paper [157] did not aim to compare different query strategies, it can be inferred from the results that random selection with dimensionality reduction was never worse than other query strategies. Okabe *et al*. [156] proposed an active learning and feature selection based method for interactive spam filtering. In that study [156], Naive Bayes was used as the classifier. As the feature selection algorithm, information gain was used which is perhaps more suitable for text classification. The reported results showed that feature selection affects the active learning performance positively. However, no comparison with random instance selection was included while uncertainty selection and error reduction sampling were used.

2.6 Summary

As seen fish behaviour understanding studies are mainly about water quality monitoring and chemical identification in fish tanks or aquaria. However, fish behaviour analysis in natural underwater scenes is very limited while it is more challenging due to the reasons given in Chapter 1. On the other hand, there is a large amount of work on unusual trajectory detection. These works can be distinguished based on the trajectory representation method and the learning method. Based on the review, we can see that fish behaviour analysis using natural underwater videos is very limited and an

unusual fish trajectory approach using those videos has not yet been addressed. In this thesis, we investigate unusual fish trajectory detection from underwater videos. As the trajectory representation we are using trajectory positions and also extract novel multiple features from fish trajectories. All the proposed methods are based on supervised learning which is rare compared to unsupervised unusual trajectory detection methods.

Imbalanced data classification is also a very popular area especially in pattern recognition and machine learning since many real world data sets are implicitly imbalanced and traditional methods are not very successful with these data sets. As given above there are many different approaches to handle class imbalance. In this thesis, we propose a hierarchical decomposition method (Chapter 5) for imbalanced data classification which is different from previously proposed solutions to the class imbalance problem. Additionally, it does not require any data pre-processing step as many other solutions need. The proposed hierarchical decomposition method is also novel as the hierarchy is constructed using the similarity of labelled data subsets at each level of the hierarchy with different levels being built by different data and feature subsets. This is in contrast to previous research that uses the same feature set for every level of the hierarchy or a flat classifier (the most similar work is [124] but it uses a fixed taxonomy which we do not use).

Active learning is a successful approach to provide faster learning with lower labelling cost. It is also one of the solutions for imbalanced data sets classification. As seen, there are popular query strategy methods such as uncertainty [145] and recent strategies are generally based on it. On the other hand, feature selection is a well-studied subject which generally increases the classification performance by selecting the best features and decreases the size of the feature space. However, the literature is very limited in terms of active learning with feature selection. In this thesis, we mainly integrate the proposed hierarchical decomposition method (Chapter 5) with active learning using the popular query strategies. Additionally, feature selection is also integrated with active learning in Chapter 7 while performance of standard query strategies and random selection are examined which has not yet been investigated by any other study.

Chapter 3

A Filtering Mechanism for Normal Fish Trajectories

Unusual trajectories are generally defined as outliers or rare trajectories and outlier detection can be based on clustering. In this perspective, clusters with small numbers of elements are expected to represent rare trajectories and samples that are different from the other samples in the same cluster are considered as outliers [54]. On the other hand, many traditional clustering algorithms create clusters having similar amounts of samples and merge the small clusters to the closest cluster. Therefore, although clustering based outlier detection approach is reasonable, when the number of trajectories is huge like hundred thousands, millions etc. or the number of normal trajectories is much bigger than the number of unusual trajectories, such as 100 times bigger (or more), normal trajectories can dominate unusual trajectories and extracting small clusters and detecting outliers might be inaccurate. This might be even worse if classes contain sub-classes even though they are considered as the same class or sub-classes are overlapping.

With this assumption and considering the huge amount of data that the *Fish4Knowledge* repository has (see Chapter 1), in this chapter we present a rule based trajectory filtering mechanism to extract normal fish trajectories. The aim of this filtering mechanism is to *reject normal trajectories as much as possible (ideally all) while not rejecting any unusual trajectories.* This approach is very useful and very fast when scanning the large trajectory repository especially to filter out normal trajectories and detect possible unusual trajectories. This method was used to make the ground-truth data set for testing the other methods presented in Chapters 4 and 5. Moreover, this method is also not bad at detecting unusual fish trajectories (especially for the data sets presented in

Chapters 4 and 5).

3.1 Methodology

In this section, we give the definition of the fish trajectories and present the filtering mechanism. The fish trajectories are directly used by the proposed method in this chapter and multiple features are extracted from them to be used by the proposed methods in Chapters 4 and 5.

3.1.1 Trajectory Description

The tracker [19] gives the trajectories for fish moving across the image. For any fish i tracked through n frames, a trajectory is defined as the centre of fish bounding boxes as given in Eq. 3.1.

$$T_i = \{(x_{f_1}, y_{f_1}), (x_{f_2}, y_{f_2}), ..., (x_{f_n}, y_{f_n})\} \tag{3.1}$$

where (x, y) refers to the fish's position in the image (centre of fish bounding box) and f_n is the frame number in the corresponding video.

3.1.2 Proposed Filtering Mechanism

In Figure 3.1, the block diagram of the filtering mechanism is given. This procedure is like a cascade classifier. First, all fish trajectories are filtered by Filter 1. In each step, the trajectories satisfying the rule (filtered) are defined as normal trajectories (such as $Normal1$, $Normal2$ in Figure 3.1). The trajectories which do not satisfy the rule (not filtered) are called the remainders of the corresponding filter ($Remainder1$, $Remainder2$ in Figure 3.1) and are used as inputs to the following filter. This is continued until all the filters are used. At the end, the remainders of the last filter are called unusual trajectories.

3.1.3 Definition of Filters

Filters are defined in terms of the direction of the motion (left to right, right to left, up to down and down to up; which are also in terms of straight and/or cross motions as defined below) and/or being stationary (see the description given below). They are defined as one, two and three length combinations of the direction of the motion and

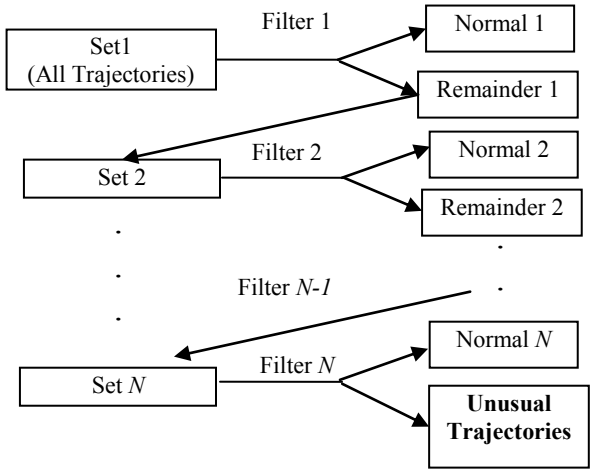


Figure 3.1: The block diagram of the rule based normal fish trajectory filtering method

being stationary such as moving right to left (length is one), moving right to left and then being stationary (length is two), moving left to right and then up to down (length is two), being stationary for a while, then moving down to up and then left to right (length is three), etc. Similar trajectories like going left to right and right to left are modelled by same filter. Altogether 21 rules were defined as given in Figure 3.2.

The filters given in Figure 3.2 are based on left to right, right to left, up to down, down to up and stationary. They are defined in terms of **straight and/or cross motions** (defined by Type 1, Type 2 and Type 3 as given below) or being **stationary** as follows. We assume that the origin of the image is the top-left corner.

- **Left to Right** (a,b)**:** $\forall\ (x_{f_i},y_{f_i})\ i=a$ to b: [$x_{f_i} \leq x_{f_{i+1}} \wedge$ (Type 1 $\vee$ Type 2 $\vee$ Type 3)] where f_a is the first frame number of the trajectory segment and b is the last frame number of that segment.

- **Right to Left** (a,b)**:** $\forall\ (x_{f_i},y_{f_i})\ i=a$ to b: [$x_{f_i} \geq x_{f_{i+1}} \wedge$ (Type 1 $\vee$ Type 2 $\vee$ Type 3)] where f_a is the first frame number of the trajectory segment and b is the last frame number of that segment.

- **Up to Down** (a,b)**:** $\forall\ (x_{f_i},y_{f_i})\ i=a$ to b: [$y_{f_i} \leq y_{f_{i+1}} \wedge$ (Type 1 $\vee$ Type 2 $\vee$ Type 3)] where f_a is the first frame number of the trajectory segment and b is the last frame number of that segment.

- **Down to Up** (a,b)**:** $\forall\ (x_{f_i},y_{f_i})\ i=a$ to b: [$y_{f_i} \geq y_{f_{i+1}} \wedge$ (Type 1 $\vee$ Type 2 $\vee$

Filter No	Description
1	Left to right *(1, N, p)* Right to left *(1, N, p)*
2	Up to down *(1, N, p)* Down to up *(1, N, p)*
3	Stationary *(1, N, p)*
4	Left to right *(1, a, p)*, stationary *(a, N, p)* Right to left *(1, a, p)*, stationary *(a, N, p)*
5	Left to right *(1, a, p)*, up to down *(a, N, p)* Left to right *(1, a, p)*, down to up *(a, N, p)* Right to left *(1, a, p)*, up to down *(a, N, p)* Right to left *(1, a, p)*, down to up *(a, N, p)*
6	Stationary *(1, a, p)*, left to right *(a, N, p)* Stationary *(1, a, p)*, right to left *(a, N, p)*
7	Stationary *(1, a, p)*, up to down *(a, N, p)* Stationary *(1, a, p)*, down to up *(a, N, p)*
8	Up to down *(1, a, p)*, left to right *(a, N, p)* Up to down *(1, a, p)*, right to left *(a, N, p)* Down to up *(1, a, p)*, right to left *(a, N, p)* Down to up *(1, a, p)*, left to right *(a, N, p)*
9	Up to down *(1, a, p)*, stationary *(a, N, p)* Down to up *(1, a, p)*, stationary *(a, N, p)*
10	Left to right *(1, a, p)*, stationary *(a, b, p)*, up to down *(b, N, p)* Left to right *(1, a, p)*, stationary *(a, b, p)*, down to up *(b, N, p)* Right to left *(1, a, p)*, stationary *(a, b, p)*, up to down *(b, N, p)* Right to left *(1, a, p)*, stationary *(a, b, p)*, down to up *(b, N, p)*
11	Left to right *(1, a, p)*, up to down *(a, b, p)*, stationary *(b, N, p)* Left to right *(1, a, p)*, down to up *(a, b, p)*, stationary *(b, N, p)* Right to left *(1, a, p)*, up to down (a, b, p), stationary *(b, N, p)* Right to left *(1, a, p)*, down to up (a, b, p), stationary *(b, N, p)*
12	Stationary *(1, a, p)*, left to right *(a, b, p)*, up to down *(b, N, p)* Stationary *(1, a, p)*, left to right *(a, b, p)*, down to up *(b, N, p)* Stationary *(1, a, p)*, right to left *(a, b, p)*, up to down *(b, N, p)* Stationary *(1, a, p)*, right to left *(a, b, p)*, down to up *(b, N, p)*
13	Stationary *(1, a, p)*, up to down *(a, b, p)*, left to right *(b, N, p)* Stationary *(1, a, p)*, up to down *(a, b, p)*, right to left *(b, N, p)* Stationary *(1, a, p)*, down to up *(a, b, p)*, left to right *(b, N, p)* Stationary *(1, a, p)*, down to up *(a, b, p)*, right to left *(b, N, p)*
14	Up to down *(1, a, p)*, stationary *(a, b, p)*, left to right *(b, N, p)* Up to down *(1, a, p)*, stationary *(a, b, p)*, right to left *(b, N, p)* Down to up *(1, a, p)*, stationary *(a, b, p)*, left to right *(b, N, p)* Down to up *(1, a, p)*, stationary *(a, b, p)*, right to left *(b, N, p)*
15	Up to down *(1, a, p)*, left to right *(a, b, p)*, stationary *(b, N, p)* Up to down *(1, a, p)*, right to left *(a, b, p)*, stationary *(b, N, p)* Down to up *(1, a, p)*, left to right *(a, b, p)*, stationary *(b, N, p)* Down to up *(1, a, p)*, right to left *(a, b, p)*, stationary *(b, N, p)*
16	Left to right *(1, a, p)*, stationary *(a, b, p)*, left to right *(b, N, p)* Left to right *(1, a, p)*, stationary *(a, b, p)*, right to left *(b, N, p)* Right to left *(1, a, p)*, stationary *(a, b, p)*, right to left *(b, N, p)* Right to left *(1, a, p)*, stationary *(a, b, p)*, left to right *(b, N, p)*
17	Left to right *(1, a, p)*, up to down *(a, b, p)*, left to right *(b, N, p)* Left to right *(1, a, p)*, up to down *(a, b, p)*, right to left *(b, N, p)* Left to right *(1, a, p)*, down to up *(a, b, p)*, left to right *(b, N, p)* Left to right *(1, a, p)*, down to up *(a, b, p)*, right to left *(b, N, p)* Right to left *(1, a, p)*, up to down *(a, b, p)*, right to left *(b, N, p)* Right to left *(1, a, p)*, up to down *(a, b, p)*, left to right *(b, N, p)* Right to left *(1, a, p)*, down to up *(a, b, p)*, right to left *(b, N, p)* Right to left *(1, a, p)*, down to up *(a, b, p)*, left to right *(b, N, p)*
18	Stationary *(1, a, p)*, left to right *(a, b, p)*, stationary *(b, N, p)* Stationary *(1, a, p)*, right to left *(a, b, p)*, stationary *(b, N, p)*
19	Stationary *(1, a, p)*, up to down *(a, b, p)*, stationary *(b, N, p)* Stationary *(1, a, p)*, down to up *(a, b, p)*, stationary *(b, N, p)*
20	Up to down *(1, a, p)*, stationary *(a, b, p)*, up to down *(b, N, p)* Up to down *(1, a, p)*, stationary *(a, b, p)*, down to up *(b, N, p)* Down to up *(1, a, p)*, stationary *(a, b, p)*, down to up *(b, N, p)* Down to up *(1, a, p)*, stationary *(a, b, p)*, up to down *(b, N, p)*
21	Up to down *(1, a, p)*, left to right *(a, b, p)*, up to down *(b, N, p)* Up to down *(1, a, p)*, left to right *(a, b, p)*, down to up *(b, N, p)* Up to down *(1, a, p)*, right to left *(a, b, p)*, up to down *(b, N, p)* Up to down *(1, a, p)*, right to left *(a, b, p)*, down to up *(b, N, p)* Down to up *(1, a, p)*, left to right *(a, b, p)*, up to down *(b, N, p)* Down to up *(1, a, p)*, left to right *(a, b, p)*, down to up *(b, N, p)* Down to up *(1, a, p)*, right to left *(a, b, p)*, up to down *(b, N, p)* Down to up *(1, a, p)*, right to left *(a, b, p)*, down to up *(b, N, p)*

Figure 3.2: The descriptions of 21 rules. 1 represent the first fish detection, a is the ath fish detection (the end of first segment that is also the beginning of the second trajectory segment), b refers to bth fish detection (the end of second segment that is also the beginning of the third trajectory segment), N is the last fish detection in the whole trajectory and p is the parameter that is used to define the search area for straight and/or cross motions and being stationary (see text for definition).

Type 3)] where f_a is the first frame number of the trajectory segment and b is the last frame number of that segment.

- **Stationary** (a,b)**:** $\forall\ (x_{f_i}, y_{f_i})$, $i = a+1$ to b $(x_{f_i} - x_{f_a})^2 + (y_{f_i} - x_{y_a})^2 \leq p^2$ where (x_{f_a}, y_{f_a}) is the centre of the first detection's bounding box for a given trajectory segment, b is the last frame number of that trajectory segment and p (is a parameter, see Section 3.3 for the pixel values used) is the radius of a circular search area where the centre is the first detection's bounding box. Figure 3.3d shows the circular area with the bounding boxes for a stationary segment. Being stationary is defined this way considering the fact that a fish cannot stay at the same point in most of the cases due to the sea currents.

The filters are defined in terms of 1 to 3 instances of left to right, right to left, up to down, down to up or being stationary conditions. A trajectory is analysed in terms of 1, 2 or 3 subsegments by using the filters with the above definitions but without considering the extra Type 1, Type 2 and Type 3 conditions (defined below). For a trajectory having two segments, the first segment should obey the first rule of the filter and the trajectory point $(a+1)$ that no longer obeys the first part of the filter defines the last point of first segment which is also the first point of the second segment (shown as a in the Figure 3.2). From trajectory point a, the rule in the second part of the filter should be obeyed through to the end of the trajectory. For a trajectory having three segments, the trajectory point $(a+1)$ that no longer obeys the rule of the first segment of the filter defines the last point of the first segment which is also the first point of the second segment (shown as a in the Figure 3.2) and the trajectory point $(b+1)$ that no longer obeys the rule of the second segment of the filter defines the last point of the second segment which is also the first point of the third segment (shown as b in the Figure 3.2). After the trajectory is segmented in terms of the basic motions, the non-stationary segments are checked if they satisfy at least one of these rules:

- **Type 1:** All centres of the fish bounding boxes over the given trajectory segment are inside a rectangular area (search area) that is determined by the centre of the first bounding box of that segment. There are 4 types of area depending on the type of the segment (Figure 3.3a shows the rectangular area with the bounding boxes for a left to right segment). The corners of the rectangular area are:
  - Left to Right: [$(x_{f_a}, y_{f_a} - p)$, $(x_{f_a} + p, y_{f_a} - p)$, $(x_{f_a}, y_{f_a} + p)$, $(x_{f_a} + p, y_{f_a} + p)$]

  - Right to Left: [$(x_{f_a}, y_{f_a} - p)$, $(x_{f_a} - p, y_{f_a} - p)$, $(x_{f_a} - p, y_{f_a} + p)$, $(x_{f_a}, y_{f_a} + p)$]
  - Up to Down: [$(x_{f_a} - p, y_{f_a})$, $(x_{f_a} - p, y_{f_a} + p)$, $(x_{f_a} + p, y_{f_a} + p)$, $(x_{f_a} + p, y_{f_a})$]
  - Down to Up: [$(x_{f_a} - p, y_{f_a} - p)$, $(x_{f_a} - p, y_{f_a})$, $(x_{f_a} + p, y_{f_a})$, $(x_{f_a} + p, y_{f_a} - p)$]

  where p is a parameter (see Section 3.3 for the pixel values used) and (x_{f_a}, y_{f_a}) is the first detection's bounding box centre in that segment. These search areas are illustrated in Figure 3.4a.

- **Type 2:** The centre of the fish bounding box in frame f_i is inside a rectangular area (search area) which is determined by the detection bounding box in frame f_{i-1} while the fish is going only one direction in that segment of the trajectory (Figure 3.3b shows the rectangular areas with the bounding boxes for a right to left segment). This rule is similar to the previous rule. The corners of the rectangular area are:

  - Left to Right: [$(x_{f_{i-1}}, y_{f_{i-1}} - p)$, $(x_{f_{i-1}} + p, y_{f_{i-1}} - p)$, $(x_{f_{i-1}}, y_{f_{i-1}} + p)$, $(x_{f_{i-1}} + p, y_{f_{i-1}} + p)$]
  - Right to Left: [$(x_{f_{i-1}}, y_{f_{i-1}} - p)$, $(x_{f_{i-1}} - p, y_{f_{i-1}} - p)$, $(x_{f_{i-1}} - p, y_{f_{i-1}} + p)$, $(x_{f_{i-1}}, y_{f_{i-1}} + p)$]
  - Up to Down: [$(x_{f_{i-1}} - p, y_{f_{i-1}})$, $(x_{f_{i-1}} - p, y_{f_{i-1}} + p)$, $(x_{f_{i-1}} + p, y_{f_{i-1}} + p)$, $(x_{f_{i-1}} + p, y_{f_{i-1}})$]
  - Down to Up: [$(x_{f_{i-1}} - p, y_{f_{i-1}} - p)$, $(x_{f_{i-1}} - p, y_{f_{i-1}})$, $(x_{f_{i-1}} + p, y_{f_{i-1}})$, $(x_{f_{i-1}} + p, y_{f_{i-1}} - p)$]

  where p is a parameter (see Section 3.3 for the pixel values used) and $(x_{f_{i-1}}, y_{f_{i-1}})$ is the previous detection's bounding box centre in that segment. These search areas are illustrated in Figure 3.4a.

- **Type 3:** The centres of the fish bounding boxes over the given trajectory segment are inside a rectangular area (search area) which is determined by the centres of first and last detection's bounding box boundaries in that trajectory segment while fish is going only one direction (Figure 3.3c shows the rectangular area with the bounding boxes which is for an up to down segment). The corners of the rectangular area are:

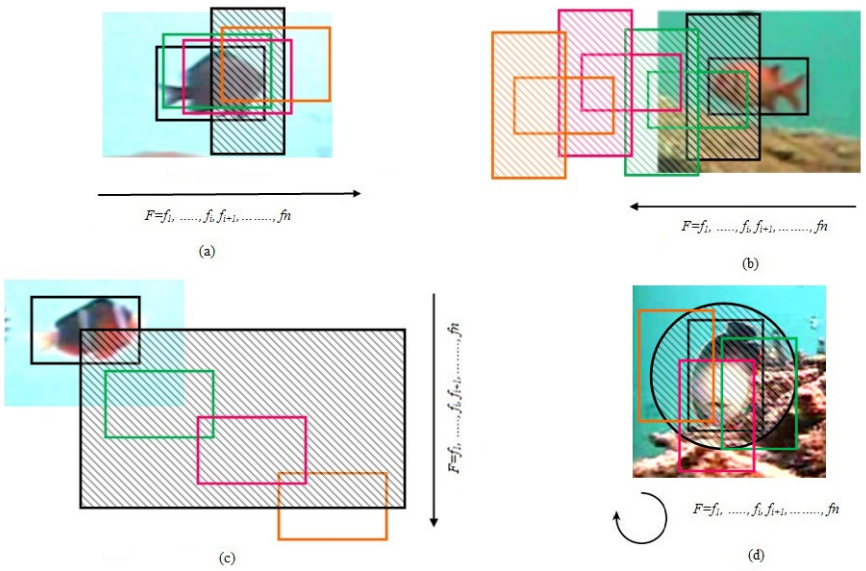


Figure 3.3: Example illustrations for straight and/or cross movements: a) Type 1 for left to right, b) Type 2 for right to left, c) Type 3 for up to down and d) being stationary. From the first to the last detection the bounding boxes are shown with black, green, pink and orange, respectively. The shaded areas are the search areas which is rectangular for Type 1 (shown as black shaded), Type 2 (shown with the same colour of shading with the corresponding bounding box), Type 3 (shown as black shaded) and circular for being stationary (shown as black shaded).

- Left to Right or Right to Left and $y_{f_a} < y_{f_b}$: [(x_{f_a}, y_{f_a}), (x_{f_b}, y_{f_a}), $(x_{f_b}, y_{f_b} + p)$, $(x_{f_a}, y_{f_b} + p)$]
- Left to Right or Right to Left and $y_{f_a} > y_{f_b}$: [(x_{f_a}, y_{f_a}), (x_{f_b}, y_{f_a}), $(x_{f_b}, y_{f_b} - p)$, $(x_{f_a}, y_{f_b} - p)$]
- Up to Down or Down to Up and $x_{f_a} < x_{f_b}$: [(x_{f_a}, y_{f_a}), (x_{f_a}, y_{f_b}), $(x_{f_b} + p, y_{f_b})$, $(x_{f_b} + p, y_{f_a})$]
- Up to Down or Down to Up and $x_{f_a} > x_{f_b}$: [(x_{f_a}, y_{f_a}), $(x_{f_b} - p, y_{f_a})$, $(x_{f_b} - p, y_{f_b})$, (x_{f_a}, y_{f_b})]

where p is a parameter (see Section 3.3 for the pixel values used), (x_{f_a}, y_{f_a}) is the first detection's bounding box centre and (x_{f_b}, y_{f_b}) is the last detection's bounding box centre in that segment. These search areas are illustrated in Figure 3.4b.

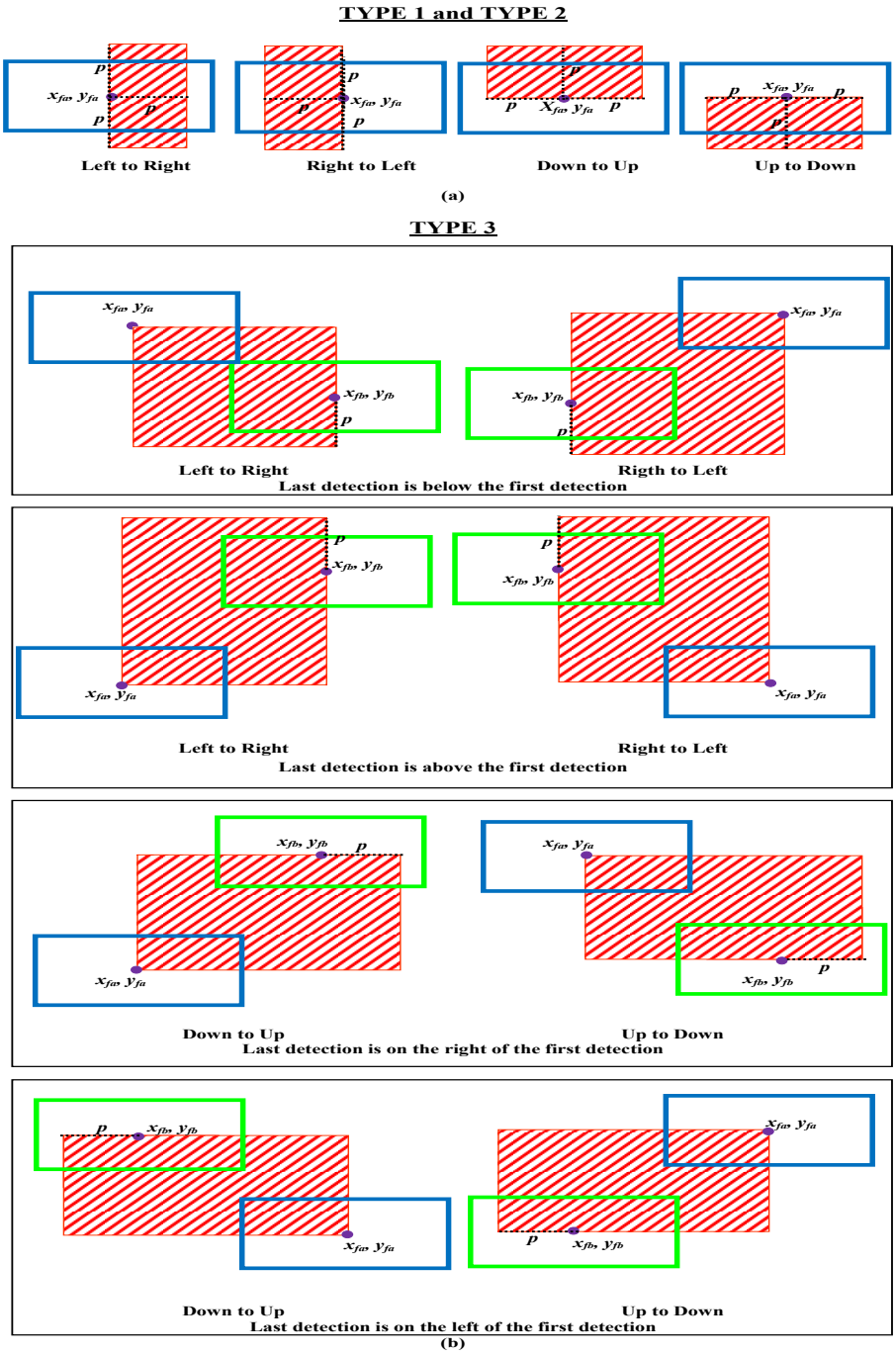


Figure 3.4: Illustration of the search area (red shaded area) for each type of the straight and/or cross movements. p is a parameter that defines the search area. a) For Type 1 (x_{f_a}, y_{f_a}) is the centre of the first detection and for Type 2 it is the centre of every detection from $i = 2$ to b where b is the end of that trajectory segment. b) For Type 3, (x_{f_a}, y_{f_a}) is the centre of the first detection (bounding box is shown with blue) and (x_{f_b}, y_{f_b}) is the centre of the last detection (bounding box is shown with green) of a given trajectory segment.

3.2 Data Set

The proposed method was tested using 271 sample underwater videos (320x240 resolution, 5 frames per second) from the *Fish4Knowledge* repository which includes 4 different locations and 2486 trajectories (46 unusual, 2440 normal) belonging to 10 different species. The normal and unusual trajectories are determined based on visual inspection. In this context, freely swimming fish were considered as **normal trajectory** since this is the most frequent behaviour in the data set. In this data set, the **unusual trajectories** were: *i)* fish stationary for a long time (compared to the detection length) inside of coral: this kind of a behaviour is assumed to be an eating behaviour hence differentiated from swimming, *ii)* biting at coral (also interaction with plankton, Figure 3.5c), *iii)* fish suddenly (usually in one frame) diving (Figure 3.5d), *iv)* fish suddenly (usually in one frame) changing direction, *v)* fish turning around in an area like a predator. Example normal (blue) and unusual trajectories (red) in this data set are shown for four camera locations in Figure 3.6 to give a clearer idea about the data set. Different normal and unusual behaviours can be observed in each location. The fish trajectories are complex compared to other trajectory problems (e.g. pedestrians, vehicles, etc.). For instance, there are no well defined clusters of trajectories which often exist in a pedestrian scenario as pedestrians use similar paths to walk. However, fish can appear anywhere in the underwater. Additionally, normal and unusual trajectories are overlapping in terms of trajectory points.

The data set used in this chapter is different than the data sets used in Chapters 4 and 5. This data set covers many fish species and 4 different camera locations. However, the data sets used in Chapters 4 and 5 belong to a single species (*Dascyllus reticulatus*) from a single camera location as we found that the fish behaviour might change from species to species (there are specific behaviours which belong to specific fish species) and can be affected by the geographic properties of the underwater environment. On the other hand, the method proposed in this chapter will be compared with the methods presented in Chapters 4 and 5 using the data sets presented in those chapters (see Chapters 4 and 5 for results).

3.3 Experimental Work

To evaluate the proposed filtering mechanism a 9-fold cross validation was performed. Train (8/9) and test (1/9) sets were constituted randomly where the normal and unusual

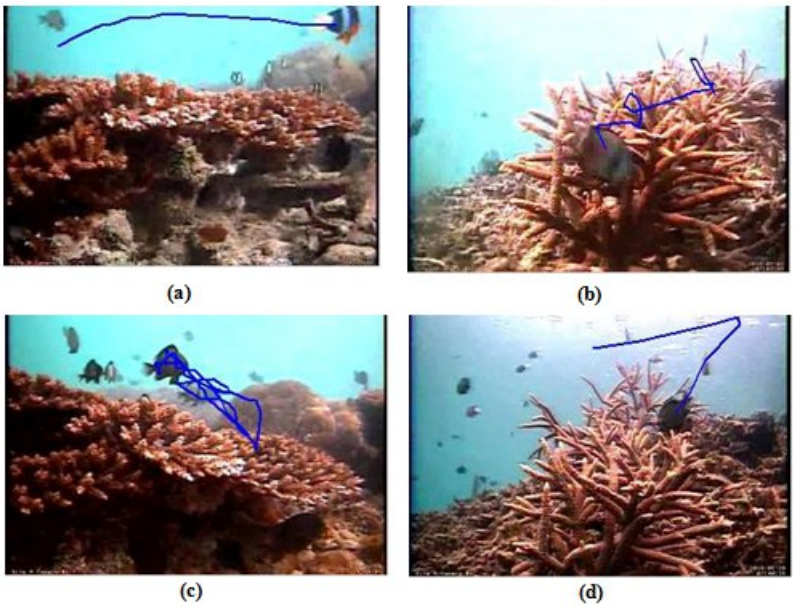


Figure 3.5: (a-b) Examples of normal fish trajectories which are classified by the proposed method, (c-d) Examples of unusual fish trajectories.

trajectories were distributed equally.

During training, for each filter the best parameter (given as p in Section 3.1.3) which defines the search area is found. The best p value for each filter is the one which does not filter out any unusual trajectories. In the case of having more than one p value which does not filter out any unusual trajectories, the one that filtered the most normal trajectories is selected. If there are no p values which do not filter out any unusual trajectories, then that filter is not used and the process continues with the following filter. In this chapter, the parameter p was used as $\{2, 4, 8, 10, 16 \text{ and } 20\}$ pixels. The best p value of each filter can be different.

During testing, the filters with the best p values (found during training) are used to classify new trajectories. Filters that were removed during training are not used during testing.

3.4 Results

The performance of the proposed method is given in Table 3.1 with the average TP, TN, FP, FN values (see Section 2.3.1 for descriptions) and the standard deviations considering cross validation folds (after the $\pm$ sign). In this section, the positive class represents the unusual trajectories and the negative class represents the normal

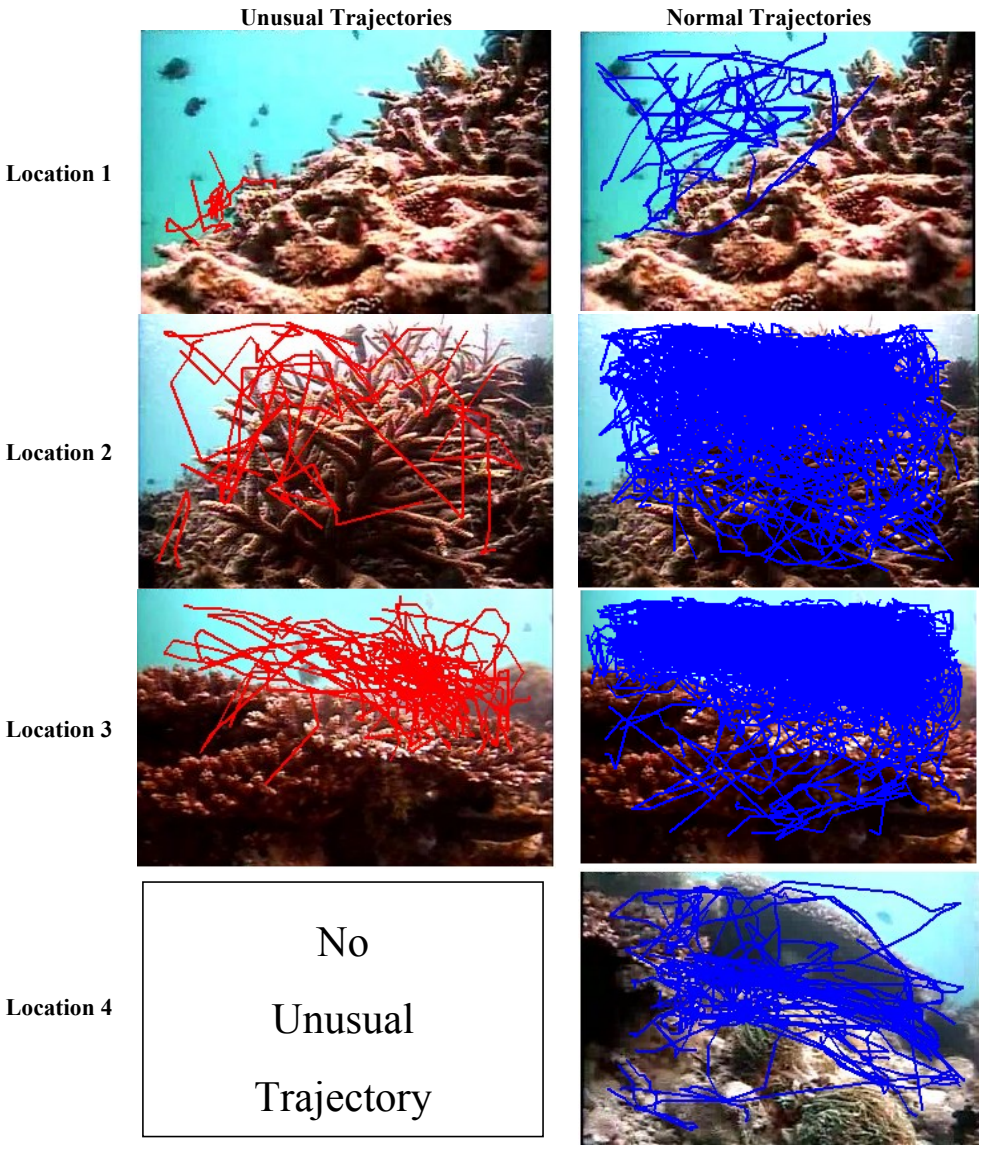


Figure 3.6: Example unusual (red) and normal (blue) trajectories for 4 different camera locations

Table 3.1: Performance of the rule based normal fish trajectory filtering method

	Prediction as Normal (Filtered)	**Prediction as Unusual (Maintained)**
Normal	101.78 ±9.24	169.33 ±9.30
Unusual	0.67 ±0.87	4.44 ±1.01

trajectories to be consistent with the evaluations given in the following chapters.

The results show that 38% of normal trajectories were detected by the filtering mechanism with 99% precision while 13% of the unusual trajectories were also detected and filtered out as normal trajectories (which ideally should have been zero). However, we believe that this is still a good result since data set used contains different fish species and camera locations which cause more variation in the fish behaviour. Additionally, filters and parameters are defined without considering the type of unusual trajectories to propose a general mechanism which is independent of the data.

3.4.1 Comparison with a State of Art Method

The proposed method is also compared with method [18] since it is the most applicable/similar study that can be compared. When applying that method, first trajectories are linearly interpolated to fill the gaps between detections. Then the Douglas-Peucker algorithm is applied to reduce the number of points that each trajectory has. When applying Douglas-Peucker algorithm, it is assumed that the maximal Euclidean distance allowed between the new line and a vertex is one. The trajectories were clustered using k-means with the number of clusters from 2 to 150. The unusual trajectories were determined by being in a small cluster. A small cluster has fewer trajectories than the mean – one standard deviation of the all cluster cardinalities. The best result of that method [18] was obtained when the number of clusters is 142 which gave the mean and the standard deviation of the cluster cardinalities as 17.51 and 8.89, respectively.

The *GeoMean* (see Section 2.3.1 for description) was used as the evaluation metric to compare the proposed method with method [18]. According to the results in Table 3.1 the proposed method has 0.57 *GeoMean* while method [18] has 0.29 *GeoMean*. The paired t-test (α=0.05) using the *GeoMean* results also showed that the proposed method presents significantly better results compared to [18].

The poor results of method [18] also showed that it is hard to distinguish normal

and unusual fish trajectories using trajectory points only as there are many overlapping descriptions between the two classes. This motivated us to define other features which are extracted from trajectories such as velocity and shape based features and to propose the methods given in Chapters 4 and 5.

3.5 Conclusions

As a conclusion, the proposed rule based filtering method is the first algorithm for filtering normal fish trajectories in an unconstrained open sea environment. It is successful at filtering out many normal trajectories while filtering out only a few unusual trajectories (ideally should be zero). This method has been used as a preliminary method to collect ground truth data especially unusual trajectories (remember that the aim is to reject normal trajectories as much as possible while not rejecting any unusual trajectories) thanks to being fast and having low false negative rate (0.13 corresponds to the *GeoMean* given above which is for the data set used in this chapter).

As future work, this method can be combined with any unusual fish trajectory detection method which might increase the detection performance. It can be applied especially when the number of normal fish trajectories is much greater than the number of unusual fish trajectories (to make the data set less imbalanced) or when the number of trajectories is very large (to decrease the amount of data if the unusual trajectory detection method used is not scalable).

Chapter 4

Detecting Unusual Fish Trajectories Using Clustered and Labelled Data (Flat Classifier)

This chapter presents an approach to detecting unusual fish trajectories using multiple features which are extracted from the fish trajectories. The proposed method is mainly based on clustering. An outlier detection method based on the sample size of the clusters and a distance function is applied to each cluster to find the unusual trajectories. Clustered and labelled data are used together to select the best feature set (that provides the best classification performance) during training. The learned feature set and the outlier detection parameters are used to classify the new fish trajectories. For the rest of this thesis we refer to this proposed method as the **flat classifier**.

This chapter presents two innovations:

- A novel approach to unusual trajectory detection,
- Improved performance especially compared to the proposed method in Chapter 3 on unusual fish trajectory detection in unconstrained conditions.

The obtained improved results are significant considering the challenges of underwater environments, low video quality, and erratic movement of fish. The proposed method also presents the foundation for the method presented in Chapter 5.

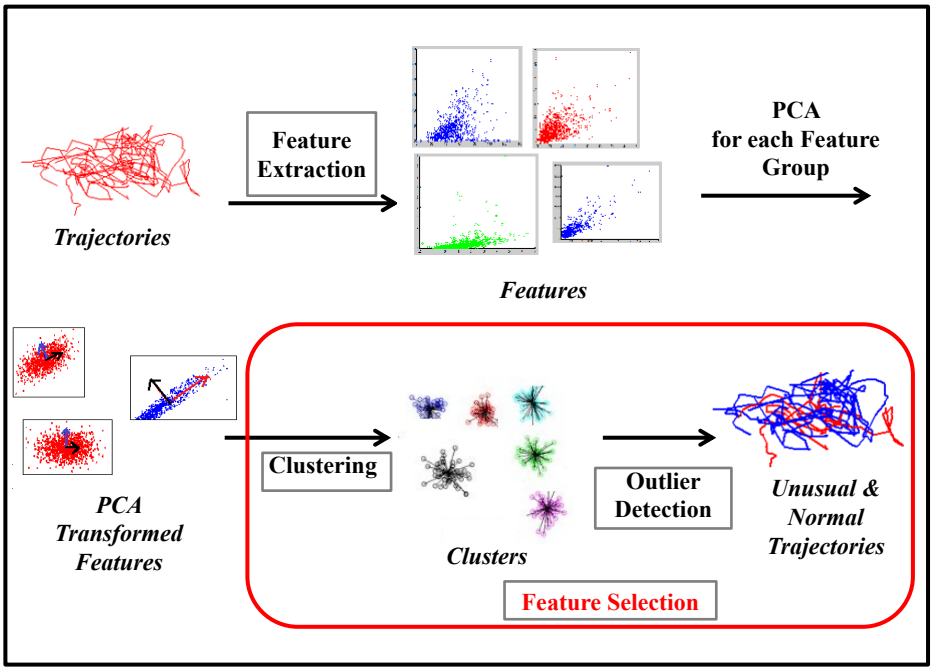


Figure 4.1: Overview of the flat classifier

4.1 Methodology

The proposed method contains four steps:

1. Feature extraction (including the pre-processing of the trajectory and Principal Component Analysis (PCA) of extracted features),
2. Clustering,
3. Outlier detection,
4. Feature selection which is embedded in the clustering and outlier detection.

The overview of the proposed method is given in Figure 4.1.

First, features from fish trajectories are extracted. Then, feature selection is applied using a training set. Feature selection is evaluated using clustering and outlier detection. For each set of features, clustering and outlier detection are applied to detect the outliers. The best set of features (with the best outlier detection parameter) which are chosen during training are used to classify new fish trajectories as normal or unusual.

4.1.1 Feature Extraction

The challenges of fish detection and tracking in the underwater environment such as sudden light changes, bad weather conditions (e.g. storms, typhoons), murky water and multiple fish occlusions [19] sometimes cause gaps in the fish trajectory. To handle this, before extracting features, all trajectories are linearly interpolated (see trajectory definition in Section 3.1). Then 10 groups of features as given below are extracted. In total, 776 features are obtained in the feature extraction step. These features are generally correlated with each other. Therefore to prevent possible over-training and the curse of dimensionality, after normalising the features, Principal Component Analysis (PCA) is applied to each group of features individually. Here, it should be stated that applying PCA to each feature groups individually provided better performance compared to applying PCA to all features. This reduce the dimensionality of the data and also remove the correlations between features. While applying PCA, to obtain a useful set of components the smallest number of components that represent 90% of the sum of all eigenvalues is used. As a result of applying PCA to for the data set given in Section 4.2, 140 features are obtained as the feature set. The extracted features are defined as follows:

4.1.1.1 Curvature Scale Space (CSS) Based Features

As a trajectory representation CSS was first introduced in [158]. CSS is a multi resolution technique which is calculated using the curvature at every point on the curve by the formula given in Eq. 4.1. This trajectory description is shaped based, rotation and translation invariant. The curvature at point f_i is calculated using:

$$K_{f_i} = \frac{x'_{f_i} y''_{f_i} - y'_{f_i} x''_{f_i}}{\left(x'^2_{f_i} + y'^2_{f_i}\right)^{3/2}} \tag{4.1}$$

where (x, y) refers to the fish's positions in an image (centre of fish bounding box) and f_i is the frame number.

To find the CSS, a Gaussian kernel is used. At each level of space the standard deviation (σ) of Gaussian kernel is increased, the (x_{f_i}, y_{f_i}) are smoothed and the curvature of that level is found. The CSS is represented with a binary image which is called a CSS image. In that image, white pixels represents the zero crossings at each scale level. As σ increases, the trajectory shrinks, the curve becomes smoother and zero crossing points on CSS image decreases. At the end, the curve becomes convex with

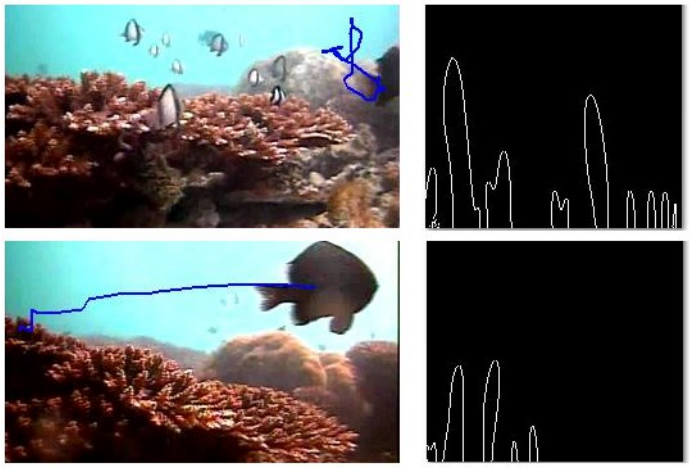


Figure 4.2: Example trajectories (left) and corresponding CSS images (right)

no zero crossings. Example fish trajectories with corresponding CSS images are given in Figure 4.2.

Statistical properties (mean and variance of length of the curves, number of zero crossings for each σ, total number of curves, mean and variance of σ in peak points, mean and variance of σ at starting points of each curve) are extracted from CSS image to use as features. Additionally, for each σ value, statistical features of absolute curvature are extracted. In our experiments σ was taken from one to 20 while increasing it with 0.1. In total 580 features are obtained.

4.1.1.2 Moment Descriptors Based Features

Moment invariants are well known, successful descriptors for recognising objects and patterns and can be used to distinguish the shape of fish trajectories. Therefore, we utilise the affine moment invariants proposed in [159] in addition to moments (Eq. 4.2), central moments (Eq. 4.3) and translation and scale invariant moments (Eq. 4.4). In total 55 features (10 from the affine moment invariants, 15 from each of moments, central moments and translation and scale invariant moments where the moment order was taken up to 4) are extracted from those moment descriptors.

$$M_{pq} = \sum_{i}^{n} {x_{f_i}}^p {y_{f_i}}^q \tag{4.2}$$

$$\mu_{pq} = \sum_{i}^{n} (x_{f_i} - x_c)^p (y_{f_i} - y_c)^q \text{ where } c_x = \frac{1}{n}\sum_{i=1}^{n} x_{f_i} \text{ and } c_y = \frac{1}{n}\sum_{i=1}^{n} y_{f_i}. \tag{4.3}$$

$$n_{pq} = \frac{\mu_{pq}}{\mu_{pq}^{1+\frac{p+q}{2}}} \text{ for } p, q=0, 1, 2... \tag{4.4}$$

where p and q are order of moment over trajectory point (x_{f_i}, y_{f_i}) through the trajectory length n.

4.1.1.3 Velocity and Acceleration Based Features

Even though a fish trajectory is spatially similar to normal trajectories due to its speed and/or speed change, it may be an unusual trajectory. Therefore using velocity and acceleration based features can be useful.

Statistical properties: mean, standard deviation, minimum, maximum, number of zero crossings, number of local minima and maxima of velocity and acceleration are extracted in three dimensions considering the fact that fish can swim in three dimensions in an open sea. Since the trajectory description in the *Fish4Knowledge* repository is in two dimensions, we estimate the position in the third dimension using the width (w_{f_i}) and height (h_{f_i}) of the fish detection bounding box at frame f_i using the formula given in Eq. 4.5. In total 42 features (7 statistical properties $\times$ 3 dimensions $\times$ 2; one for velocity and other for acceleration) are obtained.

$$z_{f_i} = \frac{1}{\sqrt{w_{f_i} h_{f_i}}} \tag{4.5}$$

4.1.1.4 Turn Based Features

Trajectory turning at frame i is defined as the orientation of the trajectory between consecutive trajectory points, which is calculated as given in Eq. 4.6 [54, 160]. It can be used to describe a fish trajectory in terms of its shape. Statistical properties are extracted from the trajectory turning values: mean, standard deviation, minimum, maximum, number of zero crossings, number of local minima and maxima. In total 7 features are obtained.

$$\theta_i = \begin{cases} \arctan\left\{\frac{y_{f_{i+1}} - y_{f_i}}{x_{f_{i+1}} - x_{f_i}}\right\}, & (x_{f_{i+1}} - x_{f_i}) > 0 \\ \arctan\left\{\frac{y_{f_{i+1}} - y_{f_i}}{x_{f_{i+1}} - x_{f_i}}\right\} + 360, & (x_{f_{i+1}} - x_{f_i}) \leq 0, (y_{f_{i+1}} - y_{f_i}) \geq 0 \\ \arctan\left\{\frac{y_{f_{i+1}} - y_{f_i}}{x_{f_{i+1}} - x_{f_i}}\right\} - 360, & (x_{f_{i+1}} - x_{f_i}) \leq 0, (y_{f_{i+1}} - y_{f_i}) < 0 \end{cases} \tag{4.6}$$

where $(x_{f_{i+1}} - x_{f_i})^2 + (y_{f_{i+1}} - y_{f_i})^2 \neq 0$ and $\theta_i \in$ [-360, +360).

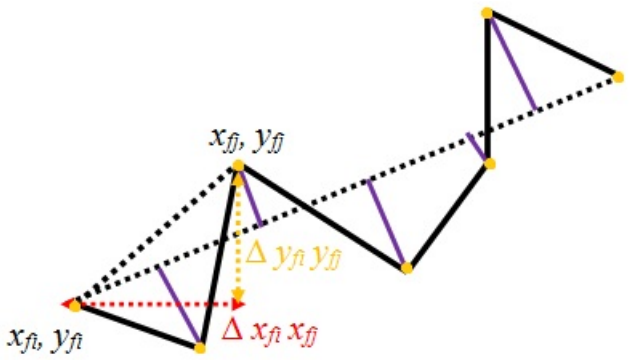


Figure 4.3: Properties of the vicinity (adapted from [1])

4.1.1.5 Centred Distance Function (CDF)

CDF is an invariant shape descriptor that gives the distance of each point in a trajectory from the centre of the trajectory (Eq. 4.7, [158]). As features the statistical properties (mean, maximum, minimum, standard deviation, number of mean crossings, number of local minima and maxima, skewness and kurtosis) are extracted from two and three dimensional CDFs to describe trajectories. In total 18 features (9 from 2D CDF and 9 from 3D CDF) are defined.

$$\begin{aligned} cdf\_2D(f_i) &= \sqrt{(x_{f_i}-x_c)^2+(y_{f_i}-y_c)^2} \\ cdf\_3D(f_i) &= \sqrt{(x_{f_i}-x_c)^2+(y_{f_i}-y_c)^2+(z_{f_i}-z_c)^2} \\ f_i &= 1,2,...n \end{aligned} \tag{4.7}$$

where x_c and y_c are as given in Eq. 4.3 while z_c is similar to them, and z_{f_i} is as given in Eq. 4.5.

4.1.1.6 Vicinity Features

Properties extracted from the trajectory vicinity were introduced in [1] for handwriting recognition but to the best of our knowledge they were never used to represent other kinds of trajectories. We adapted this description to represent fish trajectories. In Figure 4.3, the properties of the vicinity $\{\Delta x_{f_i} x_{f_i}, \Delta y_{f_j} y_{f_j}\}$ for the points (x_{f_i}, y_{f_i}) and (x_{f_j}, y_{f_j}) where $j = i+2$ are given. In our experiments we extracted these properties for each group of three consecutive points (such as (x_{f_1}, y_{f_1}), (x_{f_2}, y_{f_2}), (x_{f_3}, y_{f_3})) along the complete trajectory.

Three different group of properties [1]:

- Aspect of vicinity: Different than [1], this is defined in two ways as given in Eq. 4.8,

$$Type1 : \frac{\Delta y_{f_i} y_{f_j} - \Delta x_{f_i} x_{f_j}}{\Delta y_{f_i} y_{f_j} + \Delta x_{f_i} x_{f_j}} \qquad Type2 : \frac{\Delta y_{f_i} y_{f_j}}{\Delta x_{f_i} x_{f_j}} \tag{4.8}$$

  for the points (x_{f_i}, y_{f_i}) to (x_{f_j}, y_{f_j}).

- Vicinity curliness: The length of the trajectory from point (x_{f_i}, y_{f_i}) to (x_{f_j}, y_{f_j}) in the vicinity divided by maximum $\{\Delta x_{f_i} x_{f_j}, \Delta y_{f_i} y_{f_j}\}$,

- Vicinity linearity: the average square distance of each point in the vicinity to the straight line from the last and the first vicinity point (shown with purple lines in Figure 4.3)

are extracted to define the shape of the fish trajectories.

Statistical measures are extracted from those properties including mean, standard deviation, skewness, kurtosis, number of mean crossings, number of local minima, number of local maxima, maximum, minimum, median. In total, 40 features (10 measures for each group of aspect of vicinity with type1, aspect of vicinity with type2, vicinity curliness and vicinity linearity) are obtained from trajectory vicinity.

4.1.1.7 Loop Features

Due to the erratic motion of fish and the currents in the undersea, fish trajectories are generally very complex and contain many loops. Motivated by this, fish trajectories are described by the number of loops, maximum, minimum and median of number of points in a loop. The existence of a loop in a trajectory is found as illustrated in Figure 4.4. Reaching the common point (shown with purple) from any point in the loop (such as the point shown as rounded black) through to the final point of the trajectory (shown as rounded red) and from that black point through to the starting point of the trajectory (shown as rounded green) determines a loop. Since it is hard to detect the fish exactly in the same place twice or more, the common point is detected by obtaining two lines one from any two consecutive points (such as while going from a point to the starting point) and another from any other two consecutive points (such as while going from a point to the end of the trajectory). Then, the possible intersection of them is checked. If there is an intersection point and that point is between the points that form those lines

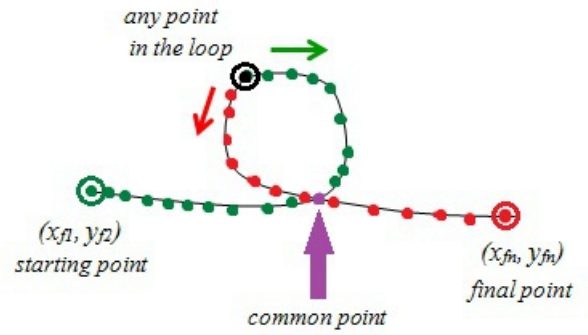


Figure 4.4: A trajectory with a loop.

Figure 4.5: Example fish trajectories with loops.

(while not one of those points), this makes it the common point. In total, 4 features are obtained. Some example images with fish trajectories having loops are given in Figure 4.5.

4.1.1.8 Fish Pass by Features

Fish trajectories are affected by the geographical properties of the underwater environment and their trajectories can be different in different locations. Therefore, while finding normal and unusual trajectories those properties can be useful to consider. In this thesis, we divide the underwater environment into three areas: open sea, under the coral and above the coral (Figure 4.6). We manually segmented each video scene once and utilise segmentations to obtain the features corresponding to all fish trajectories of a video. As features, the percentage of time being in different locations and the percentage of time crossings from one location to another are considered. In total 12 features are obtained: the percentage of time in the open sea, the percentage of time under the coral, the percentage of time above the coral, the percentage of time crossing from under the coral to under the coral, the percentage of time crossing from under the coral to above the coral, the percentage of time crossing from under the coral to

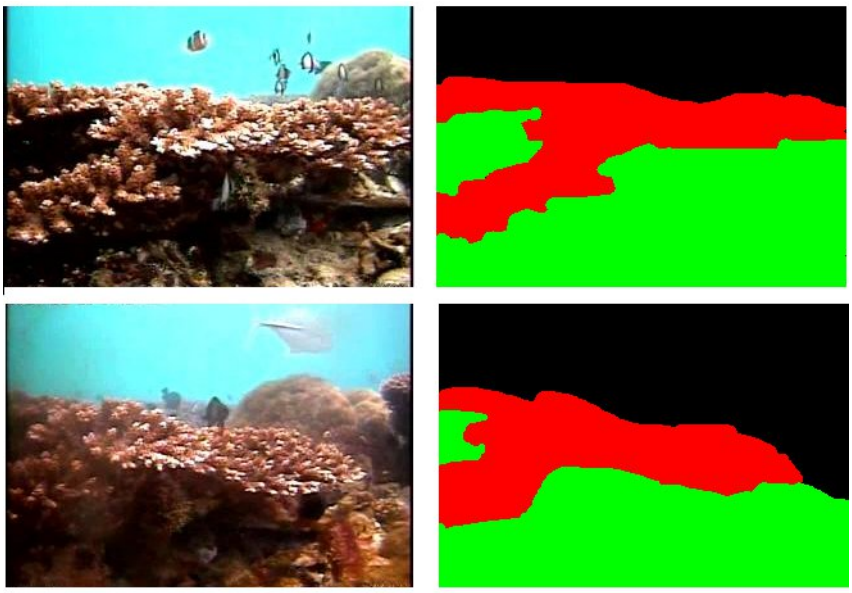


Figure 4.6: Segmented regions of the underwater image; black for open sea, red for above the coral and green for under coral

the open sea, the percentage of time crossing from above the coral to above the coral, the percentage of time crossing from above the coral to under the coral, the percentage of time crossing from above the coral to the open sea, the percentage of time crossing from the open sea to under the coral, the percentage of time crossing from the open sea to the open sea and the percentage of time crossing from the open sea to above the coral.

4.1.1.9 Features Based on Displacement on the Location

Using the segmented locations given in Figure 4.6, statistical properties: mean, maximum, minimum, standard deviation, and median of average displacement in different locations are found to define trajectories. In total, 15 features (5 statistical properties for 3 locations) are obtained from this description such as maximum average displacement in the open sea, standard deviation of average displacement in under the coral.

4.1.1.10 Features Based on Normalised Size of Bounding Box

Fish can frequently change their posture (even for adjacent frames). To distinguish the erratic random motions, aggressive motions and sudden movement of a fish, its posture can be used. To model this, a function using the ratio of width of the fish bounding box (w_{f_i}) to its height (h_{f_i}) at each fish detection (f_i) is defined. This func-

Table 4.1: The number of extracted features before and after *PCA* for the fish trajectory data set given in Section 4.2.

Feature Group	**# of features before PCA**	**# of features after PCA**
CSS Based Features	580	76
Moment Descriptors Based Features	55	13
Velocity and Acceleration Based Features	42	12
Turn Based Features	7	3
Centred Distance Function (CDF)	18	8
Vicinity Features	40	13
Loop Features	4	2
Fish Pass by Features	12	6
Features Based on Displacement on the Location	15	5
Features Based on Normalised Size of Bounding Box	3	2
Total	776	140

tion is z-normalised to eliminate the effect of small and big fish differences. By using this function, as features: the number of one crossings (i.e. from values smaller than one to values bigger or equal to one or vice versa), number of local minima and number of local maxima are extracted. In total 3 features are extracted.

To sum up, the total number of features for each group before and after *PCA* are given in Table 4.1 for the fish trajectory data set given in Section 4.2.

4.1.2 Clustering

For clustering, Affinity Propagation (AP) [161] is used. AP has been applied as a clustering method in various studies including anomaly detection.

AP selects the cluster centres from the actual data points which are called cluster exemplars. The method uses the pair-wise similarity of each pair of data points which is the negative of the Euclidean distance between these points. The objective function of AP tries to find the exemplars that maximise the overall sum of similarities between

all exemplars and their data points given the similarity matrix. There are two kinds of messages between data points [161]:

- Responsibility (r): It is from data point i to data point j that represents the accumulated evidence for how appropriate it would be for data point j to be the exemplar for data point i.
- Availability (a): It represents how appropriate it would be for data point i to choose data point j as its exemplar.

At the beginning, the availabilities are zero ($a(i,k) = 0$) and the responsibilities are calculated as given in Eq. 4.9.

$$r(i,j) = s(i,j) - max_{j' \neq j}\{a(i,j') + s(i,j')\} \tag{4.9}$$

where r refers to responsibility, a refers to availability, i, j are data points and s is the similarity.

For the later iterations, as some data points' exemplars are found, the availabilities are decreased by using the formula given in Eq. 4.10.

$$a(i,j) = min\Big\{0, r(j,j) + \sum_{i' \notin \{i,j\}} max\{0, r(i',j)\}\Big\} \tag{4.10}$$

This message-passing procedure is terminated when *i)* a fixed number of iterations is reached, *ii)* the value of messages becomes lower than a threshold or *iii)* after the exemplars and data points stay constant for a certain number of iterations. In our study, we apply the last strategy by taking the maximum number of iterations as 4000.

AP has many advantages over traditional clustering methods such as its fast processing speed, being non-parametric (different than k-means), not requiring initialisation (different than SOM), not depending on sample order (different than hierarchical clustering) and scalability (which makes our methods scalable as well). However, in our case the main reasons for using this method are its ability:

- to produce smaller clusters,
- to produce uneven sized clusters with minimum error rate [161]

which are compatible with the outlier detection method (Section 4.1.3) that we use.

4.1.3 Outlier Detection

An outlier is defined as a datum which is distant from other data points in the same cluster. Most of the time, the cardinality of the outliers is smaller than the other data points in the same cluster. On the other hand, an unusual trajectory can be defined as one that deviates from other trajectories in its cluster or the one which builds a cluster with a few other unusual trajectories.

In this thesis, we adapted the outlier detection method from [54] and use it to detect unusual fish trajectories. We assume two types of outliers:

1. Those located in small clusters,

2. Those in dense clusters but far from cluster exemplars.

To detect the small and dense clusters, a threshold is defined based on the cardinality of all clusters. A cluster which has fewer trajectories (data samples) than 10% of the median cardinality of clusters or a cluster that has only one trajectory (data point) is defined as a **small cluster**. All trajectories that belong to such a cluster are classified as **unusual trajectory**. Otherwise, the cluster is a **dense cluster**, and outliers are detected using the Euclidean distance between the trajectory features and the cluster exemplar. Small clusters are illustrated as the clusters having boundaries with thick lines while dense clusters are the clusters having boundaries with dashed lines in Figure 4.7. In dense clusters, a trajectory which is far away compared to threshold $\tau = \mu + w\sigma$ (with mean (μ), weight (w) and standard deviation (σ) of all distances between all trajectories and the cluster exemplar) is defined as an **outlier** (**unusual trajectory**). Otherwise, it is defined as **normal trajectory** (see Figure 4.7). This threshold is different and specific for a given cluster and is calculated from the training data for that cluster.

4.1.4 Feature Selection

For feature selection Sequential Forward Feature Selection ($SFFS$) [162] is applied, embedded in clustering and outlier detection. Feature selection provides better feature subsets which also decreases the chance of over-fitting. It eliminates irrelevant, redundant features. Moreover, it might filter out the features which misguide the clustering. Different from the standard procedure for $SFFS$, we use the mean of $TPrate$ (represents unusual trajectory detection) and $TNrate$ (represents normal trajectory detection) as suggested in [97] rather than the accuracy (see Section 2.3.1 for descriptions and the reason why accuracy is not suitable).

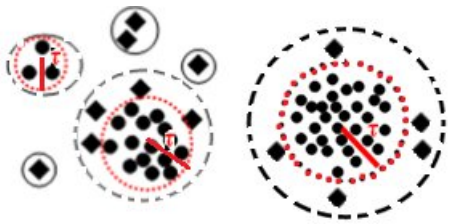


Figure 4.7: A representation of clustered data. For small clusters, boundaries are shown with thick lines and dense clusters' boundaries are shown with dashed lines. Outlier detection in dense clusters: samples which are inside of the inner circle are classified as the normal trajectories whereas the rest of the samples are classified as the unusual trajectories, given threshold τ.

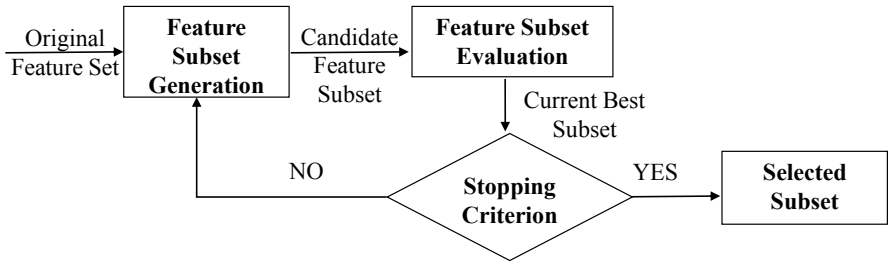


Figure 4.8: Sequential Forward Feature Selection (adapted from [2])

Feature selection is applied as follows: Given the current set of features, an additional feature is added to create a new candidate feature subset. Using this extended feature set, clustering and outlier detection are performed. The mean of $TPrate$ and $TNrate$ are found using the ground-truth labels of trajectories (feature subset evaluation). All possible additional features are added to the feature set in the same way. The feature set is extended by the feature which gives the best performance (current best subset). Adding features to the current best subset stops when the classification performance (mean of $TPrate$ and $TNrate$) on the training set decreases compared to the previous feature subset (stopping criterion) and the final subset is defined by the previous feature subset (selected subset). Those steps are illustrated in Figure 4.8.

The experiments given in this chapter explored different feature selection algorithms as well. We applied the Laplacian score [163] and Multi-cluster approach [164],

that are filtering feature selection algorithms (meaning that class labels are not being used). The results showed that $SFFS$ performs better than these filtering methods even though it is much slower. We tried different feature selection criteria as well, such as F-measure, accuracy, mutual information etc. using the $GeoMean$ as the evaluation metric. Those criteria did not perform as well as the mean of $TPrate$ and $TNrate$ (see Section 2.3.1 for definitions of evaluation metrics).

4.2 Data Set

The proposed method was tested using 683 trajectories (652 normal, 31 abnormal) from 15 hours of video (320x240 resolution, 5 frames per second) which belong to *Dascyllus reticulatus* in the Taiwanese coral reef (the most frequent species, about 150 times more common than the least common observed one while in total 15 different species were recognised [11]). This data set is different than the one presented in Section 3.2 as it belongs to only one species and one location. Considering that the fish behaviour can change during the time of the day and *Dascyllus reticulatus* is more active in the morning, we used the videos that were captured in the morning. On the other hand, the variety of normal and unusual fish trajectories are similar to the data set in Section 3.2. The trajectories of freely swimming fish were considered as normal behaviour as it is the most frequent behaviour while rare (unusual) trajectories which were not observed as much as normal trajectories such as: fish suddenly (in one frame) changing direction, interaction with coral, aggressive movements of fish (sometimes due to another fish or because of being frightened) were considered.

4.3 Results

In the testing phase, the new trajectories are classified using the outlier detection parameter w and the best feature set that are found during the training. In detail, first clustering is applied to the testing trajectories using the best features that are found in training. Outlier detection is applied using corresponding w parameter to detect unusual fish trajectories.

To evaluate the proposed flat classifier 5-fold cross validation was performed. Training and test sets were constituted randomly with the normal and unusual trajectories distributed equally in each set. The outlier detection threshold w is taken as $\{$-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6$\}$. For the best performance of the flat classifier (which

Fold	**Average of TPrate and TNrate**							
1	**0.65**	0.61						
2	0.60	0.65	0.69	0.71	**0.73**	0.70		
3	0.57	0.65	0.69	0.72	0.78	0.82	**0.84**	0.82
4	0.57	0.59	0.62	0.68	**0.75**	0.74		
5	**0.59**	0.58						

Figure 4.9: Mean of $TPrate$ and $TNrate$ after the feature is added at each iteration of $SFFS$ (Training). Best feature selection criterion value is emphasised as bold.

is obtained when w is 3), different features were selected in each fold during training. However, in 4 of 5 folds, the same feature from loop features category (Section 4.1.1.7) was selected as the first feature. The mean of $TPrate$ and $TNrate$ while features were adding one by one including the value that ended $SFFS$ [162] is given for each fold in Figure 4.9.

The proposed flat classifier was compared with the methods given in Table 4.2. The alternative feature selection methods Laplacian Score [163] and Multi-cluster approach [164] give a ranking of features from the best feature to worst feature. In these experiments, we used the same number of features as the number of features used for the flat classifier (Proposed_M2) for each corresponding fold.

Table 4.3 shows the best results of each method in terms of the *GeoMean* with the corresponding $TPrate$ and $TNrate$. For each evaluation metric the standard deviations (over the 5 cross validation folds) are also given after $\pm$ sign. The best results in terms of each evaluation metric are emphasised in bold-face.

The proposed flat classifier which integrates SFFS as the feature selection method (Proposed_M2) performed the best in terms of the $TPrate$ (unusual trajectory detection rate) and the *GeoMean* (overall trajectory detection rate). Proposed_M2_Alter3 performed as well as Proposed_M2 in terms of the *GeoMean* which is expected as the F-measure (Section 2.3.1) is a recommended metric for imbalanced data set classification. However, as we pay more attention to unusual fish trajectory detection compared to normal trajectory detection, we believe that Proposed_M2 is better than Proposed_M2_Alter3 as it has better $TPrate$. On the other hand, Proposed_M3_Alter4 was bad at unusual fish trajectory detection while performed better at normal fish trajectory detection. It can also seen that the unusual fish trajectory detection is improved

Table 4.2: Methods that are used for comparison.

Method	**Description**	**Abbreviation**
Filtering Mechanism	As described in Chapter 3.	Proposed_M1
Flat Classifier	SFFS is used for feature selection while criterion is mean of $TPrate$ and $TNrate$ and w={-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6}.	Proposed_M2
Flat Classifier Alternative 1	Laplacian Score [163] is used for feature selection, the number of features is taken as the same with Proposed_M2, and w={-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6}. To construct the k-Nearest Neighbours graph k is taken as 5 (default setting).	Proposed_M2_Alter1
Flat Classifier Alternative 2	Multi-cluster approach [164] is used for feature selection, the number of features is taken as the same with Proposed_M2 and w={-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6}. To construct the k-Nearest Neighbours graph k is taken as 5 and the number of eigenvectors is taken as 5 (default setting).	Proposed_M2_Alter2
Flat Classifier Alternative 3	SFFS is used for feature selection while criterion is F-measure and w={-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6}.	Proposed_M2_Alter3
Flat Classifier Alternative 4	SFFS is used for feature selection while criterion is accuracy and w={-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6}.	Proposed_M2_Alter4

Table 4.3: Best results of each method in terms of average $GeoMean$ with the corresponding $TPrate$ and $TNrate$. The best results are emphasised in bold-face. The standard deviations considering the cross-validation folds are also given after the $\pm$ sign.

Method	$TPrate$	$TNrate$	$GeoMean$
Proposed_M1	0.65 $\pm$0.19	0.61 $\pm$0.02	0.63 $\pm$0.10
Proposed_M2	**0.78 $\pm$0.08**	0.64 $\pm$0.10	**0.71 $\pm$0.06**
Proposed_M2_Alter1	0.77 $\pm$0.28	0.51 $\pm$0.03	0.62 $\pm$0.13
Proposed_M2_Alter2	0.61 $\pm$0.22	0.49 $\pm$0.02	0.54 $\pm$0.10
Proposed_M2_Alter3	0.70 $\pm$0.10	**0.72 $\pm$0.03**	0.70 $\pm$0.04
Proposed_M2_Alter4	0.58 $\pm$0.16	0.71 $\pm$0.02	0.63 $\pm$0.09

over the method from Chapter 3 (Proposed_M2 versus Proposed_M1). Additionally, it is seen that filtering feature selection methods (Proposed_M2_Alter1 and Proposed_M2_Alter2) are not as suitable as SFFS since they performed poor.

Some examples of misclassified unusual trajectories (FN) and misclassified normal trajectories (FP) are given in Figure 4.10. Those misclassified unusual trajectories include aggressive fish and fish suddenly diving under the coral. On the other hand, misclassified normal trajectories belong to freely swimming fish whose trajectories are complex.

4.4 Conclusions

In this chapter, we represented fish trajectories with novel descriptors which were never used before (except velocity) for fish behaviour analysis. Clustered and labelled data were used together to select the best feature set and classify trajectories as normal or unusual. As seen from the results, the flat classifier improved performance of unusual fish detection compared to filtering mechanism which was presented in Chapter 3. The proposed flat classifier is also good at detecting normal trajectories which may help marine biologists by eliminating many normal trajectories with relatively low error rate. This characteristic of the flat classifier allows the marine biologists to focus on data that is potentially unusual which is valuable especially considering the amount of data that they might have to consider. Moreover, the flat classifier's unusual trajectory detection performance can be useful especially to detect more interesting behaviours

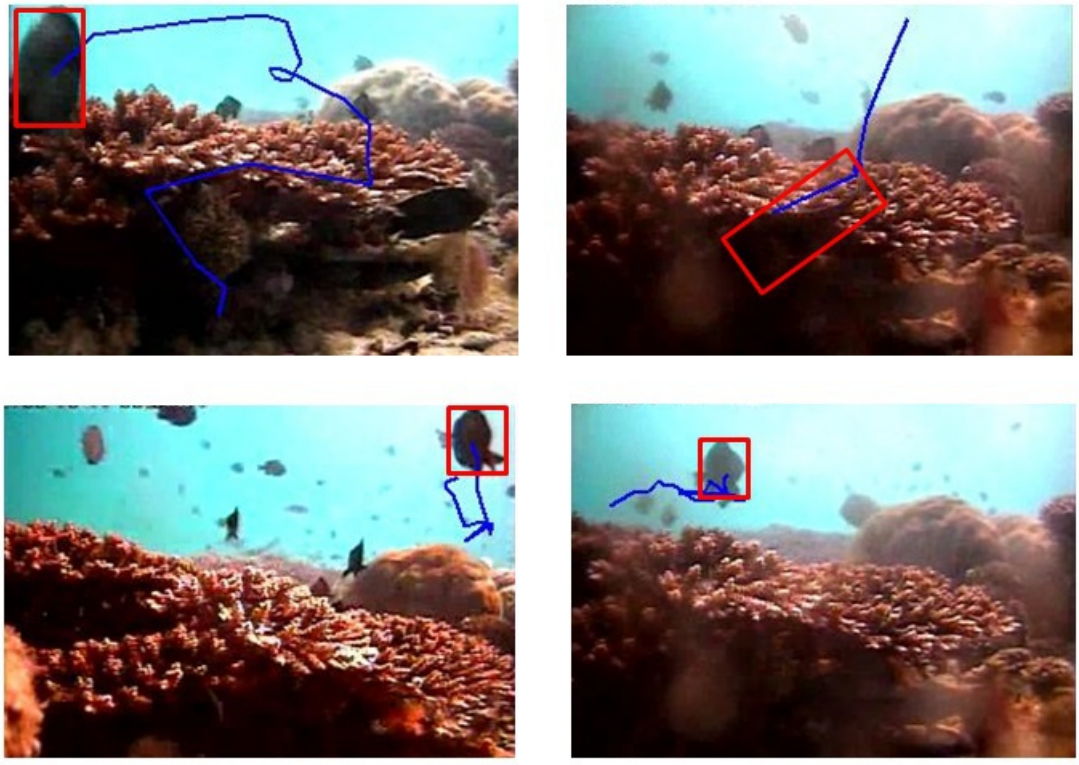


Figure 4.10: Examples of misclassified unusual trajectories (top) and misclassified normal trajectories (bottom). Trajectories are shown with blue while the last detections of the fish are shown with a red bounding box.

or even behaviour changes for a specific fish species.

The performance of the flat classifier is improved by integrating it into a hierarchical decomposition method as presented in Chapter 5. This hierarchical decomposition method should allow selecting more specific features for different trajectory clusters which can be useful considering the trajectory variety that exists even for a single fish species.

Chapter 5

Detection of Unusual Fish Trajectories Using a Clustering Based Hierarchical Decomposition

A novel hierarchical decomposition method to detect unusual fish trajectories is presented in this chapter. The basics of the proposed hierarchical decomposition method are the same as the method presented in Chapter 4. Therefore, clustering of data based on selected features without initially using the known labels is the key to partitioning the data into separable subsets. The hierarchy is automatically generated using the clustered and labelled trajectories together during training unlike research that uses a fixed hierarchy that is based on features or classes. Additionally, different from the traditional way that uses the same feature set for every level of hierarchy or a flat classifier (Chapter 4), different data and feature sets at different level of the hierarchy are used which allows more specific features to be used once the data focuses onto specific subclasses.

The main contributions of this chapter are:

- A novel approach for unusual fish trajectory detection which builds a feature or class taxonomy independent hierarchy,
- Significantly improved performance on unusual fish trajectory analysis from unconstrained underwater videos.

5.1 Methodology

The proposed hierarchy decomposition method utilises *i)* clustering, *ii)* outlier detection and *iii)* feature selection (as given in Chapter 4) to build the hierarchy. To automatically construct the hierarchy during training, clustering and outlier detection is combined with feature selection. The data is partitioned using the selected features which are determined by feature selection, outlier detection and the ground-truth labels of the training data. In other words, the clustered and labelled data are used together to determine the best feature set for a subset of training data at each level of the hierarchy. The details of the proposed method are given below.

5.1.1 Hierarchy Decomposition

At each level of the hierarchy, data is first clustered using the best feature subset which is determined by adding a single feature at each iterations of the feature selection (see Section 4.1.4). After clustering, outlier detection is applied to each cluster and outliers (unusual trajectories) for the current level of the hierarchy are found. Then, using the ground-truth data for each cluster, misclassified normal or unusual trajectories are found (if they exist). The clusters which do not contain any misclassified trajectory are kept for that level and the corresponding trajectories are not used for construction of the rest of the hierarchy. Such clusters are called **perfectly classified cluster**. On the other hand, a cluster which has at least one misclassified trajectory no matter unusual or normal (called **misclassified cluster**) is used to continue the hierarchy construction. Using the clusters that have misclassified trajectories, the hierarchy construction recurses in the same way. By repeating clustering, outlier detection and feature selection, the hierarchy construction continues until there is no cluster which is perfectly classified or all trajectories are perfectly classified.

In summary, at each level of the hierarchy, different trajectories are used and to distinguish those trajectories, different feature subsets are utilised. Once a trajectory that belongs to a perfectly classified cluster at any level of the hierarchy is detected, it is never used for hierarchy construction at the next levels.

The leaf nodes of the hierarchy contain either: **perfectly classified clusters** (mostly observed at the upper levels of the hierarchy) or **misclassified clusters** (only observed in the leaf nodes belong to the last level of the hierarchy).

A cluster called perfectly classified can be either:

- **Perfectly classified mixed cluster:** Contains unusual and normal trajectories. All trajectories are correctly classified using the outlier detection threshold.
- **Perfectly classified pure normal cluster:** A dense cluster which contains only normal trajectories which are correctly classified using the outlier detection threshold.
- **Perfectly classified pure unusual cluster:** Contains only unusual trajectories which are correctly classified, due to being in small clusters. We assume that small clusters contain only unusual trajectories.

A cluster called misclassified can be either:

- **Misclassified mixed cluster:** A dense or small cluster which contains both unusual and normal trajectories with at least one trajectory wrongly classified using the outlier detection threshold.
- **Misclassified pure normal and dense cluster:** Contains only normal trajectories with at least one trajectory wrongly classified as an unusual trajectory using the outlier detection threshold.
- **Misclassified pure normal and small cluster:** Contains only normal trajectories with at least one trajectory wrongly classified as an unusual trajectory due to being in a small cluster.
- **Misclassified pure unusual cluster:** A dense cluster that contains unusual trajectories with at least one trajectory wrongly classified as a normal trajectory using the outlier detection threshold.

Illustration of hierarchy construction is given in Figure 5.1 and the pseudo-code for that is given in Figure 5.2.

5.1.2 New Trajectory Classification Using the Constructed Hierarchy

A new trajectory is classified using the constructed hierarchy with all perfectly classified clusters and misclassified clusters at all levels, the selected feature subsets for each level and the outlier detection thresholds for each cluster. It is rule based and

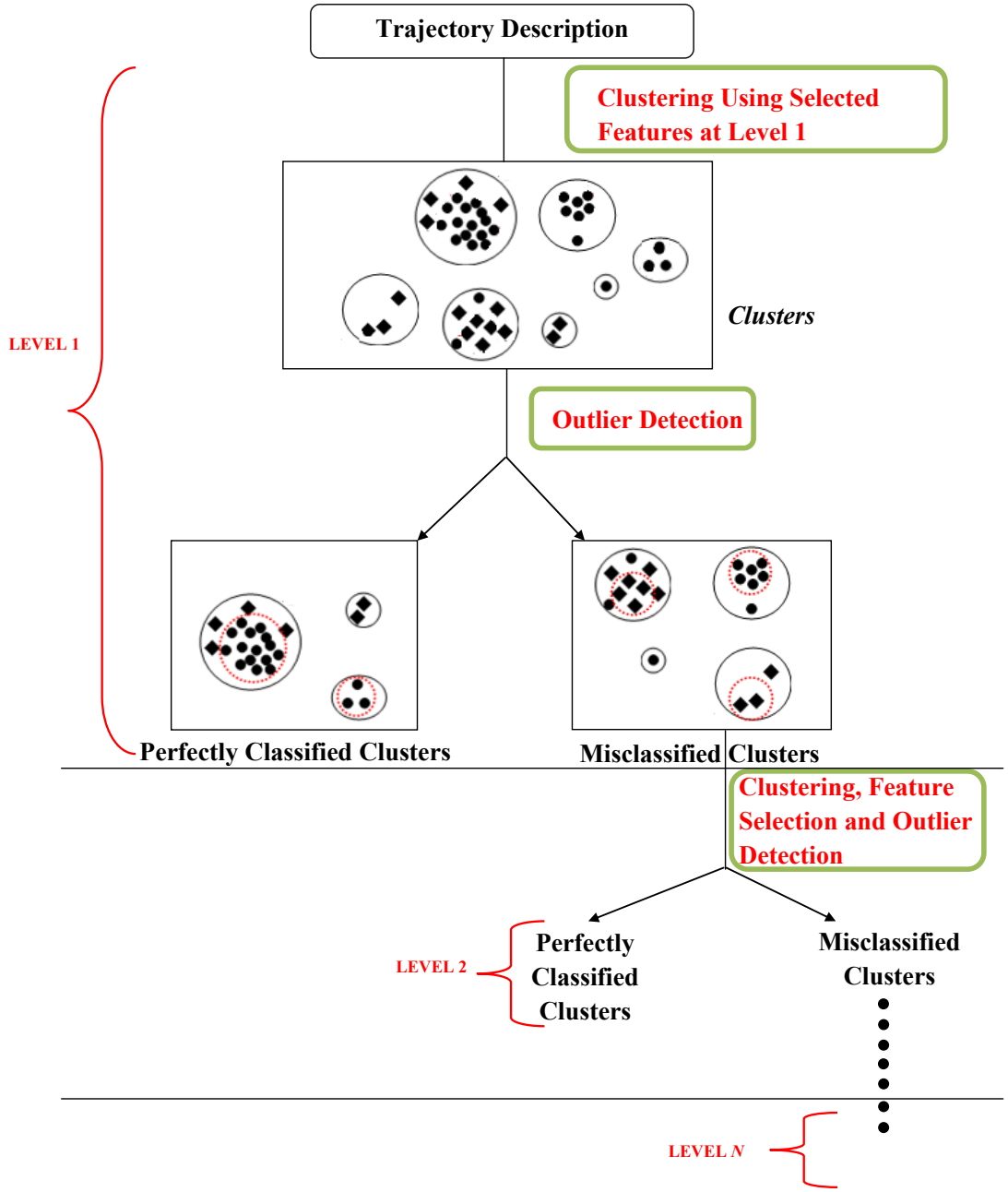


Figure 5.1: Hierarchy Construction

```
Input: Training Set: X ={X_1, X_2,...X_N}
       Ground-truth labels: G ={G_1, G_2,...G_N}
       Size of training set: N
       Features: F ={f_1, f_2,...,f_M} % all possible features
       Total number of features: M
       Feature Selection Criterion Function: E
       Outlier detection thresholds: w
Output: Hierarchy H ={ Total number of levels: L
                       Selected Feature Subsets: selFea ={sf_1, sf_2,... sf_L }, where sf_i ∈ F
                       Perfectly Classified Clusters: C_perfect ={C_P1, C_P2,... C_PL}
                       Misclassified Clusters: C_mis ={C_M1, C_M2,...C_ML }
                                          where C_Pi and C_Mi ⊂ set of all subsets of X
begin:
   for z=1:size(w)
      w_z=w(z); % current outlier detection threshold
      current_level=1
      while current_level >=1
            if current_level ==1
                  remaining_samples=X;
            else
                  remaining_samples=samples(C_Mcurrent_level-1);
            end
            featureSelection_converged=false;

            sf_current_level ={}; F̂=F;    % all features

            while (NOT featureSelection_converged)
                  for f_i ∈ F̂
                      [C] =Clustering (remaining_samples, (sf_current_level ∪ {f_i }));

                       [C_Pi, C_Mi] =OutlierDetection (C, w_z);

                      e_i =evaluate( C_Pi , C_Mi , G);

                  end
                   select j=argmax_i e_i
                  sf_current_level= sf_current_level ∪ {f_j}
                  F̂ =F̂ \ {f_j}
                  featureSelection_converged =E( sf_current_level ) ≤ E( sf_current_level \ {f_j})
            end
            H. L =current_level;
            H. C_perfect(H.L) =C_Pj-1;
            H. C_mis(H.L) =C_Mj-1;
            H. selFea(H.L) =sf_current_level \{f_j};

            if notEmpty(H.C_perfect (H.L)) and size(samples(H.C_perfect )) ≠ N
                                          % there is at least one perfectly classified cluster
                                          % and the total number of perfectly classified
                                          % samples are not equal to N
                  current_level =current_level+1;
            else
                   current_level =0;
            end
       end
     end
end
```

Figure 5.2: The pseudo-code for hierarchy construction

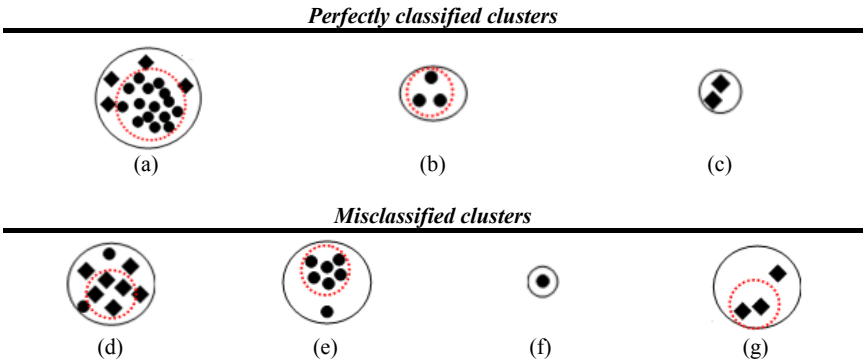


Figure 5.3: Cluster types: perfectly classified clusters, misclassified clusters. A perfectly classified cluster can be *a)* perfectly classified mixed, *b)* perfectly classified pure normal, *c)* perfectly classified pure unusual. A misclassified cluster can be *d)* misclassified mixed, *e)* misclassified pure normal when the cluster is a dense cluster, *f)* misclassified pure normal when the cluster is a small cluster, *g)* misclassified pure unusual. Diamonds represent the unusual trajectories while circles represent the normal trajectories. The outlier detection thresholds for dense clusters are shown with dashed red circles.

based on finding the closest cluster at each level of the hierarchy. The closest cluster is found using the Euclidean distance between the new trajectory and the cluster exemplars with the selected features for that specific level, including misclassified clusters as well. Therefore, at each level in the hierarchy, the closest cluster can be:

- Perfectly classified mixed cluster (Figure 5.3a),
- Perfectly classified pure normal cluster (Figure 5.3b),
- Perfectly classified pure unusual cluster (Figure 5.3c),
- Misclassified mixed cluster (Figure 5.3d),
- Misclassified pure normal and dense cluster (Figure 5.3e),
- Misclassified pure normal and small cluster (Figure 5.3f) or
- Misclassified pure unusual cluster (Figure 5.3g) as described above.

Based on the closest cluster and the position in the closest cluster, the class decision for the new trajectory can be:

- Unusual trajectory,
- Candidate normal trajectory or
- No effect on the decision.

At each hierarchy level (with its clusters, outlier detection thresholds, and selected features) and a new trajectory:

- The closest cluster is a perfectly classified pure unusual cluster which makes the new trajectory an **unusual trajectory** and **classification stops** (there is no need to look at any other level of the hierarchy).
- The closest cluster is a perfectly classified mixed cluster and the new trajectory is further than the outlier detection threshold of that cluster which makes the new trajectory an **unusual trajectory** and **classification stops** (there is no need to look at any other level of the hierarchy).
- The closest cluster is a perfectly classified pure normal cluster and the distance between the new trajectory and the corresponding cluster's centre is smaller than the outlier detection threshold of that cluster. This makes the new trajectory a **candidate normal trajectory**. The new trajectory **goes to the next level** of the hierarchy.
- The closest cluster is a perfectly classified pure normal cluster and the new trajectory is further than the outlier detection threshold of that cluster. This makes the new trajectory an **unusual trajectory** and **classification stops** (there is no need to look at any other level of the hierarchy).
- The closest cluster is a perfectly classified mixed cluster and the distance between the new trajectory and cluster centre is smaller than the threshold, then the new trajectory is a **candidate normal trajectory**. The new trajectory **goes to the next hierarchy level**.
- The closest cluster is a misclassified cluster (pure or mixed) then the new trajectory **proceeds to the next level**. This does **not have any effect on the classification** of the new trajectory unless all the closest clusters at each level are misclassified clusters.

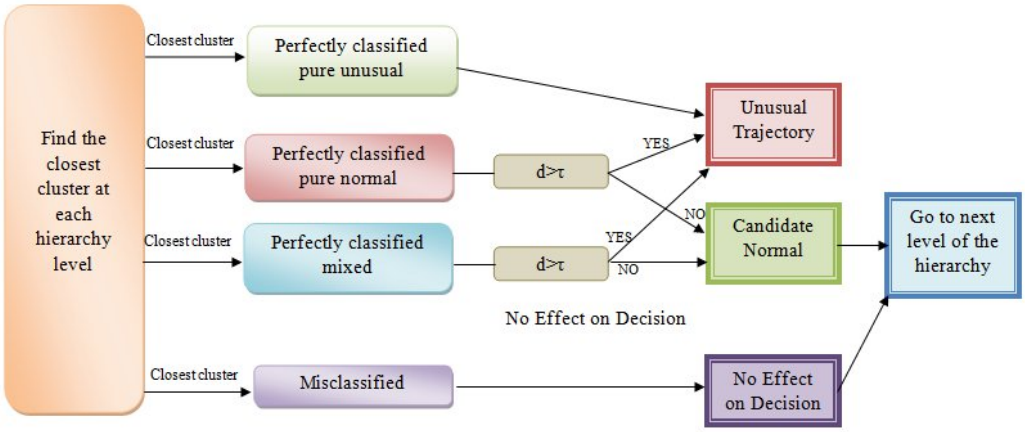


Figure 5.4: New trajectory classification using the hierarchy.

Those rules are illustrated in Figure 5.4.

In summary, even a single level's decision as unusual trajectory is enough to classify the new trajectory as an unusual trajectory regardless of the level of the hierarchy. On the other hand, if there is no decision as an unusual trajectory at any level and if the decision of at least one level is candidate normal then the class of the new trajectory is declared to be **normal**. However, it is possible that the closest cluster at each level of the hierarchy is a misclassified cluster. In this case, we use the ground-truth labels of the training trajectories and apply the following rules, starting from the top of the hierarchy:

- The closest cluster at the current level contains all normal trajectories by looking at the ground-truth class labels: If the new trajectory is further than the rest of the samples in that cluster this makes it an **unusual trajectory** and **classification stops here**. Otherwise the data **goes to the next hierarchy level**.
- The closest cluster contains all unusual training trajectories by the ground-truth: The new trajectory is classified as an **unusual trajectory** and **classification stops here**.
- The closest cluster contains both normal and unusual training trajectories: In this case, we apply the nearest neighbour rule which makes the class of the new trajectory the same as the closest training sample's class. If the class is an **unusual** class then **classification stops**. Otherwise, the data **goes to the next level** to apply above rules.

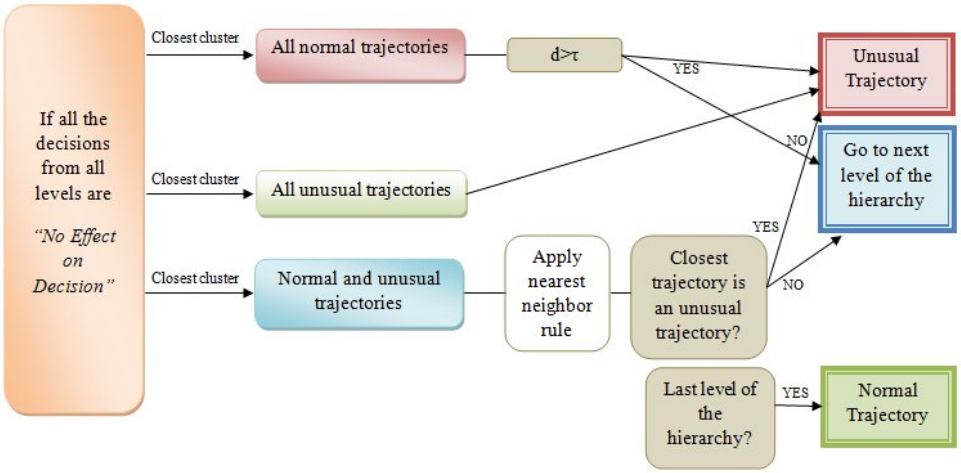


Figure 5.5: New trajectory classification when the decisions of all levels are **no effect on decision**.

- If the new trajectory reaches the last level and could not be classified using the rules given above, then it is classified as a **normal trajectory**.

These rules are illustrated in Figure 5.5.

Other heuristics than we use (decision as an **unusual trajectory** at any level **stops classification** of the new trajectory while decision as a **normal trajectory send the new trajectory to the next level**), can also be applied. For instance, the inverse heuristic: any decision as normal trajectory stops classification regardless of the level of the hierarchy while a decision as an unusual trajectory send the new sample to the next hierarchy level can be applied. Alternatively, majority voting on the decisions at each level can determine the final class of the new trajectory. The experiments comparing different heuristics are given in Section 5.2.2.2.

The flow chart of the proposed heuristic for the classification of a new trajectory using the previously constructed hierarchy is given in Figure 5.6.

5.2 Experimental Work

The proposed method was compared with the state of art classification algorithms, outlier detection methods and trajectory analysis methods. The evaluations were performed using the fish trajectory data set and the pedestrian data set in terms of $GeoMean$ (Eq. 2.7) with the corresponding $TPrate$ (represents unusual trajectory de-

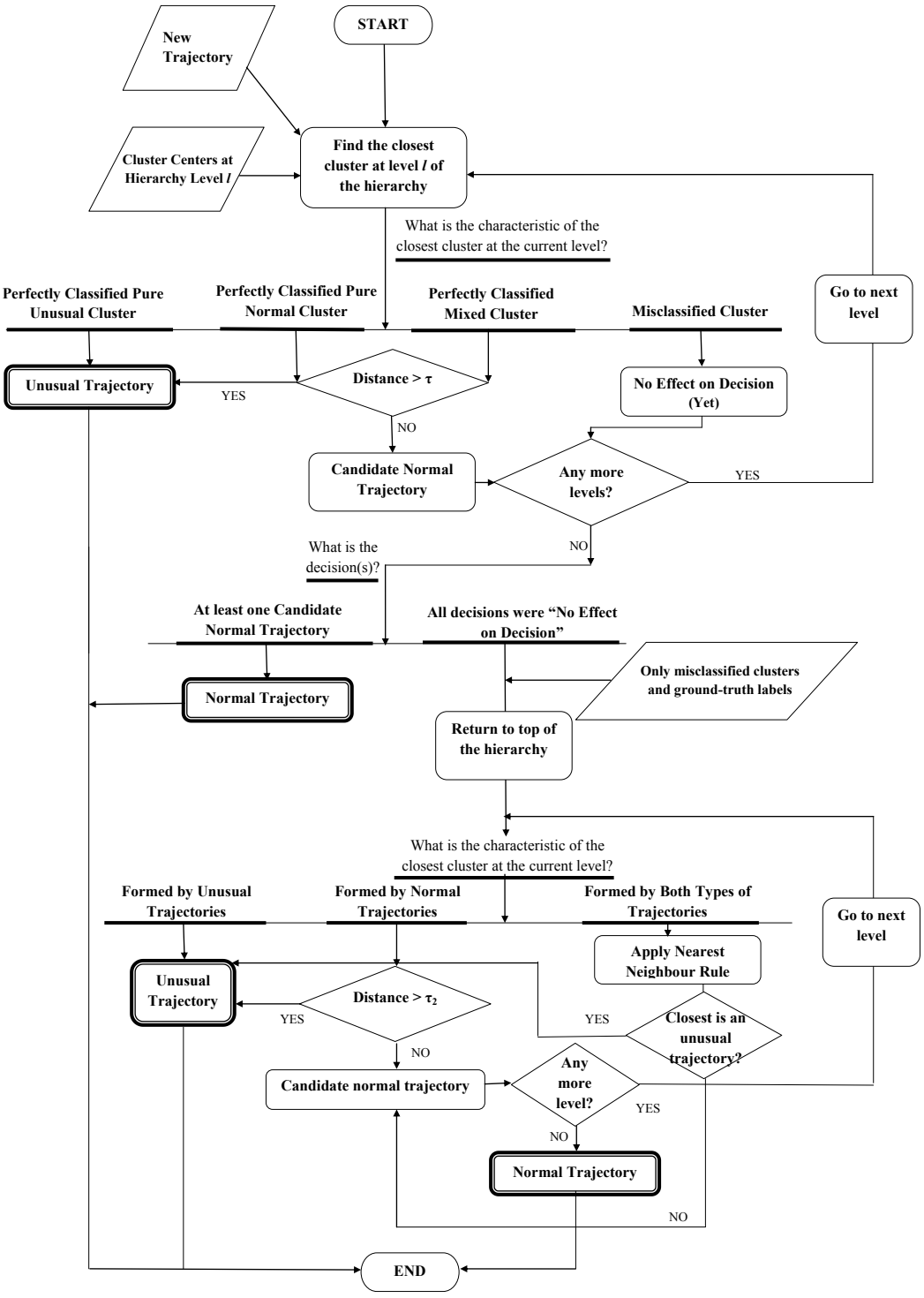


Figure 5.6: The flow chart of classification of a new trajectory using a previously (during training) constructed hierarchy. Decisions are all shown with rounded rectangles either with single or double line. Rounded rectangles with double lines represent the final class of the new trajectory whereas single line rounded rectangles indicate provisional decisions.

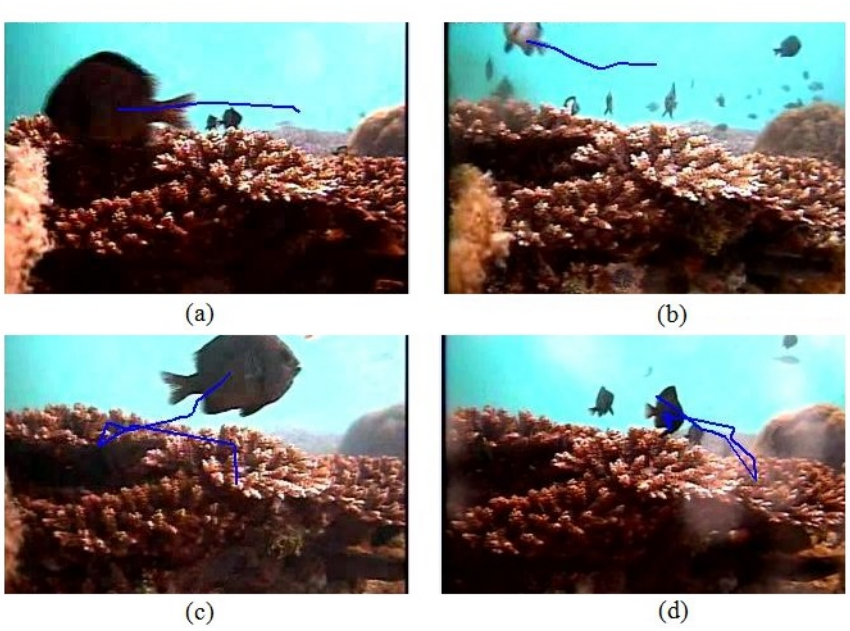


Figure 5.7: (a-b) Normal fish trajectory examples, (c-d) Unusual fish trajectory examples.

tection, Eq. 2.2) and $TNrate$ (represents normal trajectory detection, Eq. 2.3).

5.2.1 Data Sets

The proposed methods and all the methods were applied to 3102 fish trajectories (3043 normal, 59 unusual trajectories). To the best of our knowledge, this data set is **the largest fish trajectory data set** and **the largest labelled trajectory data set** in general (`http://groups.inf.ed.ac.uk/f4k/GROUNDTRUTH/BEHAVIOR/`). Data includes trajectories of *Dascyllus reticulatus*. Data was collected from 93 different videos having 320x240 resolution, 5 frames per second and captured in the morning. The normal and unusual behaviours are determined by visual inspection and also examined by marine biologists.

The most usual and frequent behaviours in the data set are: fish hovering over the coral (Figure 5.7a) and freely swimming fish in the open sea (Figure 5.7b). On the other hand, unusual trajectories are: fish suddenly (in one frame) changing direction (predator avoidance, Figure 5.7c), fish biting at coral (also interaction with plankton, Figure 5.7d), fish diving quickly between the coral branches when frightened or to hide from predators, and aggressive fish which are moving fast. A trajectory that has normal and unusual segments is assumed as unusual.

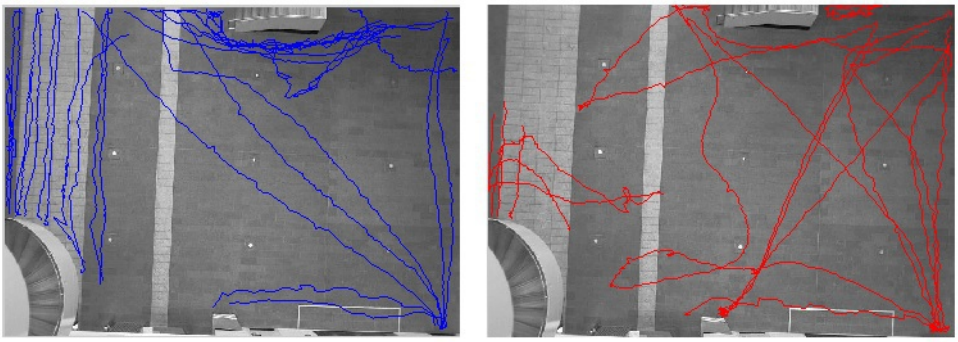


Figure 5.8: Data set which belongs to 1st of September 2009 in the Forum Pedestrian database [3]. Examples of normal (blue) and abnormal trajectories (red).

The proposed method was also applied to a pedestrian trajectory data set in the Forum Pedestrian database (`http://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/`, [3]). In this database, the field of the view of the camera was divided into regions as: main entrance to the building, lifts, access to the Atrium, access to the hall, staircase, reception desk and four exits. A trajectory is classified as normal *i)* if it represents a clear goal such as going from one exit to another and *ii)* the goal was achieved in an efficient way which means with a trajectory close to a straight line. Otherwise, it was labelled as abnormal. The first rule is more serious than the second rule as there are some cases that a normal trajectory does not meet the second rule but obeys the first rule [3].

The data set from the 1st of September 2009 (which is one of the largest sets) was chosen to analyse. This set includes 1624 normal trajectories and 718 abnormal trajectories that were captured from different people. Examples of normal and abnormal trajectories from this data set are shown in Figure 5.8. The normal and abnormal trajectories were labelled by the implementation in [3]. The labelling of the data set was also manually inspected.

Features, similar to those for fish trajectory data set were extracted (Section 4.1.1): velocity and acceleration based, vicinity based, curvature scale space based, centre distance function in 2 dimensions based, loop based, moment based, and turn based. Additionally, trajectory points after cubic B-spline fitting and the statistical features (mean, standard deviation, minimum, maximum, median, the number of local minima, the number of local maxima, skewness and kurtosis) which are extracted from the deviation between the reconstructed trajectory and the original trajectory were also used. Each trajectory was approximated with a cubic spline with 6 control points

using the implementation in [31]. Altogether 758 features were obtained. To prevent possible over-training or curse of dimensionality, *PCA* was applied to each group of features individually (except the trajectory points that are obtained after cubic B-spline fitting) as described in Section 4.1.1. As a result, 57 features were obtained.

5.2.2 Results

The results presented in this section can be divided into 4 subsections:

- Comparison with the state of art methods using the fish trajectory data set,
- Results when alternative hierarchy decomposition methods are applied to the fish trajectory data set,
- Results when outlier removal and normalised fish trajectory features are used,
- Comparison with the state of art methods using the Forum pedestrian database [3].

For all experiments presented in this section, 9-fold cross validation was performed. Training, validation and test sets were constituted randomly and the normal and unusual trajectories were distributed equally in each set. For the methods combined with $SFFS$ (Section 4.1.4), validation sets were used to pick the best feature set for each method individually. For others including the proposed methods, validation sets were not used. The training and testing sets were kept the same for all methods.

5.2.2.1 Comparison with the state of art methods using the fish trajectory data set

The proposed hierarchical decomposition method was compared with the proposed methods in Chapters 3 and 4, the state of art classification methods, other popular outlier detection and trajectory analysis methods using fish trajectory data set. The definitions of these methods with the applied parameters are given in Table 5.1.

The best results of the methods using fish trajectory data set are given in terms of the $GeoMean$ with the corresponding $TPrate$ and $TNrate$ in Table 5.2. Additionally, to compare the methods more precisely, the $approximated\ TPrate$ results for the fixed $TNrate = 0.88$ which was found by $Proposed\_M3$ are also given. The $approximated\ TPrate$ are obtained using the formula given in Eq. 5.1 (see Section 2.3.1 for definitions), AUC value of each method and the fixed $TNrate$. Rearranging equation and

Table 5.1: Methods that are used for comparison.

Method	Description	Abbreviation
k-Nearest Neighbours	k={1, 2, 3, 4, 5, 10, 15, 25} were used as the common parameters.	kNN
kNN with Feature Selection	The same k values with the kNN were used while sequential SFFS was applied as given in Section 4.1.4.	kNN_wFea
Support Vector Machine	As the kernel function, a radial basis function with varying kernel parameters was used. Hyper-planes were separated by Sequential Minimal Optimisation. All features were used.	SVM
SVM with Feature Selection	Applied as given in the SVM description but integrated with SFFS as given in Section 4.1.4.	SVM_wFea
Random Forest with Balanced Training [165]	A number of trees {10, 30, 50, 70, 100, 120, 150, 200, 500, 1000} were tested and the trees were grown without pruning. For node splitting, the Gini index [166] was used. All unusual trajectories were kept, and subsets of the normal trajectories were chosen randomly to build the decision trees. The number of normal trajectories in the chosen subset was equal to the number of total unusual trajectories. All features were used.	RF_BT
RF_BT with Feature Selection	Applied as given in the RF_BT description but integrated with SFFS as given in Section 4.1.4.	RF_BT_wFea
Unsupervised Modelling of Object Tracks [51]	Normalised Cuts spectral clustering was applied to unusual and normal trajectories individually and each cluster of behaviour was modelled as a mixture of Gaussian in the spectral embedding space. A new trajectory was classified by projecting it into the spectral embedding space for normal and unusual classes and based on the likelihood that the new track was classified as a normal or unusual trajectory. Different sigma values such as {1, 10, 20 etc.} and different number of cluster sizes {10, 15, 20, 30, 40, 50, 60, 80, 90} for normal and usual clusters were tested.	UMOT
Local Outlier Factor [167]	It is a density based method which considers a sample to be an outlier if its surrounding space contains few samples. It does not use any clustering technique. Training is performed only using normal classes. During validation normal and unusual class trajectories are used and the best feature set is selected using sequential forward feature selection. The neighbourhood is defined with a parameter called k. k was taken as {1, 3, 5, 10, 15, 20 and 25}.	LOF
Filtering Mechanism	As described in Chapter 3. Pixels size {2, 4, 8, 16, 20} were taken to define the search area.	Proposed_M1
Flat Classifier	As described in Chapter 4. The outlier detection parameter w was taken as {-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, 6}.	Proposed_M2
Hierarchical Decomposition	The outlier detection parameter w was taken as {0, 0.3, and 1}.	Proposed_M3

Table 5.2: Best results of each method in terms of average $GeoMean$ with the corresponding $TPrate$ and $TNrate$ when fish trajectory data set is used. $Approximated$ $(Approx.)$ $TPrate$ is calculated for fixed $TNrate = 0.88$. The standard deviations (considering cross validation folds) of the methods are also given after $\pm$ sign. The best results are emphasised in bold-face.

Method	$TPrate$	$TNrate$	$GeoMean$	$Approx.TPrate$
kNN	0.26 $\pm$0.08	**0.99 $\pm$0.01**	0.50 $\pm$0.09	0.37
kNN_wFea	0.37 $\pm$0.28	**0.99 $\pm$0.01**	0.60 $\pm$0.27	0.48
SVM	0.21 $\pm$0.07	**0.99 $\pm$0.01**	0.45 $\pm$0.07	0.32
SVM_wFea	0.81 $\pm$0.16	0.93 $\pm$0.03	0.86 $\pm$0.09	0.86
RF_BT	0.87 $\pm$0.01	0.93 $\pm$0.06	0.90 $\pm$0.03	0.92
RF_BT_wFea	0.88 $\pm$0.01	0.91 $\pm$0.10	0.89 $\pm$0.05	0.91
UMOT	0.57 $\pm$0.20	0.85 $\pm$0.11	0.70 $\pm$0.04	0.54
LOF	0.62 $\pm$0.17	0.97 $\pm$0.01	0.77 $\pm$0.08	0.71
Proposed_M1	0.80 $\pm$0.20	0.77 $\pm$0.04	0.78 $\pm$0.09	0.66
Proposed_M2	0.81 $\pm$0.17	0.76 $\pm$0.02	0.78 $\pm$0.09	0.70
Proposed_M3	**0.94 $\pm$0.10**	0.88 $\pm$0.02	**0.91 $\pm$0.05**	**0.94**

substitute terms allow us to approximate the $TPrate$ given the AUC and $TNrate$. For each evaluation metric the standard deviations (considering cross validation folds) are also given after the $\pm$ sign. The best results of each evaluation metric are emphasised in bold-face.

$$\begin{aligned} AUC &= \frac{1 + TPrate - FPrate}{2} \\ TPrate &= 2AUC - TNrate \end{aligned} \tag{5.1}$$

The results showed that the proposed hierarchical decomposition method ($Proposed\_M3$) had the highest unusual fish trajectory detection rate ($TPrate$) and the highest $Approx.$ $TPrate$ while it was also the best method overall ($GeoMean$). For the proposed method the best performance was observed when the outlier detection threshold w is 1. The depth of the hierarchy was at most 11 while at least 3 for the 9-folds. Paired t-tests were applied to the $GeoMean$ data between each method and the proposed method. The proposed method performed significantly better than all methods except RF_BT, RF_BT_wFea and SVM (α=0.05).

5.2.2.2 Results when alternative hierarchy decomposition methods are applied to the fish trajectory data set

The proposed method was evaluated by applying different heuristics that are used to classify the new trajectories $(Alter1-4)$. The effect of having different levels with different subsets of trajectories and features was shown by applying all selected features from the different levels as they were selected in a single level and for all training trajectories $(SingleLevProposed)$. The features selected by the proposed method were evaluated using the SVM classifier $(SVMwPropFea)$. Additionally, the benefit of the outlier detection algorithm was tested by keeping the same heuristic but replacing the decision maker by SVM $(Hie\_SVM, Hie\_SVM\_Alter1)$. These methods are defined in more detail in Table 5.3. The best results in terms the $GeoMean$ with corresponding $TPrate$, $TNrate$ and the $Approx.\ TPrate$ when $TNrate = 0.88$ are given in Table 5.4. For each evaluation metric the standard deviations (considering cross validation folds) are also given after the $\pm$ sign. The best results of each evaluation metric are emphasised in bold-face.

As seen in Table 5.4, Proposed_M3 was the best in terms of the $GeoMean$, $TPrate$ and $Approx.\ TPrate$. SVM_wPropM3Fea also performed well which means that the selected features by Proposed_M3 are representative to detect unusual fish trajectories. SingleLev_Proposed_M3 did not perform as well as Proposed_M3 which means that utilising different features for different trajectory subsets is more successful. Proposed_M3_Alter1 and Proposed_M3_Alter4 did not perform as well as Proposed_M3 and Proposed_M3_Alter2 and Proposed_M3_Alter3. That is because $TNrate$ of them were not as good as $TPrate$ of them which decreased the $GeoMean$ as well. Hie_SVM did not perform significantly worse than Proposed_M3 but on average Proposed_M3 was better with higher $TPrate$. Similar to Proposed_M3_Alter1, Hie_SVM_Alter1 also tended to classification of normal class therefore its $TNrate$ was greater than the $TNrate$ of Hie_SVM but its $TPrate$ was much worse, which made its $GeoMean$ worse than the $GeoMean$ of Hie_SVM.

5.2.2.3 Results when outlier removal and normalised fish trajectory features are used

To further explore the features used to represent fish trajectories (given in Section 4.1.1), some pre-processing steps were applied. A 5% outlier removal was applied to each feature and then each feature was normalised by subtracting its mean and dividing

Table 5.3: Definition of alternative hierarchical decomposition methods.

Method	**Description**	**Abbreviation**
Hierarchical Decomposition	Outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$.The heuristic is: a decision as a **unusual trajectory** at any level **stops the classification** of the new trajectory and the new trajectory become **unusual**, while a decision as a **normal trajectory sends the new trajectory to the next hierarchy level**.	Proposed_M3
Single level classification using features selected by Proposed_M3	A single level of clustering and outlier detection were applied, but using all of the features selected by all levels of Proposed_M3, and without further feature selection. Hence, the new hierarchy has only one level with the selected features of original Proposed_M3. Outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$.	SingleLev_Proposed_M3
Proposed_M3 Alternative Heuristic 1	Outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$. The heuristic is: a decision as a **normal trajectory** at any level **stops the classification** of the new trajectory and it become **normal**, while a decision as an **unusual trajectory sends the new trajectory to the next hierarchy level**.	Proposed_M3_Alter1
Proposed_M3 Alternative Heuristic 2	Find the closest cluster at each level using corresponding features. Then, find **the closest cluster of all** which might be from any level of the hierarchy. If the closest cluster is a perfectly classified cluster then, a decision as unusual trajectory makes the new trajectory **unusual** and a decision as normal trajectory makes the new trajectory **normal**. If the closest cluster is a misclassified cluster then, ground-truth labels are used as Proposed_M3 applies. As outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$.	Proposed_M3_Alter2
Proposed_M3 Alternative Heuristic 3	Apply Proposed_M3, but instead of classifying the new trajectory as unusual with an unusual trajectory decision of any level, classify the new trajectory using **majority voting**. If the number of levels classifying the trajectory as unusual and normal are equal, then the new trajectory is **unusual**. Outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$.	Proposed_M3_Alter3
Proposed_M3 Alternative Heuristic 4	Apply Proposed_M3, but instead of classifying the new trajectory as unusual with an unusual trajectory decision of any level, classify the new trajectory using **majority voting**. If the number of levels classifying the trajectory as unusual and normal are equal then the new trajectory is **normal**. Outlier detection parameter w was taken as $\{0, 0.3 \text{ and } 1\}$.	Proposed_M3_Alter4
SVM using features selected by Proposed_M3	The features selected by Proposed_M3 in all levels are utilised. SVM was applied with the settings given in Table 5.1.	SVM_wPropM3Fea
Hierarchical SVM	Applying Proposed_M3 but using SVM as the classifier instead of the proposed outlier detection algorithm. SVM was applied with the settings given in Table 5.1.	Hie_SVM
Hierarchical SVM Alternative Heuristic 1	Applying heuristic Alter1 but using SVM as the classifier instead of outlier detection algorithm. SVM was applied with the settings given in Table 5.1.	Hie_SVM_Alter1

Table 5.4: Best results for the alternative hierarchy decomposition methods in terms of average *GeoMean* with the corresponding $TPrate$ and $TNrate$ when the fish trajectory data set is used. $Approximated\ (Approx.)\ TPrate$ is calculated for fixed $TNrate = 0.88$. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face.

Method	$TPrate$	$TNrate$	$GeoMean$	$Approx.TPrate$
Proposed_M3	**0.94 ±0.10**	0.88 ±0.02	**0.91 ±0.05**	**0.94**
SingleLev_Proposed_M3	0.58 ±0.16	0.90 ±0.03	0.72 ±0.10	0.60
Proposed_M3_Alter1	0.37 ±0.16	0.97 ±0.01	0.59 ±0.13	0.40
Proposed_M3_Alter2	0.92 ±0.02	0.80 ±0.17	0.85 ±0.09	0.84
Proposed_M3_Alter3	0.88 ±0.10	0.91 ±0.02	0.89 ±0.05	0.90
Proposed_M3_Alter4	0.48 ±0.21	0.96 ±0.02	0.68 ±0.17	0.53
SVM_wPropM3Fea	0.89 ±0.11	0.86 ±0.05	0.87 ±0.06	0.86
Hie_SVM	0.92 ±0.10	0.82 ±0.09	0.86 ±0.02	0.86
Hie_SVM_Alter1	0.36 ±0.34	**0.98 ±0.03**	0.59 ±0.34	0.41

by the corresponding standard deviation (z-score normalisation). The methods given in Table 5.5 were applied to pre-processed features and compared with the results using the un-processed features used in previous experiments (Table 5.6). For each evaluation metric the standard deviations (considering cross validation folds) are also given after $\pm$ sign. The best results of each evaluation metric are emphasised in bold-face.

According to results given in Table 5.6, when the pre-processed fish trajectory features were used, for all methods $TPrate$ (unusual trajectory detection) decreased and $TNrate$ (normal trajectory detection) increased except SVM. For the *GeoMean*, performance of Proposed_M3_wPreProFea was slightly better but since its $TPrate$ was worse than Proposed_M3, features without pre-processing can be preferred. For Proposed_M2 the *GeoMean* did not change when pre-processed features were used but since the $TPrate$ decreased for this method, using the pre-processed features is still worse. We hypothesise that removing the outliers during training has somehow damaged the ability to detect the unusual trajectories, as observed with the decreases in the $TPrate$.

Table 5.5: Applied methods using pre-processed (outlier removed and normalised) fish trajectory features.

Method	**Description**	**Abbreviation**
Flat Classifier	As defined in Table 5.1 using outlier removal and normalised fish trajectory features.	Proposed_M2_wPreProFea
SVM	As defined in Table 5.1 using outlier removal and normalised fish trajectory features.	SVM_wPreProFea
Hierarchical Decomposition	As defined in Table 5.1 using outlier removal and normalised fish trajectory features.	Proposed_M3_wPreProFea

Table 5.6: Best results of the methods in terms of average $GeoMean$ with the corresponding $TPrate$ and $TNrate$ for the fish trajectory data set with and without pre-processed features. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face.

Method	$TPrate$	$TNrate$	$GeoMean$
Proposed_M2	0.81 $\pm$0.17	0.76 $\pm$0.02	0.78 $\pm$0.09
Proposed_M2_wPreProFea	0.75 $\pm$0.10	0.81 $\pm$0.02	0.78 $\pm$0.05
SVM	0.81 $\pm$0.16	0.93 $\pm$0.03	0.86 $\pm$0.09
SVM_wPreProFea	0.77 $\pm$0.20	0.90 $\pm$0.06	0.83 $\pm$0.10
Proposed_M3	**0.94 $\pm$0.10**	0.88 $\pm$0.02	0.91 $\pm$0.05
Proposed_M3_wPreProFea	0.88 $\pm$0.13	**0.96 $\pm$0.05**	**0.92 $\pm$0.08**

Table 5.7: Methods that are used for comparison when using the pedestrian data set [3].

Method	Description	Abbreviation
Hierarchical Decomposition	The outlier detection parameter w was taken as $\{$-1, 0, 0.3, 0.6, 1 and 2$\}$.	Proposed_M3
Hierarchical Decomposition Alternative 1	As given in Table 5.1 but with outlier detection parameter w=$\{$-1, 0, 0.3, 0.6, 1 and 2$\}$.	Proposed_M3_Alter1
Support Vector Machine with Feature Selection	As given in Table 5.1.	SVM_wFea
Random Forest with Balanced Training [165]	As given in Table 5.1.	RF_BT
RF_BT with Feature Selection	As given in Table 5.1.	RF_BT_wFea
Local Outlier Factor [167]	As given in Table 5.1.	LOF

5.2.2.4 Comparison with the state of art methods using the Forum Pedestrian Database [3]

To show that the proposed hierarchical decomposition method is not limited to fish trajectory analysis but a general unusual trajectory detection method as well, we applied it to the pedestrian data set [3] as given in Section 5.2.1. The performance of the proposed method was compared with RF_BT, RF_BT_wFea, SVM_wFea as they performed about as well as the Proposed_M3 when the fish trajectory data set is used (Section 5.2.2.1). Additionally, LOF [167] was also considered since this method was one of the most popular outlier detection methods and was applied in [168] as one of the state of art methods for that pedestrian data set. The parameter settings used for each method are given in Table 5.7.

The best results of the methods are given in Table 5.8 in terms of the *GeoMean* with the corresponding $TPrate$ and $TNrate$. $Approximated\ TPrate$ is calculated when $TNrate = 0.87$ which was obtained by $Proposed\_M3$. For each evaluation metric the standard deviations (considering cross validation folds) are also given after the $\pm$ sign. The best results of each evaluation metric are emphasised in bold-face. For this data set the best performance of the $Proposed\_M3$ was observed when the outlier detection

Table 5.8: Best results of each method in terms of average $GeoMean$ with the corresponding $TPrate$ and $TNrate$ when the pedestrian trajectory data set is used. $Approximated\ (Approx.)\ TPrate$ is calculated for fixed $TNrate = 0.87$. The standard deviations (considering cross validation folds) of the methods are also given after the $\pm$ sign. The best results are emphasised in bold-face.

Method	$TPrate$	$TNrate$	$GeoMean$	$Approx.TPrate$
Proposed_M3	**0.87 ±0.06**	0.86 ±0.05	**0.86 ±0.02**	**0.87**
Proposed_M3_Alter1	0.63 ±0.13	**0.96 ±0.02**	0.77±0.08	0.68
SVM_wFea	0.83 ±0.03	0.79 ±0.04	0.81 ±0.01	0.74
RF_BT	0.80 ±0.02	0.86 ±0.03	0.83 ±0.02	0.78
RF_BT_wFea	0.79 ±0.04	0.81 ±0.05	0.80 ±0.04	0.72
LOF	0.53 ±0.07	0.95 ±0.02	0.71 ±0.04	0.57

threshold w is 0.3. The depth of the hierarchy was at most 5 while mostly 3 for the 9-folds.

For this data set, the proposed method performed the best to detect unusual trajectories ($TPrate$, $Approx.\ TPrate$) and also in terms of the $GeoMean$. Paired t-tests were applied between each method and the proposed method using the $GeoMean$ results. These showed that the proposed method was significantly better than the other methods (α=0.05).

5.3 Conclusions

In this chapter, we presented a hierarchical decomposition method which constructs the hierarchy based on clustered and labelled trajectories using the similarity of trajectories. Different feature sets were applied to different subsets of the trajectories at the different levels formed the hierarchy.

The results showed that the proposed method had a significantly better performance compared to the methods presented in Chapters 3 and 4, state of art classification methods and unusual trajectory detection methods especially in terms of the unusual fish trajectory detection rate ($TPrate$). Besides, its high normal fish trajectory detection rate ($TNrate$) is helpful for marine biologists since it allows filtering out many normal trajectories with a low error rate and allows them to focus more on unusual trajectories which are important given that they have huge amounts of data. The proposed

algorithm's performance was also validated on the pedestrian trajectory data set. The results showed that the proposed hierarchical decomposition method was significantly better than the state of art methods. The experiments applied using different heuristics to classify a new trajectory also showed that the proposed heuristic (Section 5.1.2) is the best for unusual trajectory detection.

On the other hand, the proposed method is also computationally efficient at classifying a new trajectory as it is only based on distance calculations while traversing the built hierarchy. In Chapter 6, we investigate the performance of the proposed method on imbalanced data sets from various application areas and also with synthetic data sets to better understand its performance.

Chapter 6

Classifying Imbalanced Data Sets Using Similarity Based Hierarchical Decomposition

In recent years, classification with imbalanced data sets has become one of the key topics in machine learning and data mining due to its challenges especially for real-world applications. Data sets are dominated by normal examples where there is a small amount of unusual examples [87, 95, 123]. In class imbalance problems, usually, the samples are grouped into binary classes. The well-represented class is called the **majority class** and the under-represented class is called the **minority class**. In such a case, a problem usually occurs because traditional classification algorithms tend to be biased towards the majority class [89, 100]. Even though being imbalanced is not always a problem, for instance for a case where the classes are separable, imbalanced data sets usually contain overlapping regions where the prior probabilities of the two classes are almost equal [169]. Small disjuncts (samples of different classes that lie in the overlap region), and small sample size with high feature dimensionality [96] are frequently observed challenges in imbalanced data sets causing classification errors. Issues such as the feature selection criterion and/or the criterion to evaluate the performance are also important when dealing with imbalanced data sets. A comprehensive discussion about this is given in Section 2.3.1.

In this chapter, we apply the hierarchical decomposition method presented in Chapter 5 to imbalanced data set classification problems although not dedicated to any application field specifically. The proposed method with its alternative-1 version (see Table 5.3 for details) was applied to 20 public imbalanced data sets (Section 6.1.1) which are

Table 6.1: Pre-processing algorithms that are used.

Method	**Description**
Feature Selection	SFFS method [162] with the criterion of the mean of $TPrate$ and $TNrate$ was used as described in Section 4.1.4.
SMOTE [95]	Number of neighbours were selected as to make the data set's imbalance ratio (the number of minority class samples over majority class samples [87, 89]) equal to 1. If this was not possible (when imbalance ratio is too small), then we took the number of neighbours equal to the number of minority class samples which made the set as balanced as it can be.
Balanced Training [165]	This was only applied with Random Forest. All minority samples were kept, and subsets of the majority class were chosen randomly to build the decision trees. The number of majority class examples in the chosen subset was equal to the number of total minority class data samples.

from different fields and 300 synthetic data sets (Section 6.1.2). The proposed method is compared with popular supervised methods in combination with algorithmic level and data level approaches (see Section 2.3 for description). The comparison with synthetic data sets allowed to understand the performance of the proposed method in detail and in different conditions. The results showed that the proposed method's classification performance is better than the state of art methods. It is especially successful if the minority class is sparser than the majority class. It has accurate performance even when classes have sub-varieties and minority and majority classes are overlapping. Moreover, its performance is also good when the class imbalance ratio (the number of the minority class samples over the number of the majority class samples) is low, i.e. classes are more imbalanced.

6.1 Experimental Works and Results

To evaluate the classification performance of the proposed method, the experiments can be divided into two sections: *i)* experiments using public imbalanced data sets and *ii)* experiments using synthetic data sets. In both sections the pre-processing algorithms that are given in Table 6.1 were applied.

6.1.1 Experiments and Results using Public Imbalanced Data Sets

In this section, the data sets that were used and the state of the art imbalanced data classification algorithms that were applied to compare with the proposed hierarchical decomposition method are given. The results are evaluated in terms of different metrics. Moreover, different statistical tests were applied to assess the performance significance between the proposed method and the state of art methods.

6.1.1.1 Data Sets

Twenty popular imbalanced data sets were used to evaluate the effectiveness of the proposed method. The data sets are from different fields such as biology, physics, medicine, etc. The number of features (#$Fea.$), the total number of samples (#$Sam.$), the total number of minority and majority samples (#$Min.$, #$Maj.$), the imbalance ratio (IR=the number of minority class samples over majority class samples [87, 89]) and the corresponding citations for each data sets ($Ref.$) are given in Table 6.2. While choosing these data sets, we tried to cover the range of variety in the data sets. The selection was based on: unique data set name (as many of the data sets are combinations of the same data set but with different class combinations), a range of IR values (from 0.57 to 0.02), variation in the amount of class overlap (as given in the KEEL repository [170]), a varying number of samples (from 106 to 7420) and variation in the number of features (from 7 to 294).

The Hepato data set [4] originally had 4 classes, the Scene data set [173] originally had 6 classes and the Satimage data set [7] originally had 7 classes. For those data sets, we chose the smallest class as the minority class and collapsed the rest of the classes into one in order to obtain a two-class imbalanced data set. The other data sets (Pima [4, 5], Ionosphere [171, 7], Appendicitis [172] and data sets from the KEEL repository [170]) originally had binary classes or they were supplied as binary by the given references therefore we used those data sets as they are provided.

6.1.1.2 Results

To evaluate the proposed method 2-fold cross validation with the Appendicitis data set [172] (because this data set is small) and 5-fold cross validation for the rest of the data sets was performed. The data sets from the KEEL repository [170] were provided as 5-fold already. We used the testing sets of the corresponding data sets as provided but to obtain the validation sets (which is needed for feature selection especially) we ran-

Table 6.2: Summary of used imbalanced data sets.

Data Sets	$\#Fea.$	$\#Sam.$	$(\#Min, \#Maj.)$	IR	$Ref.$
Ionosphere	34	351	(126, 225)	~ 0.57	[7, 171]
Pima	8	768	(268, 500)	~ 0.50	[4, 5]
Vehicle1	18	4230	(1085, 3145)	~ 0.35	[170]
Vehicle2	18	4230	(1090, 3140)	~ 0.35	[170]
Vehicle0	18	4230	(995, 3235)	~ 0.31	[170]
Hepato	9	536	(116, 420)	~ 0.28	[4]
Appendicitis	7	106	(21, 85)	~ 0.25	[172]
Satimage	36	6435	(626, 5809)	~ 0.11	[7]
Glass2	9	1070	(85, 985)	~ 0.09	[170]
Ecoli-0-1-4-7_vs_2-3-5-6	7	1680	(145, 1535)	~ 0.09	[170]
Ecoli-0-1-4-7_vs_5-6	6	1660	(125, 1535)	~ 0.08	[170]
Cleveland-0_vs_4	13	865	(65, 800)	~ 0.08	[170]
Scene	294	2407	(177, 2230)	~ 0.08	[173]
Yeast-1_vs_7	7	2295	(150, 2145)	~ 0.07	[170]
Ecoli4	7	1680	(100, 1580)	~ 0.06	[170]
Oil	49	937	(41, 896)	~ 0.05	[6]
Glass5	9	1070	(45, 1025)	~ 0.04	[170]
Yeast5	8	7420	(220, 7200)	~ 0.03	[170]
Yeast-1-2-8-9_vs_7	8	4735	(150, 4585)	~ 0.03	[170]
Winequality-red-8_vs_6-7	11	4275	(90, 4185)	~ 0.02	[170]

domly divided the supplied training sets into 4-folds where the minority and majority class samples were distributed equally. This gave us data sets having equal amounts of samples for testing and validation with 3 times bigger training sets. Similarly, for the rest of the data sets using 5-fold cross validation, training, validation and testing sets were constituted randomly where minority and majority class samples were distributed equally.

The proposed hierarchical decomposition method is compared with the state of the art methods given in Table 6.3 in combination with feature selection and imbalanced data set handling approaches: SMOTE [95] and Balanced Training [165]. For each method the same training, validation and testing sets were used. Therefore, for the standard version of the methods (*kNN*, *C*4.5, *NB*, *SVM*, *RF_BT* and *Proposed*) and all versions of them with *SMOTE* the same training and testing sets were used while for the methods with feature selection validation sets were used as well to pick the best feature set for each method on each data set (except the proposed method which uses the training set to pick the best feature subset).

The results in terms of the average *GeoMean* (Eq. 2.7), the average *AGeoMean* (Eq. 2.8) and the *AUC* (Eq. 2.9) are given in Table 6.4, 6.5 and 6.6 respectively for each method and each data set. The average performances of each method in terms of evaluation metrics considering all data sets are also given in these tables. For each evaluation metric the standard deviations (considering the folds in cross validation) are also given after the $\pm$ sign. The best results in terms of each evaluation metric on each data set are emphasised in bold-face.

The results shows that the performance of the proposed method was the best on 13 of 20 data sets for *GeoMean*, 12 of 20 data sets for *AGeoMean* and 10 of 20 data sets for *AUC* out of 16 other classification methods. The next best method was SVM with (7, 8, 4) out of 20 data sets in terms of *GeoMean*, *AGeoMean* and *AUC* respectively. All other methods were worse. The proposed method generally performed better in terms of *GeoMean* if the IR is low (such as Winequality-red-8_vs_6-7 [170], Yeast-1-2-8-9_vs_7 [170] and Oil data sets [6]). The high performance in terms of *AGeoMean* also shows that the proposed method is good at majority class classification while as good as other methods for classification of minority class (can be infer from *GeoMean* results). Additionally, the proposed method performed well enough in terms of *AUC* which can be supported by the statistical test results given in Section 6.1.1.3. Average results over the 20 data sets also show that the proposed method is the best method for each of the three metrics.

Table 6.3: State or art methods and their combinations with pre-processing algorithms that are used for comparison.

Method	**Description**	**Abbreviation**
k-Nearest Neighbours	k={1, 2, 5, 10, 15, 20, 25} were used as the common parameters. For any k value which gave local maximum we applied intermediate k values as well. For instance, if we obtain the best performance when k=5 but the performance decreased sharply when k=10 then we tried k=6, 8 as well (which did not happen a lot).	kNN
kNN with Feature Selection		kNN_wFS
kNN with SMOTE		kNN_SMOTE
kNN with SMOTE and Feature Selection		kNN_SMOTE_wFS
C4.5	Quilan's C4.5 code was used. Percentage of incorrectly assigned samples at a node (confidence level) was taken as {0.05, 0.1, 0.2, 0.3}.	C4.5
C4.5 with SMOTE		C4.5_SMOTE
Naive Bayes	As distributions: the normal distribution, kernel density estimation with different kernels such as normal, box, Epanechnikov etc. were tested with equal prior probabilities.	NB
NB with Feature Selection		NB_wFS
NB with SMOTE		NB_SMOTE
NB with SMOTE and Feature Selection		NB_SMOTE_wFS
Support Vector Machine	As the kernel function, a radial basis function with varying kernel parameters was used. Hyper-planes were separated by Sequential Minimal Optimisation.	SVM
SVM with Feature Selection		SVM_wFS
SVM with SMOTE		SVM_SMOTE
SVM with SMOTE and Feature Selection		SVM_SMOTE_wFS
Random Forest with Balanced Training	A number of trees {10, 50, 100, 150, 200, 500, 1000} were tested and the trees were grown without pruning. For node splitting, the Gini index [166] was used.	RF_BT
RF with Balanced Training and Feature Selection		RF_BT_wFS
Hierarchical Decomposition	As the outlier detection parameter w {-1, -0.3, 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.9, 1, 1.5, 2, 2.5, 3} were tested.	Proposed

Table 6.4: Best results of each method in terms of the average $GeoMean$. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign.

Data Set	kNN	kNN wFS	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB	NB wFS	NB SMOTE	NB SMOTE wFS	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.54±0.01	0.47±0.04	0.69±0.01	0.63±0.03	0.52±0.09	0.66±0.01	0.66±0.01	0.66±0.01	0.68±0.01	0.64±0.01	**0.77±0.01**	0.68±0.08	**0.77±0.01**	0.63±0.06	0.72±0.06	0.61±0.05	0.70±0.05
Yeast-1-2-8-9_vs_7	0.20±0.28	0.55±0.21	0.75±0.07	0.49±0.13	0.19±0.27	0.72±0.01	0.20±0.29	0.36±0.05	0.49±0.12	0.46±0.05	0.75±0.06	0.62±0.10	0.69±0.02	0.66±0.13	0.71±0.01	0.56±0.02	**0.82±0.06**
Winequality-red-8_vs_6-7	0.10±0.22	0.00±0.00	0.23±0.31	0.35±0.34	0.10±0.22	0.41±0.23	0.11±0.24	0.59±0.10	0.56±0.15	0.45±0.28	0.62±0.20	0.71±0.11	0.76±0.08	0.65±0.12	0.77±0.05	0.60±0.07	**0.81±0.10**
Ecoli-0-1-4-7_vs_5-6	0.86±0.10	0.75±0.13	0.89±0.05	0.79±0.10	0.85±0.12	0.82±0.06	0.86±0.08	0.76±0.14	0.83±0.11	0.72±0.15	0.87±0.08	0.79±0.09	0.80±0.11	0.72±0.10	0.87±0.11	0.76±0.14	**0.94±0.06**
Cleveland-0_vs_4	0.36±0.34	0.28±0.39	0.57±0.35	0.74±0.17	0.32±0.44	0.32±0.07	0.84±0.12	0.63±0.36	0.78±0.17	0.73±0.10	**0.90±0.18**	0.77±0.12	0.85±0.05	0.81±0.08	0.87±0.18	0.62±0.35	0.85±0.08
Glass2	0.43±0.41	0.11±0.25	0.79±0.10	0.65±0.13	0.00±0.00	0.00±0.00	0.72±0.04	0.55±0.31	0.72±0.07	0.68±0.16	0.76±0.15	0.64±0.09	0.74±0.13	0.65±0.05	0.76±0.08	0.66±0.18	**0.81±0.10**
Yeast5	0.81±0.01	0.68±0.10	0.96±0.01	**0.97±0.01**	0.52±0.17	0.95±0.01	0.90±0.03	0.86±0.06	0.80±0.07	0.93±0.07	**0.97±0.01**	0.96±0.01	0.87±0.01	0.87±0.02	**0.97±0.01**	**0.97±0.01**	0.93±0.10
Yeast-1_vs_7	0.38±0.23	0.28±0.26	0.69±0.09	0.64±0.09	0.31±0.29	0.54±0.05	0.53±0.07	0.53±0.07	0.57±0.15	0.52±0.14	0.76±0.04	0.70±0.10	0.68±0.04	0.62±0.05	0.76±0.06	0.65±0.05	**0.80±0.16**
Vehicle2	0.90±0.03	0.78±0.03	0.91±0.03	0.84±0.09	0.64±0.03	0.81±0.03	0.84±0.03	0.88±0.06	0.85±0.03	0.87±0.08	**0.97±0.02**	0.90±0.03	0.96±0.03	0.88±0.04	**0.97±0.01**	0.91±0.04	**0.97±0.05**
Vehicle0	0.89±0.04	0.81±0.05	0.92±0.02	0.88±0.02	0.89±0.02	0.90±0.03	0.76±0.02	0.77±0.03	0.78±0.02	0.77±0.02	**0.96±0.01**	0.84±0.05	0.93±0.03	0.80±0.07	**0.96±0.01**	0.89±0.03	**0.96±0.05**
Glass5	0.74±0.43	0.53±0.50	0.83±0.16	0.87±0.16	0.00±0.00	0.00±0.00	0.66±0.01	0.66±0.01	0.39±0.54	0.52±0.48	0.92±0.13	0.90±0.05	0.88±0.13	**0.95±0.04**	0.89±0.06	0.92±0.06	0.88±0.13
Ecoli4	0.89±0.06	0.86±0.10	0.97±0.02	0.91±0.10	0.85±0.09	0.91±0.07	0.20±0.29	0.36±0.05	0.88±0.11	0.89±0.17	0.97±0.06	0.92±0.08	0.90±0.07	0.83±0.10	0.92±0.07	0.83±0.19	**0.99±0.02**
Ecoli-0-1-4-7_vs_2-3-5-6	0.78±0.12	0.74±0.08	0.89±0.05	0.80±0.08	0.80±0.15	0.84±0.07	0.11±0.24	0.59±0.10	0.78±0.14	0.82±0.05	0.91±0.08	0.77±0.08	0.78±0.08	0.72±0.08	0.87±0.09	0.79±0.09	**0.92±0.08**
Appendicitis	0.66±0.17	0.73±0.10	0.76±0.03	0.75±0.04	0.69±0.01	0.73±0.07	0.86±0.08	0.76±0.14	0.68±0.16	0.66±0.06	0.74±0.08	0.79±0.01	0.74±0.08	0.78±0.01	0.72±0.05	0.77±0.04	**0.83±0.07**
Hepato	0.68±0.04	0.58±0.11	**0.73±0.06**	0.65±0.07	0.67±0.03	**0.73±0.05**	0.84±0.12	0.63±0.36	**0.73±0.06**	0.70±0.05	**0.73±0.06**	0.70±0.05	0.45±0.01	0.69±0.07	0.73±0.03	0.69±0.07	**0.73±0.04**
Ionosphere	0.80±0.05	0.82±0.04	0.86±0.05	0.90±0.06	0.56±0.34	0.80±0.09	0.72±0.04	0.55±0.31	0.88±0.02	0.81±0.08	0.88±0.04	0.89±0.01	**0.92±0.03**	0.91±0.02	0.91±0.04	0.89±0.05	0.87±0.05
Oil	0.47±0.25	0.54±0.16	0.76±0.04	0.77±0.03	0.60±0.15	0.77±0.05	0.90±0.03	0.86±0.06	0.65±0.08	0.78±0.07	0.78±0.07	0.78±0.04	0.41±0.19	0.77±0.13	0.77±0.05	0.79±0.05	**0.82±0.08**
Pima	0.66±0.03	0.68±0.02	0.72±0.02	0.72±0.04	0.62±0.05	0.70±0.04	0.53±0.07	0.53±0.07	0.72±0.07	0.62±0.06	0.73±0.03	0.71±0.02	0.74±0.02	0.73±0.03	0.73±0.04	0.68±0.08	**0.79±0.05**
Satimage	0.81±0.02	0.66±0.06	0.89±0.02	0.76±0.02	0.61±0.07	0.82±0.01	0.84±0.03	0.88±0.06	0.85±0.02	0.84±0.01	**0.90±0.02**	0.86±0.01	0.56±0.03	0.83±0.02	0.88±0.02	0.85±0.02	0.84±0.10
Scene	0.40±0.14	0.33±0.06	0.66±0.04	0.53±0.04	0.42±0.01	0.67±0.03	0.76±0.02	0.77±0.03	0.55±0.08	0.61±0.02	0.53±0.13	0.67±0.06	0.68±0.05	0.70±0.10	**0.72±0.02**	0.64±0.05	0.61±0.13
AVERAGE	0.62±0.15	0.56±0.13	0.77±0.08	0.73±0.09	0.50±0.13	0.65±0.05	0.67±0.10	0.67±0.11	0.71±0.11	0.70±0.11	0.82±0.07	0.78±0.06	0.76±0.06	0.76±0.06	0.83±0.05	0.76±0.08	**0.84±0.08**

Table 6.5: Best results of each method in terms of the average $AGeoMean$. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign.

Data Set	kNN	kNN wFS	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB	NB wFS	NB SMOTE	NB SMOTE wFS	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.68±0.01	0.66±0.01	0.68±0.01	0.64±0.01	**0.81±0.03**	0.65±0.01	0.68±0.01	0.63±0.05	0.70±0.01	0.65±0.02	0.76±0.01	0.69±0.05	0.75±0.01	0.68±0.05	0.70±0.01	0.61±0.01	0.71±0.08
Yeast-1-2-8-9_vs_7	0.33±0.47	0.77±0.12	0.75±0.04	0.78±0.08	0.32±0.45	0.77±0.05	0.35±0.49	0.58±0.14	0.72±0.07	0.49±0.01	**0.79±0.02**	0.69±0.04	0.64±0.05	0.57±0.05	0.74±0.01	0.59±0.04	0.75±0.05
Winequality-red-8_vs_6-7	0.15±0.33	0.00±0.00	0.29±0.39	0.44±0.41	0.15±0.33	0.56±0.32	0.14±0.32	0.60±0.10	0.14±0.32	0.60±0.10	0.70±0.08	0.43±0.27	**0.72±0.11**	0.69±0.12	0.71±0.04	0.57±0.09	0.70±0.10
Ecoli-0-1-4-7_vs_5-6	0.92±0.05	0.85±0.08	0.92±0.02	0.82±0.08	0.90±0.06	0.88±0.04	0.90±0.05	0.77±0.12	0.90±0.06	0.78±0.09	0.90±0.04	0.74±0.13	0.79±0.06	0.66±0.09	0.87±0.02	0.79±0.13	**0.96±.0.04**
Cleveland-0_vs_4	0.46±0.42	0.33±0.46	0.62±0.35	0.80±0.12	0.35±0.48	0.25±0.04	0.88±0.06	0.65±0.37	0.84±0.09	0.76±0.06	**0.93±0.10**	0.76±0.09	0.85±0.09	0.79±0.05	0.89±0.09	0.63±0.36	**0.93±0.08**
Glass2	0.50±0.46	0.15±0.33	0.80±0.04	0.67±0.09	0.00±0.00	0.00±0.00	0.65±0.04	0.52±0.29	0.71±0.05	0.66±0.14	**0.84±0.07**	0.64±0.11	0.77±0.07	0.61±0.12	0.71±0.07	0.62±0.14	0.83±0.09
Yeast5	0.90±0.01	0.84±0.05	0.95±0.01	0.95±0.01	0.76±0.09	0.93±0.01	0.93±0.02	0.90±0.03	0.88±0.03	0.93±0.04	0.95±0.01	0.94±0.01	0.82±0.01	0.81±0.03	**0.96±0.01**	0.95±0.01	0.92±0.10
Yeast-1_vs_7	0.56±0.32	0.43±0.40	0.72±0.04	0.70±0.08	0.44±0.40	0.60±0.06	0.74±0.04	0.74±0.04	0.74±0.09	0.54±0.14	0.77±0.03	0.73±0.04	0.62±0.07	0.59±0.10	0.76±0.04	0.62±0.07	**0.88±0.10**
Vehicle2	0.92±0.02	0.85±0.06	0.92±0.02	0.85±0.08	0.79±0.01	0.78±0.03	0.86±0.03	0.86±0.06	0.88±0.03	0.85±0.08	**0.97±0.01**	0.88±0.04	0.96±0.02	0.86±0.04	**0.97±0.02**	0.90±0.04	0.93±0.07
Vehicle0	0.91±0.02	0.85±0.03	0.92±0.02	0.86±0.04	0.87±0.02	0.90±0.02	0.68±0.03	0.71±0.04	0.70±0.03	0.70±0.02	**0.97±0.01**	0.88±0.04	0.96±0.02	0.86±0.04	0.95±0.01	0.87±0.04	**0.97±0.02**
Glass5	0.76±0.43	0.56±0.51	0.85±0.12	0.92±0.09	0.00±0.00	0.54±0.49	0.39±0.53	0.13±0.28	0.39±0.53	0.53±0.49	**0.94±0.08**	0.90±0.16	0.93±0.08	0.92±0.06	0.84±0.08	0.89±0.08	0.83±0.06
Ecoli4	0.94±0.03	0.93±0.05	0.96±0.02	0.92±0.07	0.91±0.04	0.95±0.04	0.93±0.06	0.86±0.13	0.93±0.05	0.91±0.14	0.97±0.03	0.94±0.05	0.88±0.06	0.78±0.09	0.94±0.04	0.89±0.09	**0.99±0.01**
Ecoli-0-1-4-7_vs_2-3-5-6	0.88±0.06	0.85±0.04	**0.92±0.03**	0.84±0.03	0.89±0.07	0.89±0.04	0.87±0.05	0.76±0.12	0.78±0.17	0.82±0.06	**0.92±0.05**	0.82±0.06	0.82±0.05	0.70±0.02	0.87±0.08	0.82±0.06	0.90±0.09
Appendicitis	0.60±0.18	0.69±0.07	0.75±0.03	0.74±0.02	0.77±0.06	0.78±0.04	0.61±0.18	0.60±0.16	0.62±0.17	0.61±0.16	0.70±0.07	0.77±0.04	0.69±0.07	0.78±0.08	0.68±0.05	0.75±0.02	**0.91±0.08**
Hepato	0.78±0.03	0.71±0.06	0.77±0.04	0.70±0.04	0.61±0.04	0.71±0.10	0.78±0.03	0.67±0.05	0.75±0.04	0.70±0.04	0.74±0.06	0.71±0.08	0.34±0.01	0.70±0.06	0.81±0.02	0.67±0.08	**0.92±0.07**
Ionosphere	0.75±0.06	0.79±0.06	0.82±0.06	0.88±0.07	0.64±0.37	0.84±0.07	0.90±0.03	0.79±0.03	0.90±0.02	0.80±0.08	0.95±0.02	0.88±0.06	0.95±0.03	0.89±0.02	0.91±0.04	0.91±0.04	**0.99±0.01**
Oil	0.20±0.19	0.43±0.16	0.72±0.06	0.81±0.06	0.79±0.08	0.82±0.04	0.55±0.07	0.71±0.21	0.55±0.09	0.76±0.11	0.72±0.09	0.83±0.09	0.11±0.24	0.71±0.15	0.83±0.08	0.86±0.07	**0.99±0.01**
Pima	0.61±0.04	0.63±0.02	0.76±0.04	0.74±0.04	0.73±0.03	0.70±0.03	0.72±0.11	0.69±0.08	0.75±0.08	0.65±0.06	0.77±0.06	0.73±0.02	0.77±0.05	0.75±0.03	0.77±0.05	0.69±0.09	**0.88±0.11**
Satimage	0.75±0.03	0.57±0.07	0.89±0.03	0.71±0.03	0.76±0.04	0.84±0.01	0.86±0.03	0.87±0.02	0.85±0.03	0.86±0.01	**0.90±0.02**	**0.90±0.02**	**0.90±0.02**	0.87±0.02	0.89±0.03	0.86±0.03	**0.90±0.10**
Scene	0.30±0.12	0.23±0.05	0.63±0.06	0.49±0.05	0.67±0.06	0.70±0.03	0.35±0.07	0.60±0.04	0.46±0.10	0.62±0.02	0.55±0.13	0.71±0.06	0.60±0.05	0.74±0.10	0.72±0.05	0.65±0.08	**0.83±0.01**
AVERAGE	0.65±0.16	0.60±0.13	0.78±0.07	0.76±0.08	0.61±0.13	0.70±0.07	0.69±0.11	0.68±0.12	0.71±0.10	0.71±0.09	0.84±0.05	0.78±0.07	0.74±0.06	0.75±0.07	0.83±0.04	0.76±0.08	**0.89±0.06**

Table 6.6: Best results of each method in terms of the average AUC. The best results on each data set are emphasised in bold-face. The standard deviations considering the folds in cross validation are also given after the $\pm$ sign.

Data Set	kNN	kNN wFS	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB	NB wFS	NB SMOTE	NB SMOTE wFS	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS	Proposed
Vehicle1	0.54±0.04	0.56±0.03	0.65±0.02	0.54±0.04	0.56±0.07	0.61±0.04	0.66±0.03	0.67±0.01	0.68±0.03	0.64±0.02	0.66±0.16	0.65±0.04	0.76±0.12	0.62±0.03	0.72±0.05	0.64±0.04	0.69±0.02
Yeast-1-2-8-9_vs_7	0.50±0.00	0.50±0.00	0.73±0.09	0.64±0.07	0.50±0.00	0.51±0.00	0.53±0.03	0.48±0.01	0.61±0.03	0.47±0.02	0.80±0.11	0.67±0.05	0.82±0.15	0.57±0.08	0.71±0.04	0.53±0.12	**0.86±0.03**
Winequality-red-8_vs_6-7	0.50±0.00	0.00±0.00	0.51±0.06	0.54±0.13	0.50±0.00	0.55±0.12	0.48±0.07	0.61±0.07	0.62±0.11	0.52±0.17	0.64±0.21	0.62±0.10	0.72±0.19	0.71±0.16	**0.75±0.08**	0.63±0.12	0.74±0.10
Ecoli-0-1-4-7_vs_5-6	0.51±0.05	0.49±0.01	0.92±0.06	0.82±0.06	0.60±0.17	0.77±0.11	0.87±0.08	0.77±0.14	0.84±0.10	0.74±0.13	0.88±0.08	0.63±0.29	0.92±0.11	0.72±0.11	0.86±0.10	0.74±0.07	**0.96±0.13**
Cleveland-0_vs_4	0.50±0.01	0.50±0.01	0.64±0.19	0.60±0.11	0.55±0.14	0.37±0.13	0.85±0.14	0.72±0.17	0.80±0.18	0.74±0.10	**0.91±0.14**	0.70±0.21	0.89±0.17	0.53±0.35	0.84±0.12	0.62±0.17	**0.91±0.09**
Glass2	0.50±0.01	0.50±0.00	0.82±0.07	0.59±0.08	0.09±0.02	0.09±0.02	0.73±0.08	0.70±0.15	0.73±0.08	0.70±0.15	0.56±0.34	0.71±0.22	0.70±0.28	0.63±0.27	0.77±0.08	0.62±0.16	**0.83±0.05**
Yeast5	0.72±0.01	0.55±0.01	0.98±0.01	0.95±0.04	0.50±0.00	0.97±0.01	0.90±0.04	0.83±0.04	0.79±0.03	0.90±0.07	0.96±0.02	**0.99±0.01**	0.91±0.12	0.82±0.26	0.97±0.01	0.95±0.03	0.95±0.04
Yeast-1_vs_7	0.50±0.01	0.50±0.01	0.70±0.08	0.65±0.09	0.50±0.00	0.57±0.07	0.63±0.04	0.63±0.04	0.65±0.10	0.55±0.12	0.77±0.07	0.67±0.11	0.72±0.18	0.59±0.10	0.73±0.03	0.65±0.07	**0.81±0.03**
Vehicle2	0.76±0.04	0.75±0.05	0.86±0.02	0.83±0.07	0.67±0.04	0.81±0.03	0.84±0.03	0.89±0.06	0.85±0.03	0.87±0.08	**0.97±0.01**	0.83±0.06	0.96±0.02	0.86±0.06	**0.97±0.02**	0.91±0.04	**0.97±0.14**
Vehicle0	0.84±0.05	0.81±0.05	0.95±0.01	0.89±0.02	0.86±0.07	0.82±0.07	0.79±0.02	0.79±0.02	0.80±0.01	0.79±0.02	0.95±0.01	0.83±0.04	0.90±0.06	0.82±0.06	**0.97±0.01**	0.90±0.01	0.96±0.02
Glass5	0.50±0.01	0.48±0.03	0.88±0.17	0.80±0.21	0.09±0.02	0.09±0.02	0.69±0.27	0.48±0.07	0.69±0.27	0.70±0.25	**0.94±0.09**	0.82±0.27	0.89±0.15	0.91±0.16	0.89±0.05	0.89±0.05	0.89±0.16
Ecoli4	0.50±0.01	0.49±0.01	**0.99±0.01**	0.92±0.06	0.58±0.16	0.85±0.05	0.90±0.10	0.83±0.16	0.89±0.10	0.89±0.16	0.97±0.08	0.83±0.26	0.92±0.09	0.91±0.06	0.93±0.07	0.85±0.14	**0.99±0.10**
Ecoli-0-1-4-7_vs_2-3-5-6	0.49±0.01	0.49±0.01	0.89±0.05	0.82±0.03	0.58±0.14	0.78±0.11	0.85±0.08	0.74±0.11	0.79±0.12	0.83±0.05	0.90±0.09	0.64±0.11	0.78±0.15	0.73±0.10	0.87±0.08	0.80±0.10	0.93±0.04
Appendicitis	0.91±0.06	0.92±0.08	0.79±0.12	0.78±0.08	0.55±0.07	0.59±0.01	0.70±0.16	0.67±0.12	0.71±0.14	0.69±0.11	0.89±0.09	0.75±0.16	0.83±0.14	0.84±0.09	0.79±0.09	0.84±0.16	0.96±0.05
Hepato	0.35±0.05	0.29±0.07	0.66±0.07	0.55±0.10	0.76±0.02	**0.77±0.06**	0.76±0.04	0.66±0.06	0.74±0.05	0.70±0.05	0.67±0.07	0.54±0.12	0.20±0.27	0.66±0.08	0.77±0.04	0.72±0.07	0.76±0.03
Ionosphere	0.88±0.02	0.90±0.08	**0.96±0.03**	0.84±0.08	0.67±0.12	0.70±0.16	0.90±0.03	0.83±0.01	0.89±0.02	0.80±0.08	0.89±0.05	**0.96±0.03**	0.87±0.05	0.91±0.06	0.89±0.05	0.84±0.04	**0.96±0.02**
Oil	0.58±0.01	0.58±0.01	0.82±0.02	0.81±0.04	0.64±0.08	0.77±0.11	0.69±0.03	0.76±0.13	0.69±0.05	0.79±0.06	**0.97±0.01**	0.86±0.04	0.50±0.00	0.79±0.05	0.84±0.04	0.82±0.08	**0.97±0.01**
Pima	0.71±0.05	0.70±0.07	0.72±0.03	0.67±0.01	0.63±0.03	0.70±0.05	0.64±0.06	0.68±0.09	0.73±0.07	0.64±0.06	0.74±0.05	0.71±0.08	0.77±0.04	0.75±0.06	0.72±0.04	0.63±0.04	**0.79±0.06**
Satimage	0.90±0.01	0.85±0.01	**0.91±0.01**	0.88±0.01	0.80±0.05	0.88±0.05	0.85±0.02	0.85±0.02	0.85±0.02	0.84±0.01	0.90±0.01	0.80±0.02	0.50±0.01	0.82±0.01	0.87±0.01	0.81±0.02	0.85±0.02
Scene	0.58±0.01	0.53±0.01	0.73±0.02	0.63±0.06	0.49±0.05	0.61±0.09	0.58±0.03	0.64±0.01	0.61±0.05	0.63±0.02	0.60±0.01	0.62±0.06	**0.76±0.01**	0.62±0.06	0.73±0.02	0.63±0.03	0.64±0.002
AVERAGE	0.61±0.02	0.57±0.03	0.81±0.06	0.74±0.07	0.56±0.06	0.64±0.06	0.74±0.07	0.71±0.07	0.75±0.08	0.72±0.09	0.83±0.08	0.74±0.12	0.77±0.12	0.74±0.11	0.83±0.05	0.75±0.08	**0.87±0.06**

6.1.1.3 Statistical Tests

Statistical tests were applied using the $GeoMean$, $AGeoMean$, and AUC results to compare the different methods appropriately. Parametric and non-parametric tests were carried out as suggested in the literature [174, 175, 176] and as applied in other studies related to imbalanced data set classification such as [87]. We used Wilcoxon paired signed-rank test [177] and paired t-test as pairwise comparison tests to find out if there is a significant difference between the proposed method and any other method. As a multiple comparison test, we applied the Friedman test [176] to determine the statistical significance between methods given in Table 6.3. When we found a statistical difference between the methods and the proposed method, we applied the Holm post-hoc test [178] to test if the proposed method is significantly better than the others or not.

- **Wilcoxon paired signed-rank test [177]:** Ranks the absolute differences in performances of two classifiers for each data set. Then relate the signs in front of the ranks and compare the total ranks of positive (R^+) and negative (R^-) differences by finding the minimum of R^+ (Eq. 6.1) and R^- (Eq. 6.2) and comparing it with the appropriate critical value. If the minimum of R^+ and R^- is equal or less than a critical value, the difference between the classifiers is significant and the one having larger rank is better than the other [176].

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{6.1}$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{6.2}$$

  where d_i is the difference between the performance scores of the two classifiers on i-th data set.

- **Paired t-test [176]:** It considers the differences between the paired values of two classifiers for each data set by looking at the variation of corresponding values and produces p-value which determines how likely it is that the two values are from the same population [174, 176]. The paired values are the performances on each fold from the two compared algorithm. This test can be used to justify if the performances of the two algorithms are significantly different or not. The p-value determines if the comparison is significant or not and also indicates how

significant it is: If the proposed method is better, then the smaller the p-value, the more significantly better it is. Conversely, if any other method is better, then the smaller value of p shows how much better it is than the proposed method. For all tests, the significance level is taken as 0.05.

- **Friedman test with Iman-Davenport Extension [176]:** Methods are ranked on each data set according to their performance (best performance takes the lowest rank). For each classifier the sum of its ranks on all data set is calculated. Friedman's and Iman-Davenport statistics are calculated using the formulas given in Eq. 6.3 and Eq. 6.4. The statistics are compared with the corresponding value in the F-distribution table. If the value in the F-distribution table is smaller than the found statistics, then the null hypothesis (all classifiers perform the same and the observed differences are random) is rejected and this means that there is a significant difference between the compared methods.

$$\chi_F^2 = \Big(\frac{12}{Nk(k+1)} \sum_{j=1}^{k} (R_j)^2\Big) - 3N(k+1) \tag{6.3}$$

where R_j the sum of j-th method's ranks on all data sets. N is the total number of data sets. k is the total number of methods.

$$F_{ID} = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{6.4}$$

which is distributed according to F distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom.

- **Holm post-hoc test [178]:** Is based on the value z given in Eq. 6.5. The p-value is obtained from the normal distribution corresponds to z and the adjusted alpha (described below). A p-value smaller than the corresponding adjusted alpha means that the null hypothesis is rejected; meaning that there is a significant difference between the compared methods.

$$z_i = (R_{proposed} - R_i)/\sqrt{\frac{k(k+1)}{6N}} \tag{6.5}$$

where z_i is the value of i-th method, k refers to the number of methods (which is 17), N refers to the number of data sets (which is 20).

The results of Wilcoxon's Signed rank test on the *GeoMean*, *AGeoMean* and *AUC* data is given in Figure 6.1. In this figure, R^+ represents the rank of proposed method

Methods	GeoMean			AGeoMean			**AUC**		
	R^+	R^-	**Sig.**	R^+	R^-	**Sig.**	R^+	R^-	**Sig.**
kNN	210	0	yes	210	0	yes	208	2	yes
kNN wFS	210	0	yes	209	1	yes	190	0	yes
kNN SMOTE	166	24	yes	176	14	yes	149	22	yes
kNN SMOTE wFS	205	5	yes	200	10	yes	188	2	yes
C4.5	210	0	yes	205	2	yes	190	0	yes
C4.5 SMOTE	184	6	yes	206	4	yes	204	6	yes
NB	205	5	yes	209	1	yes	171	0	yes
NB wFS	206	4	yes	210	0	yes	171	0	yes
NB SMOTE	187	3	yes	210	0	yes	190	0	yes
NB SMOTE wFS	153	0	yes	209	1	yes	210	0	yes
SVM	107	46	yes	98	38	yes	130	23	yes
SVM wFS	187	23	yes	183	7	yes	187	3	yes
SVM SMOTE	151	20	yes	171	19	yes	167	23	yes
SVM SMOTE wFS	183	27	yes	204	6	yes	209	1	yes
RF BT	111	42	yes	186	24	yes	136	35	yes
RF BT wFS	192	18	yes	205	5	yes	171	0	yes

Figure 6.1: Comparing methods with the proposed method using Wilcoxon's Signed rank test using the *GeoMean*, *AGeoMean* and *AUC*. R^+ (Eq. 6.1) represents the rank of the proposed method and R^- (Eq. 6.2) represents the rank of the compared method while α is taken as 0.05. Significance (Sig.) is shown as "yes" if there is a significant difference, otherwise it is shown as "no".

and R^- represents the rank of the compared method. α is taken as 0.05. If the minimum of R^+ and R^- is equal or less than critical value (52, for 20 data sets), then the difference between the classifiers is significant and the one having larger rank is better than the other [176]. We show the significance as "yes" if there is a significant difference, otherwise show it as "no".

As seen from the results, according to Wilcoxon's Signed rank test the proposed method is significantly better than all other methods using all data sets in terms of all metrics.

The average ranks used in the computation of the Friedman test for the metrics *GeoMean*, *AGeoMean* and *AUC* are shown in Figure 6.2. In this figure the best rank (smallest one) is shown in bold. Additionally, The Friedman and Iman-Davenport statistics are given. For the calculation using *GeoMean*, the critical value of the F-distribution (16.304) is 2.010 when $\alpha = 0.05$, which is smaller than the Iman-Davenport (18.470) statistic meaning that the null hypothesis of Friedman (given above) is rejected by a high level of significance. Similarly, for *AGeoMean* and *AUC*, Iman-Davenport statistic are 10.810, and 16.200 respectively which are larger than 2.010 which also

AVERAGE RANK USING ALL DATASETS	*kNN*	*kNN wFS*	*kNN SMOTE*	*kNN SMOTE wFS*	*C4.5*	*C4.5 SMOTE*	*NB*	*NB wFS*	*NB SMOTE*
GeoMean	12.85	14.58	5.95	9.63	14.75	10.18	10.6	11.83	9.93
AGeoMean	11.43	12.98	6.53	9.35	11.75	9.48	10.23	12.58	9.9
AUC	13.05	13.8	5.45	9.48	14.65	11.42	9.6	10.82	8.85
	NB SMOTE wFS	***SVM***	***SVM wFS***	***SVM SMOTE***	***SVM SMOTE wFS***	***RF BT***	***RF BT wFS***	***Proposed***	
GeoMean	11.1	3.75	6.83	7.28	8.55	3.75	8.08	**3.4**	
AGeoMean	11.88	3.65	7.90	7.98	10.18	4.83	9.63	**2.78**	
AUC	10.9	4.6	8.95	6.73	9.18	4.05	8.9	**2.58**	

Friedman statistic for **GeoMean**= 157.72
Iman-Davenport statistic for **GeoMean**=18.47
F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance.**

Friedman statistic for **AGeoMean**= 116.01
Iman-Davenport statistic for **AGeoMean**= 10.81
F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance.**

Friedman statistic for **AUC**= 147.26
Iman-Davenport statistic for **AUC**=16.20
F (16,304) =2.01, α=0.05, **the null hypothesis of Friedman is rejected by a high level of significance**.

Figure 6.2: The average ranks used in the computation of the Friedman test for the metrics $GeoMean$, $AGeoMean$ and AUC respectively. Lower rank means better performance. The best performance is shown in bold.

means that the null hypothesis is rejected.

Since the Friedman test results showed a high significance, we applied the post-hoc Holm test. Figure 6.3 shows the Holm test results using the performance results in terms of the $GeoMean$, $AGeoMean$ and AUC respectively. In this figure, z_i is calculated as given in Eq. 6.5. The p-value is based on the normal distribution and Holm adjusted alpha (shown as Holm in this figure) equals to $0.005/i$. Hypothesis given as "rejected" means a significant difference between the compared methods and this happens if the p-value is smaller than the corresponding Holm value. Negative values of z means that the proposed method performed better than the compared method.

As seen in Figure 6.3, the proposed method is the best over all comparisons (all z values are negative) and it is significantly better than all methods except SVM_SMOTE, SVM_wFS, kNN_SMOTE, RF_BT and SVM in terms of the $GeoMean$, is significantly better than all methods except RF_BT, and SVM in terms of the $AGeoMean$, and is significantly better than all methods except kNN_SMOTE, SVM and RF_BT in terms of the AUC. In other words, the Holm test results show that the proposed method is

i	Methods	z_i	p_value	Holm	Hypothesis
16	C4.5	-7.1076	0.0001	0.0031	**Rejected for Proposed**
15	kNN wFS	-6.9981	0.0001	0.0033	**Rejected for Proposed**
14	kNN	-5.9178	0.0001	0.0036	**Rejected for Proposed**
13	NB wFS	-5.2759	0.0001	0.0038	**Rejected for Proposed**
12	NB SMOTE wFS	-4.8219	0.0001	0.0042	**Rejected for Proposed**
11	NB	-4.5088	0.0001	0.0045	**Rejected for Proposed**
10	C4.5 SMOTE	-4.2427	0.0001	0.0050	**Rejected for Proposed**
9	NB SMOTE	-4.0861	0.0001	0.0056	**Rejected for Proposed**
8	kNN SMOTE wFS	-3.8982	0.0001	0.0063	**Rejected for Proposed**
7	SVM SMOTE wFS	-3.2251	0.0013	0.0071	**Rejected for Proposed**
6	RF BT wFS	-2.9276	0.0034	0.0083	**Rejected for Proposed**
5	SVM SMOTE	-2.4266	0.0152	0.0100	Not Rejected
4	SVM wFS	-2.1448	0.0320	0.0125	Not Rejected
3	kNN SMOTE	-1.5969	0.1103	0.0167	Not Rejected
2	RF BT	-0.2192	0.8265	0.0250	Not Rejected
1	SVM	-0.2192	0.8265	0.0500	Not Rejected

(a)

i	Methods	z_i	p_value	Holm	Hypothesis
16	kNN wFS	-10.2000	0.0001	0.0031	**Rejected for Proposed**
15	NB wFS	-9.8000	0.0001	0.0033	**Rejected for Proposed**
14	NB SMOTE wFS	-9.1000	0.0001	0.0036	**Rejected for Proposed**
13	C4.5	-8.9750	0.0001	0.0038	**Rejected for Proposed**
12	kNN	-8.6500	0.0001	0.0042	**Rejected for Proposed**
11	NB	-7.4500	0.0001	0.0045	**Rejected for Proposed**
10	SVM SMOTE wFS	-7.4000	0.0001	0.0050	**Rejected for Proposed**
9	NB SMOTE	-7.1250	0.0001	0.0056	**Rejected for Proposed**
8	RF BT wFS	-6.8500	0.0001	0.0063	**Rejected for Proposed**
7	C4.5 SMOTE	-6.7000	0.0001	0.0071	**Rejected for Proposed**
6	kNN SMOTE wFS	-6.5750	0.0001	0.0083	**Rejected for Proposed**
5	SVM SMOTE	-5.2000	0.0001	0.0100	**Rejected for Proposed**
4	SVM wFS	-5.1250	0.0001	0.0125	**Rejected for Proposed**
3	kNN SMOTE	-3.7500	0.0002	0.0167	**Rejected for Proposed**
2	RF BT	-2.0500	0.0404	0.0250	Not Rejected
1	SVM	-0.8750	0.3816	0.0500	Not Rejected

(b)

i	Methods	z_i	p_value	Holm	Hypothesis
16	C4.5	-7.5617	0.0001	0.0031	**Rejected for Proposed**
15	kNN wFS	-7.0294	0.0001	0.0033	**Rejected for Proposed**
14	kNN	-6.5597	0.0001	0.0036	**Rejected for Proposed**
13	C4.5 SMOTE	-5.5421	0.0001	0.0038	**Rejected for Proposed**
12	NB SMOTE wFS	-5.2133	0.0001	0.0042	**Rejected for Proposed**
11	NB wFS	-5.1664	0.0001	0.0045	**Rejected for Proposed**
10	NB	-4.3992	0.0001	0.0050	**Rejected for Proposed**
9	kNN SMOTE wFS	-4.3209	0.0001	0.0056	**Rejected for Proposed**
8	SVM SMOTE wFS	-4.1331	0.0001	0.0063	**Rejected for Proposed**
7	SVM wFS	-3.9922	0.0001	0.0071	**Rejected for Proposed**
6	RF BT wFS	-3.9609	0.0001	0.0083	**Rejected for Proposed**
5	NB SMOTE	-3.9296	0.0001	0.0100	**Rejected for Proposed**
4	SVM SMOTE	-2.5988	0.0094	0.0125	**Rejected for Proposed**
3	kNN SMOTE	-1.8004	0.0718	0.0167	Not Rejected
2	SVM	-1.2681	0.2048	0.0250	Not Rejected
1	RF BT	-0.9237	0.3557	0.0500	Not Rejected

(c)

Figure 6.3: Holm test results for the comparison between proposed method and the other methods using the *a)* GeoMean, *b)* AGeoMean *c)* AUC results.

significantly better than $\{11 \text{ of } 16\}$, $\{14 \text{ of } 16\}$ and $\{13 \text{ of } 16\}$ methods when results of the *GeoMean*, *AGeoMean* and *AUC* are used respectively.

In addition to those statistics, the paired t-test was applied to see how well the proposed method performs compared to each other method for each data set considering the performances in each cross validation fold. We used the results of the *GeoMean* as it had the worst statistics for the proposed method when the Holm test was applied (it was not significantly better than 5 of 16 methods). The paired t-test results between each method and the proposed method for *GeoMean* is given in Table 6.7 in terms of p-value. In this table, a p-value equal or smaller than 0.05 means there is a significant difference. Results showing a significant advantage to the proposed method are shown in bold-face and results showing significantly worse performance by the proposed method are shown in italics (though there are no such instances). High values of p (> 0.5) mean that the two methods performed nearly the same. Mid-values of p ($0.05 < p \leq 0.5$) mean that the proposed method performed better for each fold, but the performance of the other method was also very close to the proposed method for at least one fold.

As seen from paired t-test results (Table 6.7), the proposed method performed significantly better than the rest of the methods in 94 tests out of 320 tests when each data set and pairs of methods are considered separately. On the other hand, it performed worse (but not significantly worse) than another method in 36 tests (out of 320 tests). The proposed method never had significantly better performance than *SVM* and *RF_BT*. However, it performed significantly better than *kNN_SMOTE*, *kNN_SMOTE_wFS*, *NB_SMOTE_wFS* and *SVM_wFS* 2, 3, 4 and 3 times out of 20 respectively.

6.1.1.4 Summary of Experiments with Public Imbalanced Data Sets

In this section, we compared the performance of the proposed hierarchical decomposition method with state of art methods using 20 different public imbalanced data set. The evaluation was done in terms of the *GeoMean*, *AGeoMean* and *AUC*. The proposed method performed the best in all metrics. Statistical tests were also applied to the results. The Wilcoxon's Signed rank test and Friedman test with Iman-Davenport extension showed that the proposed method is significantly better than the rest for all metrics. However, the Holm test and the paired t-test showed that the proposed method is not significantly better than all methods. The paired t-test showed that the proposed method is not significantly better than *SVM* and *RF_BT* although it is better on average

Table 6.7: Paired t-test results between each method and the proposed method using the $GeoMean$ results.

Data Set	kNN	kNN wFS	kNN SMOTE	kNN SMOTE wFS	C4.5	C4.5 SMOTE	NB	NB wFS	NB SMOTE	NB SMOTE wFS	SVM	SVM wFS	SVM SMOTE	SVM SMOTE wFS	RF BT	RF BT wFS
Vehicle1	0.10	**0.02**	0.87	0.15	0.33	0.38	0.52	0.48	0.67	0.27	0.37	0.46	0.27	**0.04**	0.12	**0.02**
Yeast-1-2-8-9 vs_7	0.24	0.24	0.58	0.10	0.23	0.23	0.24	0.11	0.24	0.14	0.56	0.33	0.15	0.46	0.20	0.14
Winequality-red-8 vs 6-7	**0.01**	**0.00**	**0.03**	0.06	**0.01**	**0.03**	**0.00**	**0.01**	**0.01**	0.07	0.18	0.15	0.56	0.11	0.55	**0.00**
Ecoli-0-1-4-7 vs 5-6	0.17	**0.05**	0.20	**0.03**	0.14	**0.00**	0.06	0.09	0.11	0.06	0.18	**0.02**	**0.01**	**0.01**	0.14	**0.04**
Cleveland-0 vs 4	**0.03**	**0.02**	0.19	0.35	**0.03**	**0.00**	0.79	0.27	0.46	0.14	0.68	0.41	0.99	0.61	0.86	0.25
Glass2	0.10	**0.00**	0.09	0.11	**0.00**	**0.00**	0.11	0.14	0.07	0.22	0.36	0.06	0.11	**0.05**	0.26	0.19
Yeast5	0.17	**0.00**	0.53	0.50	0.10	0.61	0.61	0.33	**0.02**	0.97	0.51	0.59	0.37	0.48	0.47	0.50
Yeast-1_vs_7	**0.03**	**0.04**	0.08	0.20	**0.05**	**0.03**	**0.05**	**0.05**	0.13	**0.05**	0.54	0.38	0.13	**0.04**	0.63	**0.05**
Vehicle2	0.08	**0.00**	0.13	**0.04**	**0.00**	**0.00**	**0.01**	0.13	**0.02**	0.12	0.90	0.08	0.93	0.06	0.71	0.19
Vehicle0	**0.01**	**0.01**	0.08	0.06	**0.04**	0.06	**0.00**	**0.00**	**0.00**	**0.00**	0.91	**0.03**	0.47	**0.03**	0.97	0.08
Glass5	0.56	0.18	0.68	0.91	**0.00**	**0.00**	0.13	**0.00**	0.13	0.14	0.70	0.79	0.97	0.35	0.93	0.58
Ecoli4	**0.04**	0.06	0.33	0.16	**0.03**	0.09	0.15	0.10	0.11	0.26	0.50	0.13	0.08	**0.02**	0.09	0.15
Ecoli-0-1-4-7 vs 2-3-5-6	0.14	**0.04**	0.28	**0.03**	0.19	**0.00**	**0.00**	**0.04**	**0.03**	**0.04**	0.81	**0.00**	**0.04**	**0.02**	0.23	**0.06**
Appendicitis	0.26	0.13	0.25	0.12	0.19	**0.00**	0.29	0.19	0.27	0.17	0.08	0.56	0.08	0.55	0.06	0.19
Hepato	0.19	0.06	0.95	0.08	**0.01**	0.86	0.42	**0.03**	0.98	0.11	0.96	0.37	**0.00**	0.24	0.85	0.43
Ionosphere	0.08	0.19	0.67	0.58	0.13	0.15	0.30	0.04	0.68	0.24	0.86	0.43	0.17	0.36	0.30	0.78
Oil	**0.04**	**0.01**	0.17	0.30	**0.05**	0.34	**0.03**	0.19	**0.04**	0.24	0.39	0.39	**0.01**	0.44	0.22	0.50
Pima	**0.00**	**0.01**	**0.05**	0.06	**0.01**	**0.00**	**0.00**	**0.02**	0.13	**0.03**	0.08	0.09	**0.05**	0.08	0.09	**0.02**
Satimage	0.58	**0.03**	0.32	0.12	**0.02**	0.59	0.68	0.90	0.84	0.97	0.28	0.65	**0.01**	0.84	0.44	0.92
Scene	0.13	**0.03**	0.42	0.17	0.09	0.38	0.16	0.59	0.56	0.98	0.25	0.45	0.42	0.39	0.18	0.77

for the majority of the results.

6.1.2 Experiments and Results with Synthetic Data Sets

To show the proposed method's performance in detail and understand when it performs better than the other methods synthetic imbalanced data sets were also used. The experiments were applied with data sets generated using Gaussian Mixture Models (GMM) with;

- Different number of features: 2, 5 and 10,
- Different imbalance ratios:
  0.67 (300 samples from majority class, 200 samples from minority class),
  0.33 (300 samples from majority class, 100 from minority class),
  0.17 (300 samples from majority class, 50 samples from minority class)
- Different combinations for the standard deviation of majority and minority class distributions.

For both the majority and minority classes, a mixture of two equally weighted normal distributions was created as the baseline data set (see Figure 6.4a for 2 features). To create other data sets, we changed the covariance of the components for each class by multiplying the variance of each component with a constant α coefficient while keeping the mean of each component constant. Then, we sampled the same number of samples with the baseline data set for both the majority class and the minority class. For small values of α, the majority and minority classes are tighter and separable as two different classes whereas for the bigger values of α, the data sets overlap with the different components of classes and the classes themselves are sparser. In total, we obtained 100 different data sets while taking all pairs of α= {1, 2, 4, 8, 16, 32, 64, 128, 256, 512} for minority and majority classes. For the baseline data set, $\alpha_{minority}$ and $\alpha_{majority}$ are equal to 1 and the mean and co-variance of distributions are selected randomly. Figure 6.4 shows examples of data set for the same set of class centres and for different combinations of α with 2 features.

6.1.2.1 Results

For all experiments in this section, $SFFS$ (Section 4.1.4) is applied to choose the best feature subset for fair comparison since there is no prior information about features. All the experiments were run with proposed method, SVM_wFS, NB_wFS,

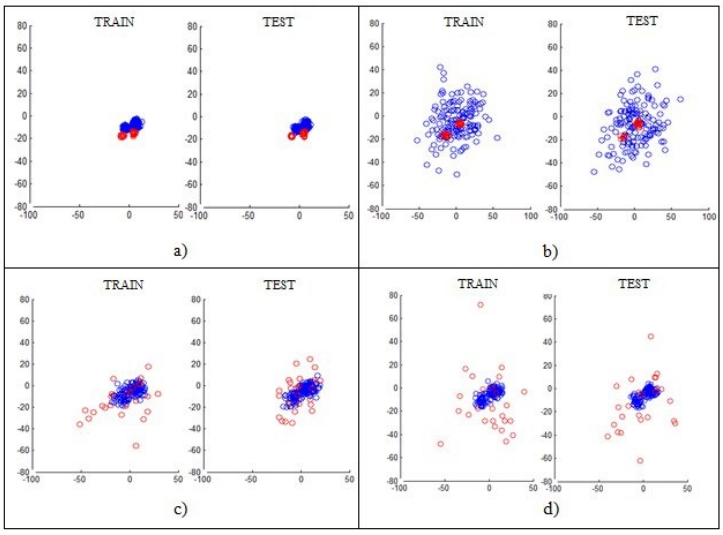


Figure 6.4: Examples of train-test pairs when the number of features is 2. Samples belong to majority class is shown with blue while samples belong to minority class is shown with red. Data sets having *a)* $\alpha_{minority}$=1 and $\alpha_{majority}$=1, *b)* $\alpha_{minority}$=4 and $\alpha_{majority}$=128, *c)* $\alpha_{minority}$=256 and $\alpha_{majority}$=8 and *d)* $\alpha_{minority}$=512 and $\alpha_{majority}$=4 for the same set of class centres. The class centres are varied on each cross validation fold.

RF_BT_wFS, *SVM_SMOTE_wFS*, and *NB_SMOTE_wFS* with the settings given in Table 6.3. The training, validation and test sets consist of 50%, 25% and 25% of the samples. The majority class samples and minority class samples were distributed appropriately to each set. All experiments were repeated **30 times** with different data instances. Therefore, for each fold the centres of the classes are also varied.

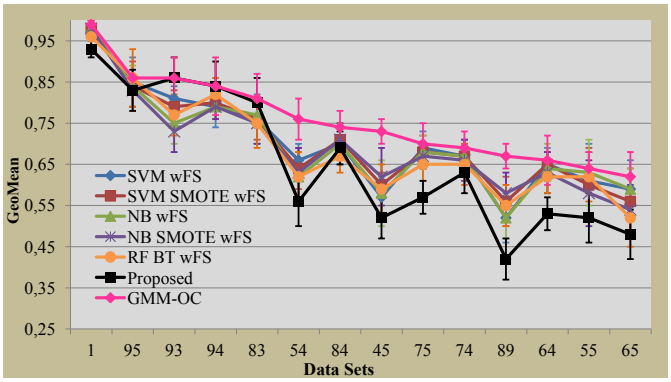
The performance of the proposed hierarchical decomposition method and comparison methods are shown in Figure 6.5 with 16 of the 100 pairs of α values which are enough to show the overall behaviour of methods with different numbers of features as 2, 5 and 10 with 300 samples from the majority class and 50 samples from the minority class. The *GeoMean* is used to show the results as it showed the worst performance (but still better than other methods) for the proposed method in Section 6.1.1.2.

As well as classifying each sample according to the proposed method or comparison methods, we also calculated the posterior probabilities of test samples using the GMM that the corresponding data set was created from including all training, validation and test samples and we then classify each test sample according to the highest posterior probability while taking the prior probabilities equally. In other words, we calculated the class decisions using the GMM that the data set is created from itself as

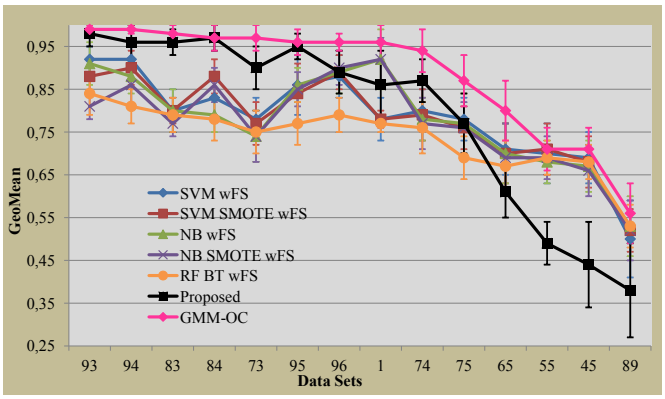
a classifier, which should model the **asymptotic performance**. We named this **GMM as the optimal classifier (***GMM_OC***)** which should show the **maximum achievable performance** for a given data set. The optimal classifier helps us to understand how difficult it is to classify the data set. For instance, a low value of the geometric mean for GMM_OC means that the data distributions are overlapping. Conversely, high values mean separable classes whose classification should be easier. The results of *GMM_OC* are also given in Figure 6.5 where the results are sorted from best performance to worse performance of *GMM_OC*.

The performance in terms of average *GeoMean* of the proposed method compared to other methods is summarised in Figure 6.6. In this figure, different data sets are grouped by their $\alpha_{minority}$ and $\alpha_{majority}$ ratios (for instance $\alpha_{minority}$=1, $\alpha_{majority}$=1 and $\alpha_{minority}$=512, $\alpha_{majority}$=512 are put into same group). For each group, the number of experiments (each experiment consist of 30 data folds over the same data set) that the proposed method performed better (the proposed method had the best performance on average over all folds) is given over the total number of experiments for each group. The total performance of the proposed method over total number of data sets using each feature set is also given as TOTAL. The results where proposed method performed better in the majority of the experiments is coloured with light green. The results where proposed method performed worse than at least one other method in the majority of the experiments is coloured with orange and the results where the proposed method performed as well as the other methods (within $\pm$0.02 of each other) are shown with pink colour.

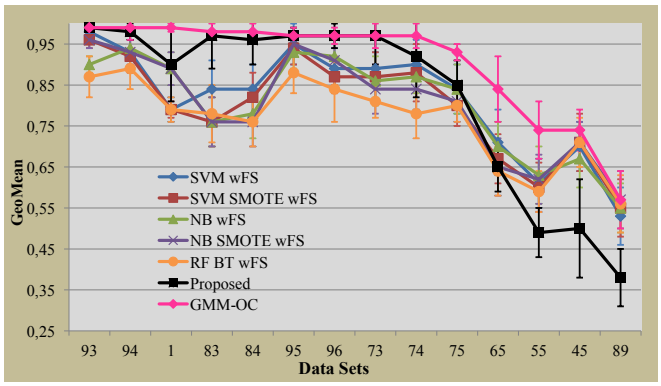
Based on the experimental results in Figures 6.5 and 6.6, independently of the number of features, the performance of the proposed method increases when $\alpha_{minority}$ is larger than $\alpha_{majority}$ meaning that the minority class is sparser than the majority class compared to data sets where the majority class is sparser than the minority or when they are equal. For the data sets where $\alpha_{minority}$ and $\alpha_{majority}$ are equal (or very close) and large enough (such as 128, 256) the minority class becomes inliers instead of being outliers. Therefore, the proposed method fails. On the other hand, when the number of features is increased even with a low ratio of $\alpha_{minority}$ to $\alpha_{majority}$, the proposed method performs better (mostly the best). It performs similarly to other methods with a high value of $\alpha_{minority}$ and $\alpha_{majority}$ ratios, and it performs worse than the rest for small values of $\alpha_{minority}$ and $\alpha_{majority}$ ratios especially when the number of features is small such as 2. Considering all generated data sets, the proposed method performs better than the rest of the methods when the ratio of $\alpha_{minority}$ and $\alpha_{majority}$ is at least



(a)



(b)



(c)

Data set 1: $\alpha_{minority}$=1 $\alpha_{majority}$=1	**Data set 64:** $\alpha_{minority}$=64 $\alpha_{majority}$=8	**Data set 75:** $\alpha_{minority}$=128 $\alpha_{majority}$=16	**Data set 93:** $\alpha_{minority}$=512 $\alpha_{majority}$=4
Data set 45: $\alpha_{minority}$=16 $\alpha_{majority}$=16	**Data set 65:** $\alpha_{minority}$=64 $\alpha_{majority}$=16	**Data set 83:** $\alpha_{minority}$=256 $\alpha_{majority}$=4	**Data set 95:** $\alpha_{minority}$=512 $\alpha_{majority}$=16
Data set 54: $\alpha_{minority}$=32 $\alpha_{majority}$=8	**Data set 73:** $\alpha_{minority}$=128 $\alpha_{majority}$=4	**Data set 84:** $\alpha_{minority}$=256 $\alpha_{majority}$=8	**Data set 94:** $\alpha_{minority}$=512 $\alpha_{majority}$=8
Data set 55: $\alpha_{minority}$=32 $\alpha_{majority}$=16	**Data set 74:** $\alpha_{minority}$=128 $\alpha_{majority}$=8	**Data set 89:** $\alpha_{minority}$=256 $\alpha_{majority}$=256	**Data set 96:** $\alpha_{minority}$=512 $\alpha_{majority}$=32

Figure 6.5: The best results of methods in terms of the average of $GeoMean$ for 16 different data sets created by different values of α for the minority and majority classes. The error bars show the standard deviations of the performance considering the 30 data folds for each data set. The number of features is *a)* 2, *b)* 5, *c)* 10, and the number of samples for the majority class is 300 while the number of samples in the minority class is 50. The data sets given on the horizontal axis refer to the index number of data set given in the legend which uses different combinations of $\alpha_{minority}$ and $\alpha_{majority}$ and are sorted by the performance of GMM_OC by decreasing order.

$\alpha_{minority}/\alpha_{majority}$	2 features	5 features	10 features
1:512	0/1	0/1	0/1
1:256	0/2	0/2	0/2
1:128	0/3	0/3	0/3
1:64	0/4	0/4	1/4
1:32	0/5	0/5	2/5
1:16	0/6	0/6	3/6
1:8	0/7	0/7	4/7
1:4	0/8	2/8	4/8
1:2	0/9	3/9	5/9
1:1	1/10	4/10	5/10
2:1	3/9	4/9	5/9
4:1	2/8	3/8	4/8
8:1	2/7	4/7	4/7
16:1	2/6	4/6	5/6
32:1	3/5	4/5	5/5
64:1	3/4	4/4	4/4
128:1	3/3	3/3	3/3
256:1	2/2	2/2	2/2
512:1	1/1	1/1	1/1
TOTAL	22/100	38/100	57/100

Figure 6.6: Summary of the performance in terms of the *GeoMean* for different feature sets having 100 different data sets for each, grouped by $\alpha_{minority}$ and $\alpha_{majority}$ ratios. Orange columns show when the proposed method performed worse, green columns show when the proposed method performed better, pink columns show when the proposed method performed about equal (see text for more detail).

32 with 2 features, at least 8 with 5 features and at least 0.0625 with 10 features. Those ratios (32, 8 and 0.0625) suggest that with more features, it is possible to have better performance by the proposed method even with low $\alpha_{minority}$ and $\alpha_{majority}$ ratios, meaning less sparse minority class data.

Observing that the proposed method potentially performs better than the rest of the methods when $\alpha_{minority} > \alpha_{majority}$, the methods with different imbalance ratios ($N_{minority}/N_{majority}$, where N represents the number of samples) were also compared. Imbalance ratios 0.67, 0.33 and 0.17 were used where the majority class has 300 samples and the minority class has 200, 100 and 50 samples respectively. We also varied the ratio of $\alpha_{minority}$ and $\alpha_{majority}$ as 32, 64 and 128. 30 trials were run for each experiment as well.

In Figure 6.7, similar performances of the methods (i.e. when within $\pm$ 0.02) are shown as "ALL". For 2 features, in the ALL cases, the methods achieved approximately 0.70 average *GeoMean*. For 5 and 10 features, ALL cases were obtained when the performances of methods were over 0.96 for average *GeoMean*. For the cases when the proposed method performed substantially better than the rest of the methods, we stated how much better it performed compared to the next best method with the performance

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$=0.67	Proposed (+0.06)	Proposed (+0.06)	Proposed (+0.07)
$N_{minority}/N_{majority}$=0.33	ALL	Proposed (+0.03)	Proposed (+0.04)
$N_{minority}/N_{majority}$=0.17	ALL	Proposed (+0.03)	Proposed (+0.05)

(a)

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$=0.67	SVM_SMOTE_wFS NB_SMOTE_wFS (-0.06)	SVM_SMOTE_wFS NB_SMOTE_wFS (-0.06)	ALL
$N_{minority}/N_{majority}$=0.33	ALL	ALL	ALL
$N_{minority}/N_{majority}$=0.17	Proposed (+0.09)	Proposed (+0.16)	Proposed (+0.06)

(b)

	$\alpha_{minority}/\alpha_{majority}$= 32	$\alpha_{minority}/\alpha_{majority}$= 64	$\alpha_{minority}/\alpha_{majority}$= 128
$N_{minority}/N_{majority}$=0.67	ALL	ALL	ALL
$N_{minority}/N_{majority}$=0.33	ALL	ALL	ALL
$N_{minority}/N_{majority}$=0.17	Proposed (+0.12)	Proposed (+0.13)	ALL

(c)

Figure 6.7: Best classification performance of methods in terms the average *GeoMean* using data sets with different imbalance ratios ($N_{minority}/N_{majority}$) when $\alpha_{minority}$ is $\{256, 256, 512\}$ and $\alpha_{majority}$ is $\{8, 4, 4\}$ which makes the ratios equal to 32, 64 and 128 respectively. Number of features is *a)* 2, *b)* 5, *c)* 10. ALL means that all classifiers had essentially equal performance.

difference in terms of average *GeoMean* after "+" sign. Similarly, if the performance of the proposed method was worse than any method then the name of the best classifier with the performance difference in terms of average *GeoMean* is given after "-" sign. For example, for 10 features when $N_{minority}/N_{majority}$=0.17 and $\alpha_{minority}/\alpha_{majority}$=32 the proposed method performed 0.96 in terms of average *GeoMean* while the next best method was *SVM* with 0.84. Therefore, this is shown as *Proposed* $(+0.12)$. On the other hand, for 5 features, $N_{minority}/N_{majority}$=0.67 and $\alpha_{minority}/\alpha_{majority}$=32, SVM_SMOTE_wFS and NB_SMOTE_wFS performed the best with 0.96 in terms of average *GeoMean* while the proposed method performed 0.90 and this is shown as $SVM\_SMOTE\_wFS\,NB\_SMOTE\_wFS\,(-0.06)$. To sum up these results, the proposed method performed better when the number of features is increased (for instance, for $N_{minority}/N_{majority}$=0.67 and $N_{minority}/N_{majority}$=0.32, it performed 0.83 with 2 features, 0.90 with 5 features and 0.97 with 10 features). Additionally, it performed better than the other methods when the imbalance ratio is low such as 0.17.

6.2 Conclusions

In this chapter, the hierarchical decomposition method introduced in Chapter 5 is presented as an imbalanced data set classification method. Outlier detection was used to detect minority class samples assuming that the minority class samples in each cluster are outliers by cardinality or by their distance to the cluster centre. The key observation and the justification for using a hierarchy was that some features allow partitioning of some samples which then allows other features to be useful on the remaining samples.

Compared to other imbalanced data set classification methods in the literature (Section 2.3), the proposed method does not need the support of any cost function, algorithmic or data level algorithm to handle imbalanced data sets. On the other hand, since it does not use all the data samples to build up the hierarchy at each level, it can be considered close to **bagging**. In our case, the bags are defined by the performance of the classifier and building up the hierarchy continues with the incorrectly classified samples in contrast to random subsets as happens in bagging. Moreover, it is different from **boosting** by using a subset of data in addition to not using a weight to support the classification of misclassified samples.

The computational complexity during **training** of the proposed method is much more than that of the other methods which can be seen as a shortcoming. To decrease the training time complexity, feature selection can be implemented in parallel on a task farming architecture with the methodology given in [2]. However, more importantly, the proposed method's **testing** complexity is as efficient as the other methods, as it requires only a few distance calculations between the closest clusters at each level and the new data point.

In conclusion, the proposed hierarchical decomposition method is successful at classifying imbalanced data sets even though the majority and minority classes contain varieties, and classes overlap which is frequently seen in real life applications. It performs much better if the minority class samples are sparse compared to the majority class samples where popular classification methods generally fail. It also performs well when the ratio between minority and majority class samples is low. In Chapter 7, we investigate the integration of the proposed hierarchical decomposition method with active learning where a substantial performance increase can be obtained while using less labelled training data compared to learning from larger labelled data sets. Additionally, we address active learning with feature selection.

Chapter 7

Active Learning with Imbalanced Data Sets

Thanks to technological improvements, data acquisition is not as difficult as it was and this has increased the amount of data in many domains. However, even though the amount of data is bigger, much of the data is unlabelled and labelled data is very limited or even does not exist. The traditional way of labelling data is to ask experts to label them which is difficult and very time consuming. Similarly, in our project one of the most time consuming parts was labelling the fish trajectories as normal or unusual. Moreover, the proposed methods (Chapters 3, 4 and 5) require labelled training data as they are all supervised learning methods. Therefore, utilising unlabelled trajectories in addition to already labelled trajectories for learning is necessary. A possible solution can be using active learning (AL), where a few informative unlabelled instances are chosen to be labelled by the expert and used to update the model. The labelling cost is decreased since the experts only need to label the selected instances. Additionally, in [138], it is also shown that active learning can achieve higher classification performance with fewer training instances compared to passive learning which uses all training data and requires the labels of all training data for learning.

In this chapter, we investigate how to integrate the proposed hierarchical decomposition method (Chapter 5) with active learning. We claim that through using a proper active learning query strategy and corresponding feature subset combinations at different active learning iterations, substantial performance (see definition in Section 7.3.3.1) can be obtained with less training data.

Since the proposed hierarchical decomposition method includes feature selection and to the best of our knowledge active learning with feature selection has never been

investigated as deeply as here, we first address active learning with feature selection (especially for imbalanced data sets) using different classifiers (Section 7.2.2). Then, we integrate the hierarchical decomposition method with active learning.

7.1 Active Learning

Active learning (AL) considers the cost of data labelling [137]. When unlabelled data is abundant and labelled data is limited, active learners seek to choose the most informative unlabelled training instances with a query strategy. Selected instances are first labelled by an expert and then combined with previously labelled data to update a model. The aim is to maximise the learning performance while decreasing the cost of labelling [137, 138, 139, 179]. As mentioned in Section 2.5, active learning has 3 subtypes. In this thesis, we consider pool based active learning and refer to it as active learning.

The basic working principle of pool based active learning is:

- Train a model using the existing labelled data.
- Using the model and a query strategy, select some instances from the unlabelled data.
- Label the selected samples (informative samples).
- Combine the newly labelled instances with the previously labelled training set and repeat all these steps until you reach a stopping criterion (such as a decrease in classification performance, having a certain amount of labelled training data, etc.).

This is illustrated in Figure 7.1.

As seen in Section 2.5, different studies proposed specific *AL* query strategies tuned to the application that they tackle. However, there are also very popular general query selection strategies such as *i)* uncertainty, *ii)* information density, and *iii)* maximum probability. Those popular query strategies are successful and recent strategies are actually based on them. However, no one query strategy is the best for all data sets [138, 139]. We summarised the 3 key query strategies in addition to random selection:

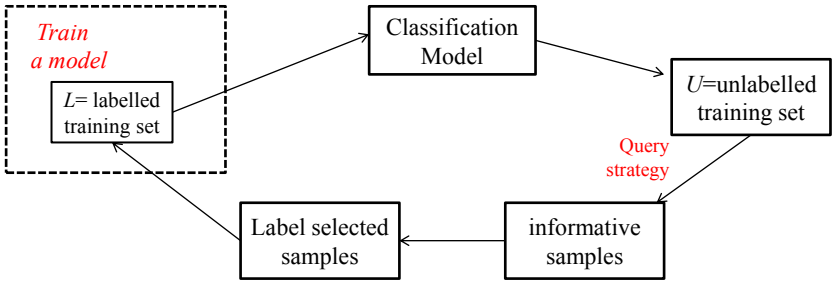


Figure 7.1: Pool Based Active Learning

- **Uncertainty [145]:** The active learner selects the instance that the learned model is least confident about which means most uncertain. This approach is based on posterior probabilities. For binary imbalanced data sets the posterior probability of being from the majority class and posterior probability of being from the minority class are used to calculate the uncertainty of each instances. Similarly, for binary balanced data sets the posterior probability of being from the positive and negative classes are used. One way to calculate uncertainty is utilising entropy (Eq. 7.1) where bigger entropy means more uncertainty. The entropy was found using:

$$E(x) = -\sum_{i=1}^{c} p_i(x) log_2 p_i(x) \tag{7.1}$$

where c is the number of classes which is 2 in our case, p refers to posterior probabilities and x means an unlabelled instance.

- **Maximum Probability [139]:** This strategy is specific for imbalanced data sets where the training data is balanced by selecting the instances that the model gives high posterior probability to be from the minority class.

- **Information Density [152]:** In this strategy, the informative instances are not only those which are uncertain (having bigger entropy) but also those which are more representative. Being representative is defined in terms of similarity. Different metrics can be used as the similarity measure. In our work, we used the inverse of the Euclidean distance. The information density of an instance

was calculated as:

$$sim(x_1,x_2) = \frac{1}{1+d(x_1,x_2)}$$
$$ID(x) = E(x)\big[\frac{1}{U}\sum_{i=1,x_i\neq x}^{U} sim(x,x_i)\big] \tag{7.2}$$

where U is the total size of the unlabelled instances, x and x_i refer to unlabelled training samples, d is the Euclidean distance and $E(x)$ is the entropy as defined in Eq. 7.1.

- **Random Selection:** Selecting any random unlabelled training sample to label without considering any criterion such as posterior probabilities, similarity, etc. This is not an *AL* query strategy but should be applied as it is the benchmark.

7.2 Active Learning with Feature Selection

Active learning aims to provide faster learning with a lower labelling cost. By selecting suitable training data for active learning, it is possible to increase classification performance on imbalanced data sets compared to passive learning [143, 146, 147, 180]. In this work, we are not presenting a novel query strategy to select informative instances. Instead, we apply the most common query strategies no matter if the set is imbalanced or balanced. On the other hand, feature selection is a well-studied subject which generally increases the classification performance by selecting the best features and decreases the size of the feature space.

Motivated by the successful performance of these two methodologies, we integrate feature selection with active learning and apply them to classification of balanced and imbalanced data sets. In the literature, there is a vast amount of research on active learning and feature selection individually. However, there is not much work which combines these two methodologies together. To the best of our knowledge, previous studies about active learning with feature selection all belong to the natural language processing field (especially text classification). A review on this topic is given in Section 2.5.

In this section, we claim that a proper feature selection criterion (especially for imbalanced data classification which forces the classifier to pay attention to the classification of the minority class or both classes equally) can give better classification

performance compared to active learning without feature selection and also passive learning. Motivating from this claim, we want to answer the following two questions:

1. How is the performance of active learning affected when it is integrated with feature selection?

2. What is the best active learning query strategy (including random selection) when it is integrated with feature selection?

To answer the first question, the performances of active learning (uncertainty [145], maximum probability [139], information density [152]) and random selection with/without feature selection are compared. What we conclude is that better classification performance can be obtained by applying active learning with feature selection. Additionally, the computational time of *AL* with/without feature selection and the number of selected features during *AL* with feature selection is investigated to determine the efficiency. To answer the second question the performance of 3 query strategies and random selection integrated with feature selection are compared with each other. What we conclude is there is no significantly better algorithm.

7.2.1 Methodology

As the feature selection method, the Sequential Forward Feature Selection Method (SFFS) as given in Section 4.1.4 was used. Feature selection is integrated with active learning as (Figure 7.2):

- For each labelled training set, apply feature selection starting with an empty set of features.

- Train a model based on the best feature subset of the training set at a specific iteration of active learning.

- Determine the most informative samples using the model and the query strategy.

- Label the selected samples and extend the training set with them.

- Repeat these steps until an active learning stopping criterion is achieved.

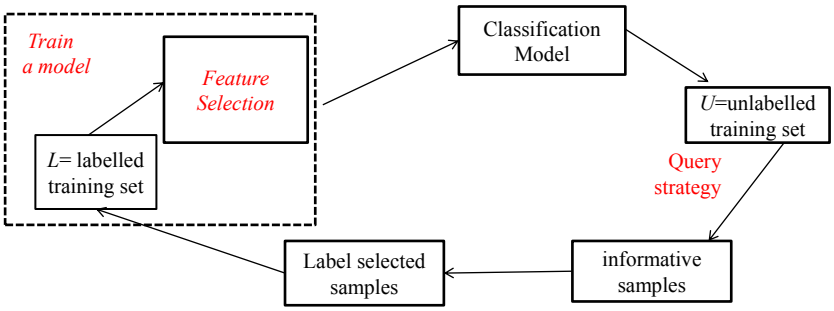


Figure 7.2: Active Learning with Feature Selection

7.2.2 Experimental Work

For the experiments presented in this section, we continued iterating active learning until all training data was labelled to investigate the methods completely. As classifiers we used Naive Bayes (NB) and a Support Vector Machine (SVM).

Naive Bayes (NB): By being simple, efficient and providing posterior probabilities which helps to determine the informative instances, NB is suitable for our purpose. Additionally, it has good performance in many imbalanced problems such as [181, 182] and the data sets used in this study. NB was applied using kernel density estimation with a normal kernel smoother. The prior probabilities were uniform for all classes.

Support Vector Machine (SVM): As the kernel function, a radial basis function was applied. Using the $MATLAB\ fitcsvm$ function [183] the scale values of the kernel functions were found automatically. The data was standardised (normalised) before applying SVM. $MATLAB\ fitPosterior$ function [183] allowed us to find the posterior probabilities which were used by the AL query strategies to find the informative samples.

7.2.2.1 Data Sets

Five imbalanced data sets and two balanced data sets from different application domains were used to evaluate the effectiveness of active learning with feature selection. The number of features ($\#Fea.$), the total number of samples ($\#Sam.$), the total number of minority ($\#Min.$) and majority samples ($\#Maj.$) (positive and negative classes for balanced data sets), the imbalance ratios (IR; the number of minority class samples

Table 7.1: Used Balanced and Imbalanced Data Sets

Data Sets	$\#Fea.$	$\#Sam.$	$(\#Min, \#Maj.)$	IR	$Ref.$
Pima	8	768	(268, 500)	~ 0.50	[4, 5]
Oil	49	937	(41, 896)	~ 0.05	[6]
Satimage	36	6435	(626, 5809)	~ 0.11	[7]
Forum Pedestrian Database	57	2342	(718, 1624)	~ 0.44	Section 5.2.1
Fish Trajectory	179	3102	(59, 3042)	~ 0.02	Section 5.2.1
Satimage_2	36	3041	(1508, 1533)	~ 0.99	[7]
Yeast	8	926	(463, 463)	1	[7]

(positive class samples for balanced data sets) over majority class samples (negative class samples for balanced data sets)) and the corresponding citations for each data sets ($Ref.$) are given in Table 7.1.

The Satimage data set was used as described in Section 6.1.1.1. To obtain the Satimage_2 data set, we used classes 1 and 7 of the original Satimage data set as given in [7]. As the other balanced data set, samples of classes: 3, 4, 5 and 10 of the Yeast [7] data set are collapsed and used as a single class while class 1 was chosen as the other class. The fish trajectory and the Forum pedestrian database were used as described in Section 5.2.1. The other data sets (Pima [4, 5] and Oil [6]) were supplied as binary classes by the given references.

7.2.2.2 Experimental Design

For the Pima [4, 5], Oil [6], Satimage [7], Satimage_2 [7] and Yeast [7] data sets, results were averaged across 5-fold cross validation. For the Forum pedestrian and the fish trajectory data sets, we applied 9-fold cross validation (to be consistent with the previous chapters). Training (3/5 or 7/9), validation (1/5 or 1/9) and testing (1/5 or 1/9) sets were formed randomly with the same class distributions. The validation set is used during feature selection to determine when to stop feature selection. For the experiments concerning only active learning, the same training and testing sets with active learning with feature selection are used. The minority (positive class for balanced data sets) and majority class (negative class for balanced data sets) samples were distributed equally to each set.

At each cross validation fold, **1 sample from the minority class and 1 sample**

Table 7.2: The number of samples selected at each iteration of active learning for different data sets and classifiers.

Classifiers	Pima	Oil	Satimage	Forum Pedestrian	Fish Trajectory	Satimage_2	Yeast
NB	5	5	25	5	25	25	5
SVM	5	5	25	25	25	25	5

from the majority class were randomly chosen as the initial labelled training set. The given query strategy was then used to pick samples from the remainder of the training data set. The number of chosen samples at each iteration of active learning and random selection is given in Table 7.2. As seen, for larger data sets more samples were chosen. We **did not apply any early stopping criterion** and active learning iterations continued until all training samples were labelled.

For all the experiments presented in this section the evaluation metric is *GeoMean* (Eq. 2.7). For the experiments when *NB* is used as the classifier, uncertainty [145], maximum probability [139], information density [152] and random selection were utilised. As uncertainty [145] and maximum probability [139] did not usually preformed as well as information density [152] and random selection (see Sections 7.2.3 and 7.3), the experiments with *SVM* was performed using only information density [152] and random selection.

7.2.3 Results when Naive Bayes is used as the classifier

Figures 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, and 7.9 show testing performances (as average of folds) at each iteration of active learning with/without feature selection for Pima [4, 5], Oil [6], Satimage [7], the Forum pedestrian database, the fish trajectory, Satimage_2 [7] and Yeast [7] data sets respectively. In these analysis, the evaluation metric is *GeoMean* (Eq. 2.7).

7.2.3.1 How is the performance of active learning affected when it is integrated with feature selection?

By looking at the performance comparisons given in Figures 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, and 7.9:

- For the Pima data set [4, 5], active learning with feature selection reached the

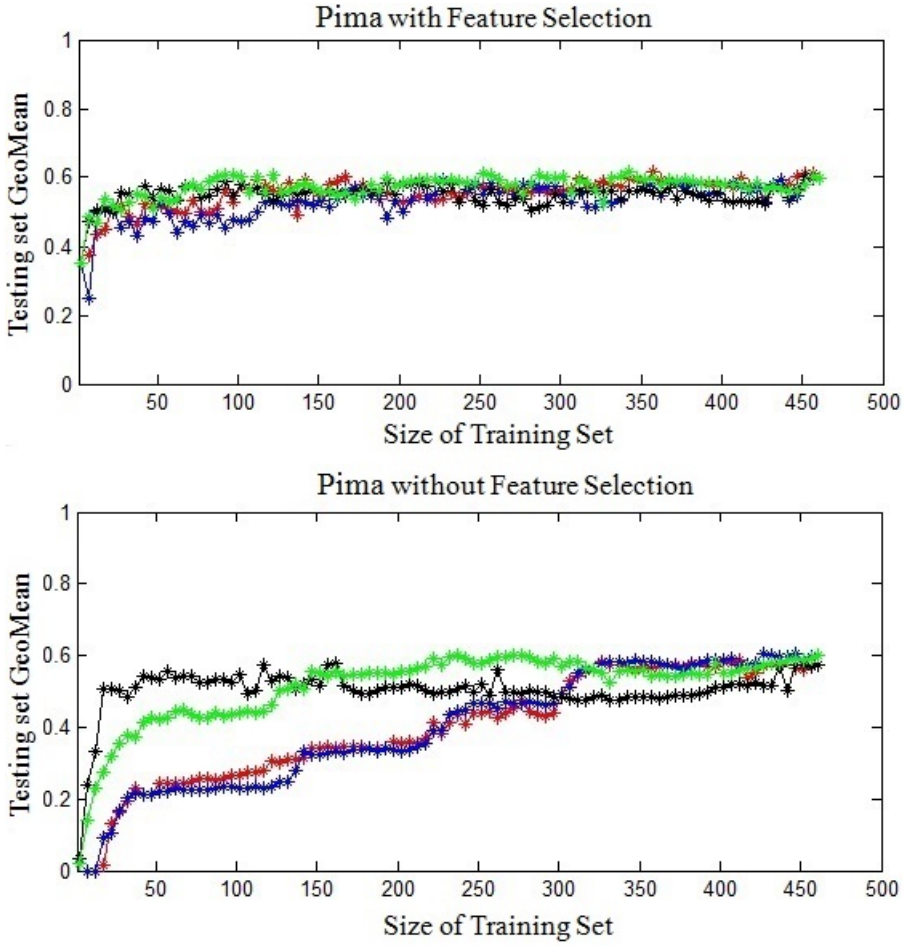


Figure 7.3: Results in terms of the $GeoMean$ for the Pima data set [4, 5] using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density.

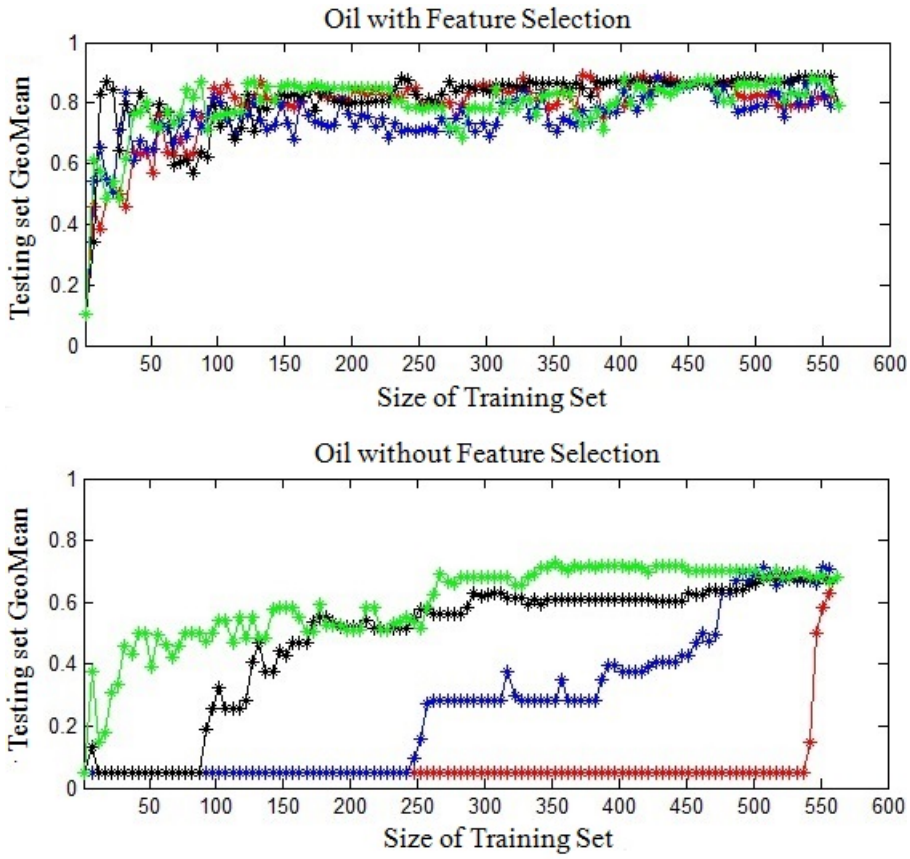


Figure 7.4: Results in terms of the $GeoMean$ for the Oil data set [6] using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density.

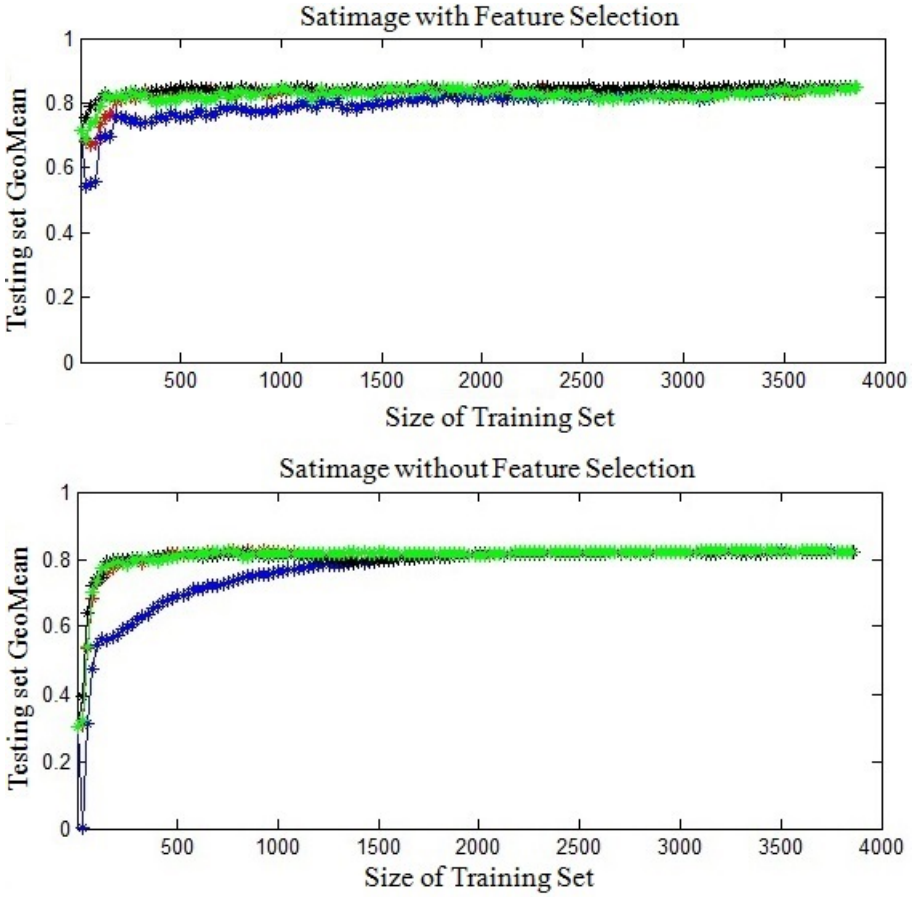


Figure 7.5: Results in terms of the $GeoMean$ for the Satimage data set [7] using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density.

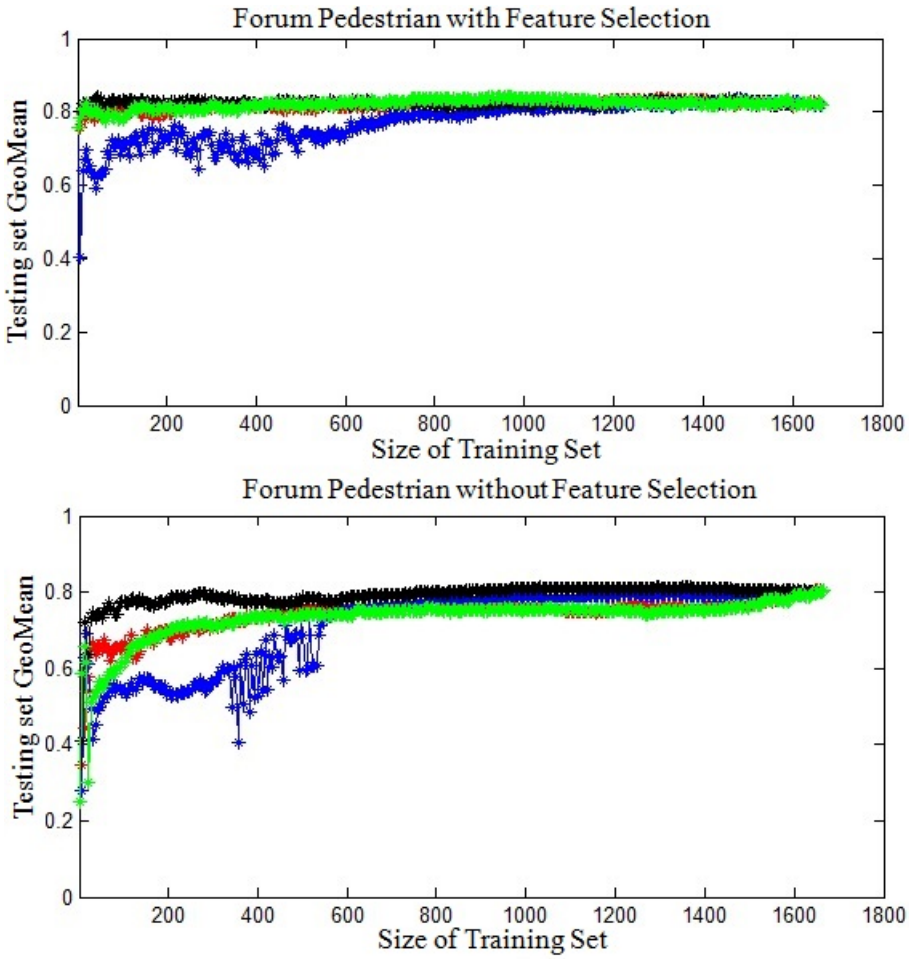


Figure 7.6: Results in terms of the $GeoMean$ for the Forum Pedestrian Database using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density.

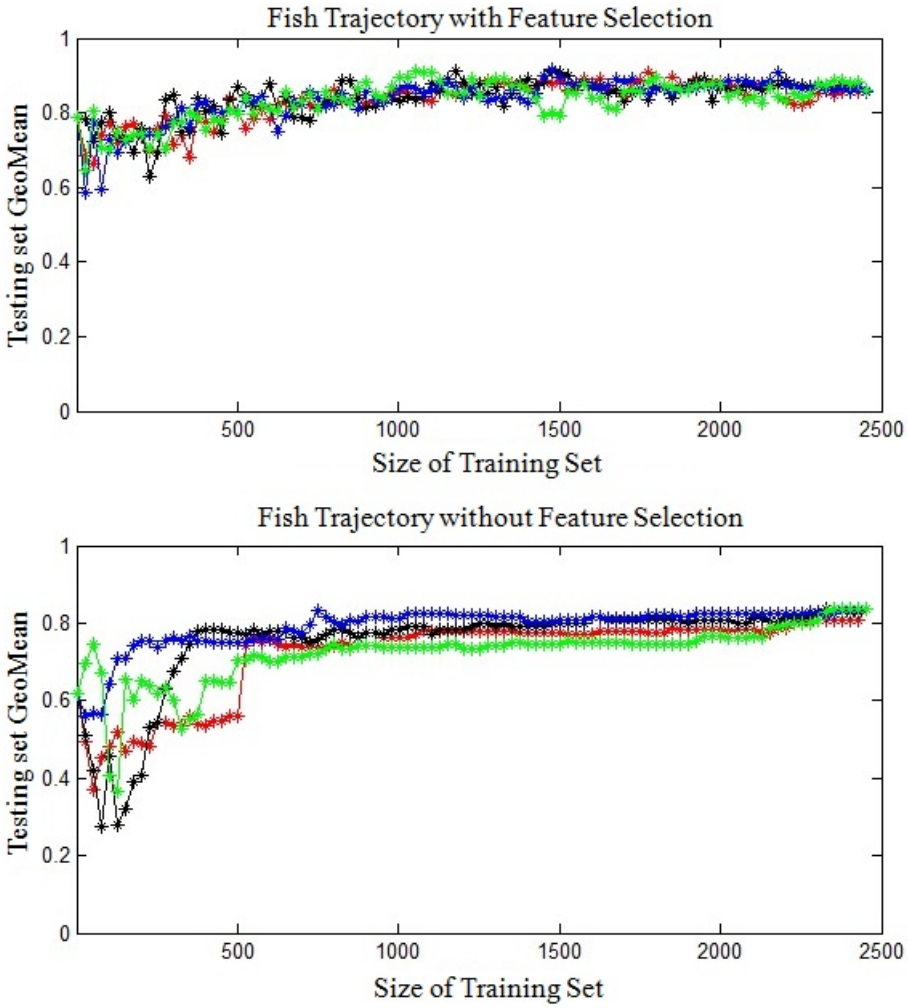


Figure 7.7: Results in terms of the $GeoMean$ for the fish trajectory data set using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, blue for maximum probability, black for random selection and green for information density.

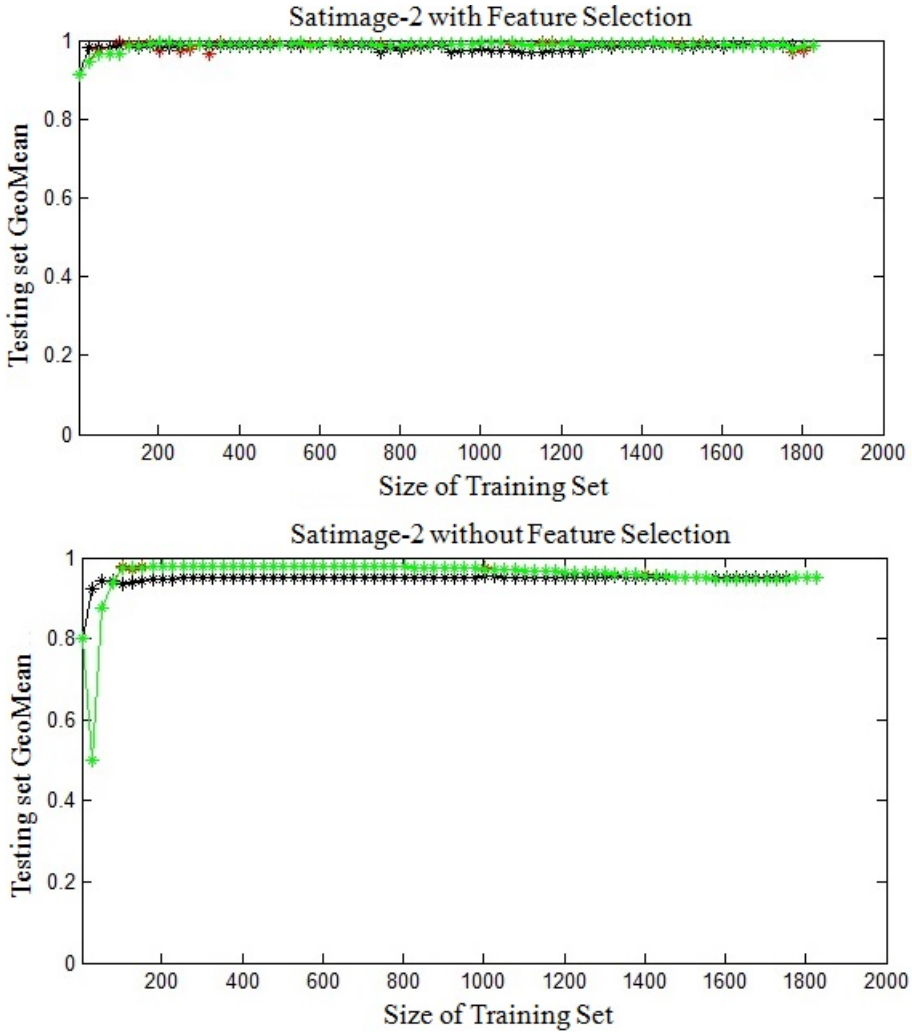


Figure 7.8: Results in terms of the $GeoMean$ for the Satimage_2 data set [7] using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, black for random selection and green for information density.

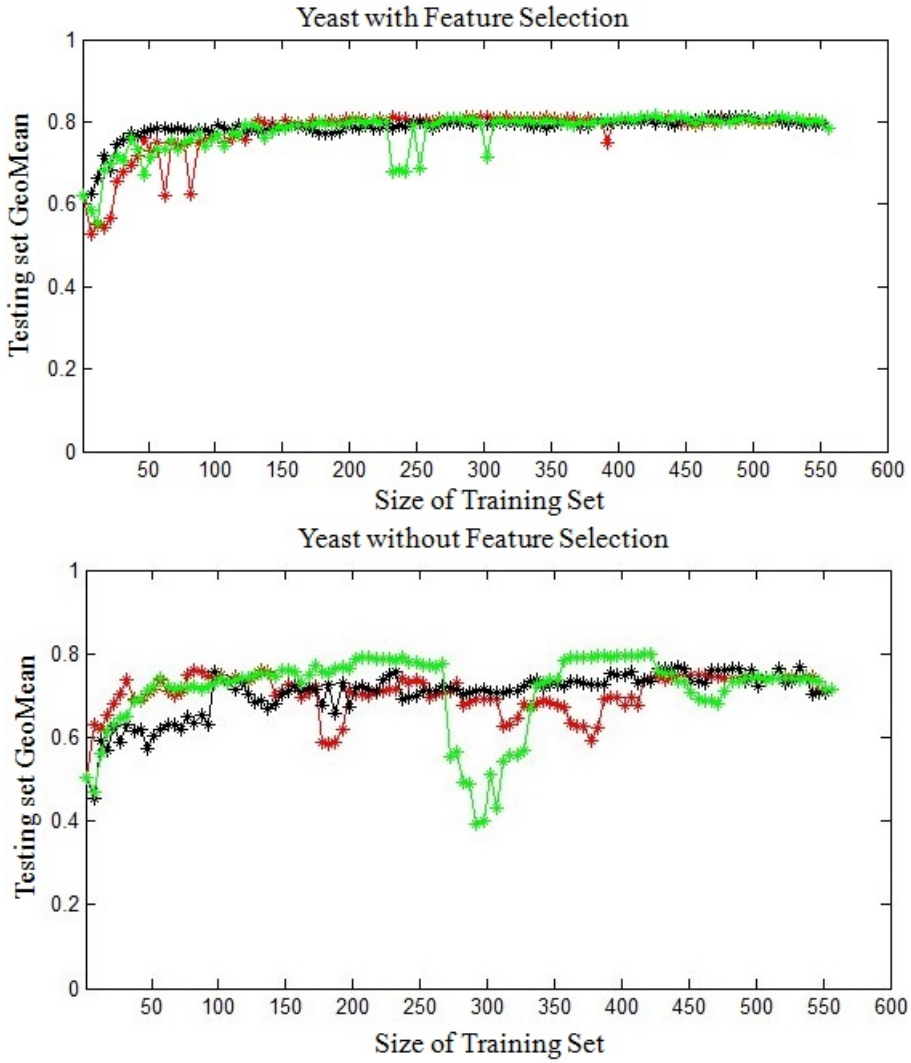


Figure 7.9: Results in terms of the $GeoMean$ for the Yeast data set [7] using NB. Active learning with feature selection (top) and without feature selection (bottom). Red for uncertainty, black for random selection and green for information density.

best performance quicker than only active learning and especially in early stages of active learning better performance was obtained with selected features compared to using all features.

- For the Oil data set [6], significantly better performance was obtained when active learning is combined with feature selection.
- Overall, 3% better performance was observed when feature selection is integrated with active learning compared to only active learning in the Satimage data set [7].
- Similarly, for the Forum Pedestrian data set, better performance was obtained even in very early stages. In general, in this data set performance with feature selection was 2-3% better than without feature selection.
- For the fish trajectory data set, the performance of active learning with feature selection was much better than without feature selection and the results showed that feature selection is required for this data set.
- For the Satimage_2 data set [7], it is easy to see that feature selection improved active learning performance as active learning with feature selection always performed better than only active learning.
- And lastly, for the Yeast data set [7], active learning with feature selection very frequently performed better than only active learning.

Over the 7 data sets, we can see that using feature selection with all of the query strategies reached the peak performance earlier, and in most cases, achieved better performance than without feature selection. However, it is also noticeable that the performance of random selection improved when feature selection was applied.

In summary, feature selection has a positive effect on the performance of active learning and random selection. To better interpret the results, we applied the **paired t-test**. Paired t-test results showed the performance differences of each query strategy (uncertainty, maximum probability and information density) and random selection when feature selection is applied and not applied with active learning and random selection (Table 7.3). The pairings are the performance after each iteration (uses different training data but the same testing data). The paired t-test provides a significance value

Table 7.3: Paired t-test of the AL query strategies and random selection with/without feature selection when NB is the classifier (see text for more detail)

Data set		M1: Uncert_wFS M2: Uncert_woutFS	M1: MaxProb_wFS M2: MaxProb_woutFS	M1: InfoDen_wFS M2: InfoDen_woutFS	M1: Random_wFS M2: Random_woutFS	Total # of iterations
	M1 $\gg$ M2	**16**	**8**	**4**	**2**	
Pima	no significance	M1:**70** M2:6	M1:**62** M2:22	M1:**72** M2:16	M1:**82** M2:8	93
	M2 $\gg$ M1	0	0	0	0	
	M1 $\gg$ M2	**107**	**90**	**75**	**85**	
Oil	no significance	M1:**6** M2:0	M1:**23** M2:0	M1:**37** M2:1	M1:**28** M2:0	113
	M2 $\gg$ M1	0	0	0	0	
	M1 $\gg$ M2	**13**	**39**	**19**	**2**	
Satimage	no significance	M1:**131** M2:5	M1:**80** M2:15	M1:**106** M2:13	M1:**154** M2:0	156
	M2 $\gg$ M1	7	22	18	0	
Forum	M1 $\gg$ M2	0	**44**	0	0	
Pedestrian	no significance	M1:**254** M2:80	M1:**271** M2:17	M1:**219** M2:115	M1:**223** M2:111	334
Database	M2 $\gg$ M1	0	2	0	0	
	M1 $\gg$ M2	**44**	**3**	**58**	**38**	
Fish Trajectory	no significance	M1:**55** M2:0	M1:**89** M2:7	M1:**40** M2:1	M1:**60** M2:1	99
	M2 $\gg$ M1	0	0	0	0	
	M1 $\gg$ M2	**39**		**49**	**47**	
Satimage_2	no significance	M1:**31** M2:4	N/A	M1:**24** M2:1	M1:**27** M2:0	74
	M2 $\gg$ M1	0		0	0	
	M1 $\gg$ M2	**14**		**13**	**10**	
Yeast	no significance	M1:**88** M2:10	N/A	M1:**91** M2:8	M1:**102** M2:0	112
	M2 $\gg$ M1	0		0	0	

which shows whether a method is significantly better than the other method. For all tests, the significance level (p-value) is taken as 0.05.

In Table 7.3, $Uncert$ means uncertainty, $MaxProb$ is used for maximum probability, $InfoDen$ means information density, $Random$ is used for random selection while $M1$ refers to method with feature selection (wFS) and $M2$ refers to method without feature selection ($woutFS$) for paired t-test. M1 $\gg$ M2 shows the number of active learning iterations where the performance of wFS is significantly better than the performance of $woutFS$ for a specific data set. Similarly, M2 $\gg$ M1 shows the number of active learning iterations that the performance of $woutFS$ is significantly better than the performance of wFS. "no significance" shows the number of iterations where the p-value between the compared methods is above 0.05. Obtaining significant differences is difficult since it requires a method which is always better than the other method in all folds. Hence, a method which is worse only in one fold results in no significance between the methods. For the "no significance" case we showed the number of active learning iterations that a method performed better than the other considering "the average performance" over cross validation folds. N/A means not applicable and is used for the maximum probability strategy when the data set is balanced.

As can be seen from the results, the performances of all AL query strategies and also random selection improved (either significantly or on average) when feature selection is combined with active learning. Moreover, for the Oil [6] and Satimage_2 [7] data sets, feature selection is very beneficial since for any query strategy and random selection as well, the performance with feature selection was significantly better than performance without feature selection in a majority of the iterations.

Given that feature selection has a positive effect on the performance of *AL* and random selection, the following issues are important to consider as well:

- What is the number of selected features for *AL* and random selection when feature selection is integrated? A small number of selected features is better as fewer features will need to be extracted from all future data samples when the trained model is in use (after active learning is stopped such as by early stopping).
- What is the computational time with/without feature selection? It is obvious that feature selection requires additional time but considering the performance gain, the time complexity might be negligible.

Table 7.4 gives the average and standard deviation (after the $\pm$ sign) of **the number of selected features** over all iterations of active learning with feature selection and passive learning, when averaged over all cross-validation folds. Additionally, the total number of different features selected in over all folds for each query strategy, which shows the variety of features usable during active learning (AL) is also given. In this table, $Uncert$ means uncertainty, $MaxProb$ is used for maximum probability, $InfoDen$ means information density and *Random* is used for random selection.

The obtained results show that not many features were selected by SFFS [162] compared to the total number of features. Hence, we see that active learning and random selection with feature selection requires fewer features to reach the performance discussed above. On the other hand, the high values of the variety of the features show that most of the features were used at different stages of training. In conclusion, in addition to providing better performance, feature selection is also useful as fewer features are needed in the testing stage where the active learning is stopped.

Active learning cycle with and without feature selection are compared in terms of their **elapsed times** using random selection. The results are given in Table 7.5 as

Table 7.4: Number and variety of selected features for AL and random selection with feature selection

Data set: Total # of features	**Average and standard deviation (after the $\pm$ sign) of the # of selected features over AL iterations and cross validation folds**	**Average and standard deviation (after the $\pm$ sign) of the # of selected features over cross validation folds in passive learning**	**Variety of selected features**
Pima: 8	Uncert: 2.37 ± 1.6 MaxProb: 2.71 ± 1.59 InfoDen: 2.71 ± 1.6 Random: 2.75 ± 1.61	7 ± 0	Uncert: 8 MaxProb: 8 InfoDen: 8 Random: 8
Oil: 49	Uncert: 7.02 ± 6 MaxProb: 6.9 ± 6.61 InfoDen: 5.05 ± 5.27 Random: 1.93 ± 0.35	1.6 ± 0.55	Uncert: 49 MaxProb: 49 InfoDen: 49 Random: 44
Satimage: 36	Uncert: 3.44 ± 0.51 MaxProb: 3.31 ± 0.70 InfoDen: 3.33 ± 0.58 Random: 3.70 ± 0.48	3 ± 0.72	Uncert: 35 MaxProb: 36 InfoDen: 33 Random: 30
Forum Pedestrian Database: 57	Uncert: 3.26 ± 0.34 MaxProb: 3.08 ± 0.38 InfoDen: 3.38 ± 0.34 Random: 2.86 ± 0.31	3.33 ± 1.41	Uncert: 31 MaxProb: 31 InfoDen: 31 Random: 30
Fish Trajectory: 179	Uncert: 3.05 ± 0.35 MaxProb: 3.10 ± 0.35 InfoDen: 3.22 ± 0.51 Random: 3.21 ± 0.45	3.11 ± 1.36	Uncert: 133 MaxProb: 134 InfoDen: 149 Random: 138
Satimage_2: 36	Uncert: 3.65 ± 0.44 InfoDen: 3.55 ± 0.45 Random: 3.48 ± 0.46	3.8 ± 0.97	Uncert: 29 InfoDen: 28 Random: 23
Yeast: 8	Uncert: 2.6 ± 0.44 InfoDen: 2.68 ± 0.45 Random: 2.87 ± 0.43	2.83 ± 0.83	Uncert: 8 InfoDen: 8 Random: 8

Table 7.5: The comparison in computation time (average and standard deviation (after the $\pm$ sign) over different cross validation folds) between AL with/without feature selection (using Random Selection)

Data set	**Time with SFFS (min.)**	**Time without SFFS (min.)**
Pima	1.14 ± 0.25	0.49 ± 0.07
Oil	11.62 ± 3.04	2.06 ± 0.12
Satimage	237.25 ± 1.48	19.54 ± 1.24
Forum Pedestrian Database	121.38 ± 35.10	12.24 ± 1.07
Fish Trajectory	349.57 ± 167.07	21.62 ± 1.20
Satimage_2	28.54 ± 6.14	2.70 ± 0.07
Yeast	1.91 ± 0.18	0.31 ± 0.01

average and standard deviation (after the $\pm$ sign) of the elapsed training time in minutes averaged over different folds.

The results showed that, in the worst case (fish trajectory data set; the biggest data set in terms of feature dimensionality) active learning with feature selection is 16 times slower than only active learning. In the best case (Pima [4, 5]) active learning with feature selection is 2 times slower than pure active learning. In conclusion, there is a computational cost for the improved performance when using feature selection. This cost could be reduced by implementing feature selection in parallel on a task farming architecture with the methodology given in [2].

7.2.3.2 What is the best active learning query strategy (including random selection) when it is integrated with feature selection?

When feature selection is combined with *AL* we investigated what the best query strategy (including random selection) for each data set was. To do that, we applied the **paired t-test** to the evaluation metric (*GeoMean*) for each *AL* query strategy and random selection paired with another query strategy and random selection at each iteration of *AL* (Table 7.6).

The paired t-test results in Table 7.6 compared all query strategies and random selection using given data sets. There is no single **significantly better** algorithm. Random selection is often better than uncertainty (such as Forum Pedestrian data set),

Table 7.6: Paired t-test of the query strategies and random selection when feature selection is integrated and NB is used as the classifier (see text for more detail)

Data set		**M1: Uncert M2: Random**	**M1: Uncert M2: MaxProb**	**M1: Uncert M2: InfoDen**	**M1: MaxProb M2: Random**	**M1: MaxProb M2: InfoDen**	**M1: InfoDen M2: Random**	**Total # of iterations**
Pima	M1 ≫ M2	3	**8**	0	0	0	**2**	93
	no significance	M1:**52** M2:30	M1:**58** M2:25	M1:28 M2:**58**	M1:36 M2:**46**	M1:10 M2:**66**	M1:**69** M2:19	
	M2 ≫ M1	**6**	0	**5**	**9**	**15**	1	
Oil	M1 ≫ M2	0	1	0	0	0	0	113
	no significance	M1:46 M2:**64**	M1:**83** M2:26	M1:52 M2:**58**	M1:17 M2:**87**	M1:27 M2:**78**	M1:42 M2:**69**	
	M2 ≫ M1	**1**	1	**1**	**7**	**6**	0	
Satimage	M1 ≫ M2	0	**73**	**8**	0	1	0	156
	no significance	M1:20 M2:**113**	M1:**52** M2:20	M1:**78** M2:61	M1:1 M2:**49**	M1:26 M2:**63**	M1:15 M2:**104**	
	M2 ≫ M1	**21**	0	1	**104**	**59**	**35**	
Forum	M1 ≫ M2	8	**195**	10	2	2	16	334
Pedestrian	no significance	M1:**151** M2:128	M1:**110** M2:26	M1:123 M2:**178**	M1:26 M2:**116**	M1:45 M2:**58**	M1:133 M2:**139**	
Database	M2 ≫ M1	**45**	1	**18**	**178**	**227**	**44**	
Fish	M1 ≫ M2	0	0	3	2	**1**	1	99
	no significance	M1:47 M2:**48**	M1:**57** M2:40	M1:**52** M2:38	M1:45 M2:**47**	M1:47 M2:**49**	M1:42 M2:**48**	
Trajectory	M2 ≫ M1	**2**	0	**4**	**3**	0	**6**	
Satimage_2	M1 ≫ M2	**5**		**2**			**6**	74
	no significance	M1:**54** M2:13	N/A	M1:35 M2:35	N/A	N/A	M1:**54** M2:12	
	M2 ≫ M1	0		0			0	
Yeast	M1 ≫ M2	**9**		3			**6**	112
	no significance	M1:**62** M2:37	N/A	M1:**64** M2:40	N/A	N/A	M1:**59** M2: 42	
	M2 ≫ M1	2		3			3	

maximum probability (such as Satimage [7], Forum Pedestrian) and information density (such as Satimage [7], Forum Pedestrian). Information density is often better than uncertainty and maximum probability (such as Satimage [7], Forum Pedestrian). Additionally, in the early stages of active learning with feature selection for data sets such as Forum Pedestrian and Yeast [7], random selection performed better (details can be seen in Figures 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, and 7.9).

Overall, we conclude that when feature selection is used, random selection is statistically as good as each of the AL query strategies especially for imbalanced data sets. The results suggest that feature selection improves classification performance of all AL query strategies however it also improves the performance of the random selection.

As random selection was effective with the feature selection setting, we investigated it in detail. The standard deviations (considering the folds in cross validation) of each with/without feature selection iteration are given as the mean plus error bars in Figures 7.10, 7.11, 7.12, 7.13, 7.14, 7.15, and 7.16 for the data sets Pima [4, 5], Oil [6], Satimage [7], Forum pedestrian database, fish trajectory, Satimage_2 [7] and Yeast [7] data sets respectively.

As seen from the results the standard deviation of random selection decreased as

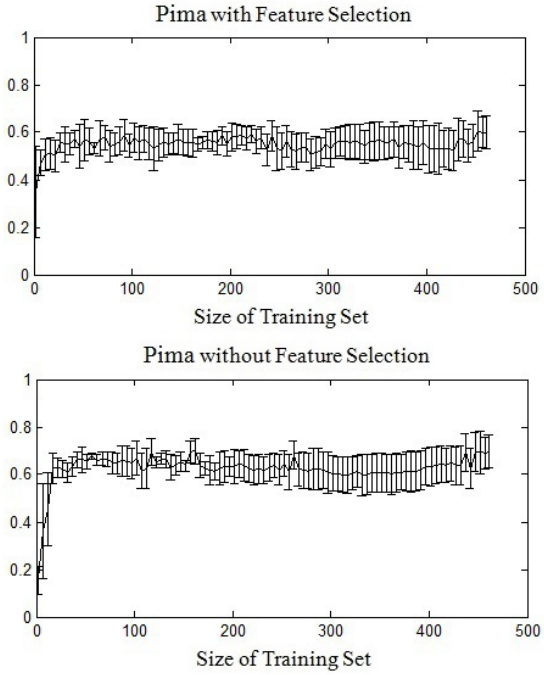


Figure 7.10: The mean plus error bars for the Pima [4, 5] data set using random selection. With feature selection (top) and without feature selection (bottom).

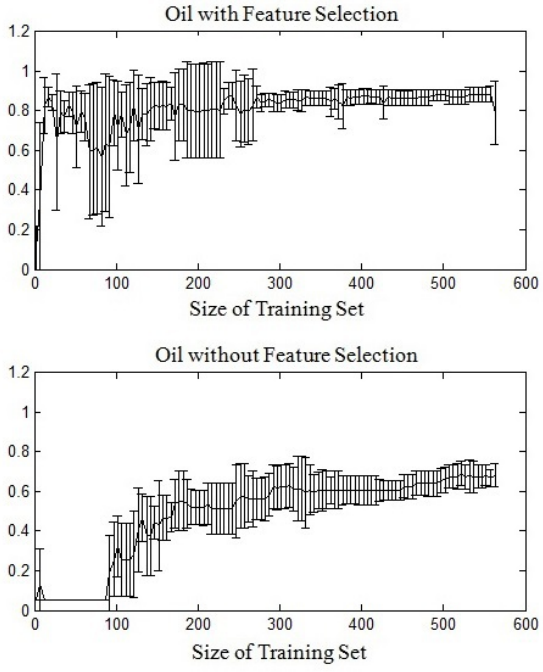


Figure 7.11: The mean plus error bars for the Oil [6] data set using random selection. With feature selection (top) and without feature selection (bottom).

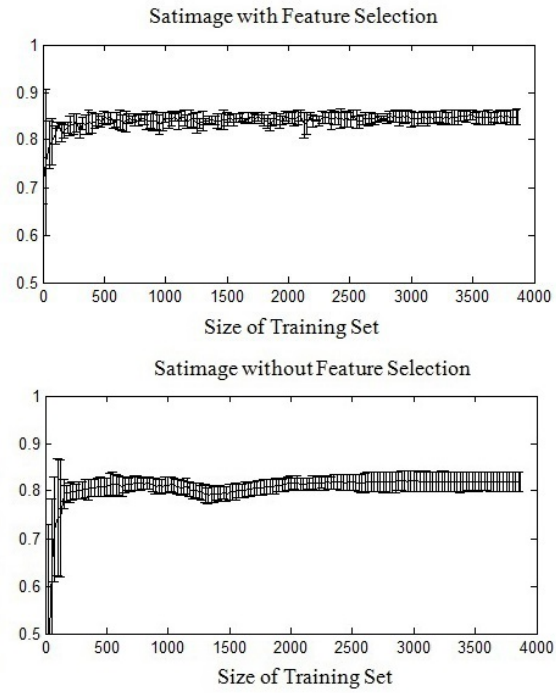


Figure 7.12: The mean plus error bars for the Satimage [7] data set using random selection. With feature selection (top) and without feature selection (bottom).

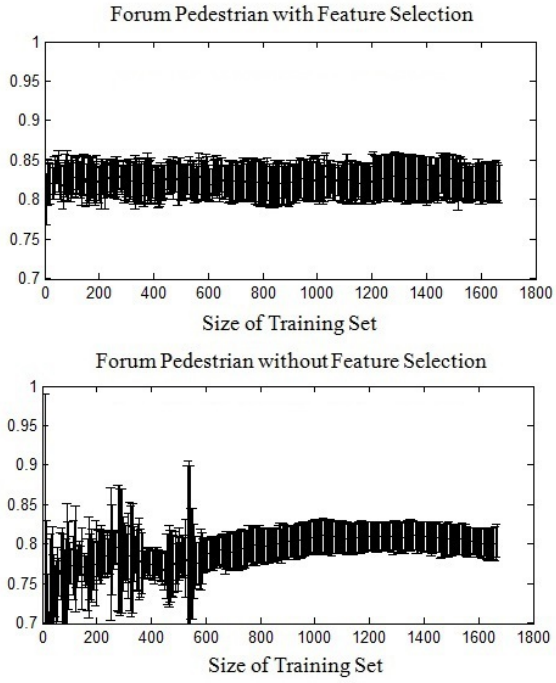


Figure 7.13: The mean plus error bars for the Forum Pedestrian Database data set using random selection. With feature selection (top) and without feature selection (bottom).

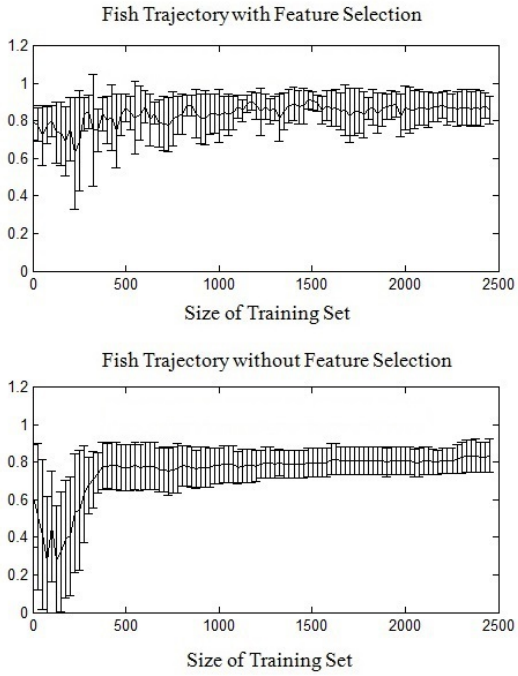


Figure 7.14: The mean plus error bars for the fish trajectory data set using random selection. With feature selection (top) and without feature selection (bottom).

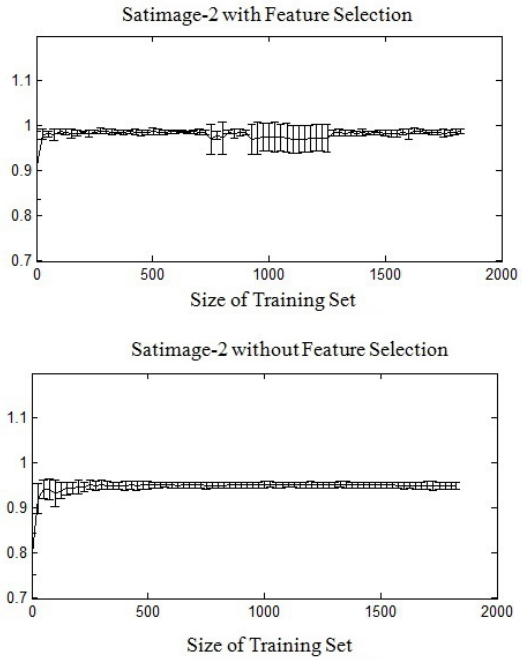


Figure 7.15: The mean plus error bars for the Satimage_2 [7] data set using random selection. With feature selection (top) and without feature selection (bottom).

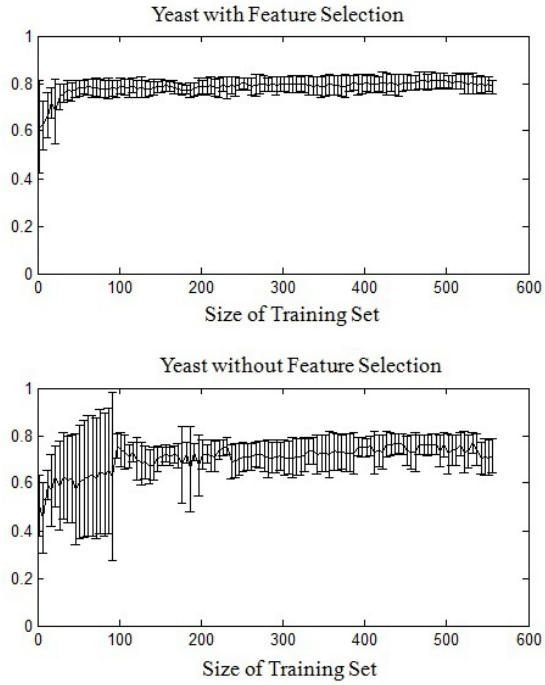


Figure 7.16: The mean plus error bars for the Yeast [7] data set using random selection. With feature selection (top) and without feature selection (bottom).

the size of the labelled training data was increased. This decrease was usually larger in without feature selection compared to with feature selection where standard deviations became almost stable in the earlier stages of active learning. It also means that random selection with feature selection not only performs well on average but also performs well in each fold of the cross validation.

7.2.4 Results when a Support Vector Machine is used as the classifier

The experiments using the *NB* classifier showed that random selection performs as good as *AL* query strategies when feature selection is integrated. However, the performance improvement that feature selection provided to the *AL* query strategies is also important. It may be the case that the benefits are seen only when a simple classifier like *NB* is used. Therefore, we further investigate active learning with feature selection using another classifier, *SVM*, to see if it is possible to obtain similar results or not.

SVM has many advantages such as *i)* being effective in high dimensional spaces and *ii)* being effective when the number of features is greater than the number of samples. Due to those advantages for *SVM* integration with feature selection might not

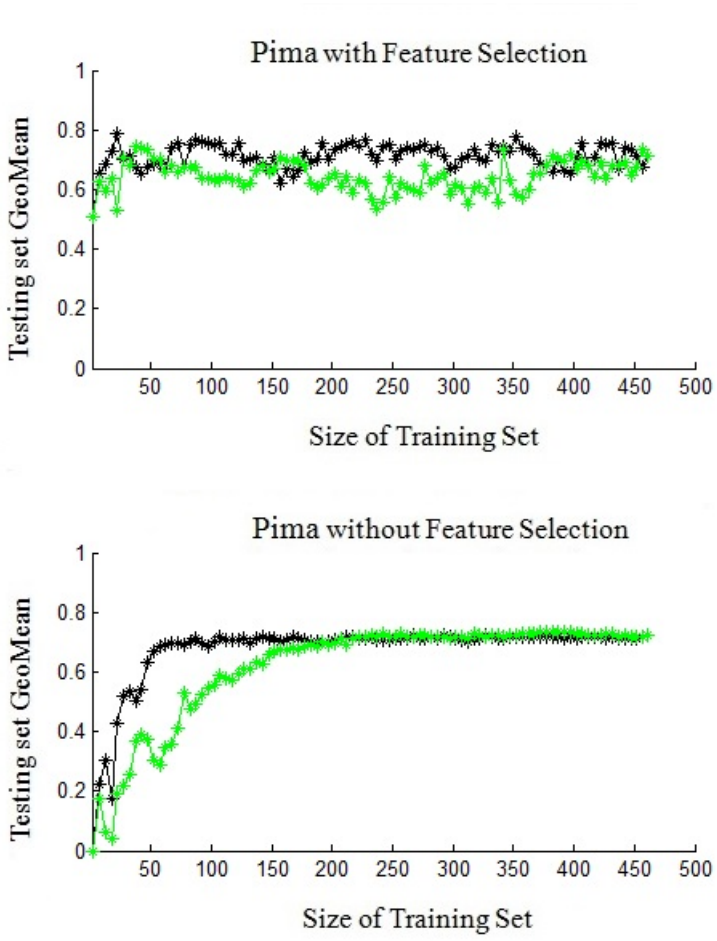


Figure 7.17: Results in terms of the $GeoMean$ for the Pima data set [4, 5] using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

be necessary or not even useful (which can also seen in Table 6.4 where SVM_wFS performed worse than SVM).

Figures 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, and 7.23 show the testing performances (as the average of cross validation folds) at each iteration of active learning with/without feature selection for Pima [4, 5], Oil [6], Satimage [7], the Forum pedestrian, the fish trajectory, Satimage_2 [7] and Yeast [7] data sets respectively. In these analysis, the evaluation metric is $GeoMean$ (Eq. 2.7) similar to the previous analysis.

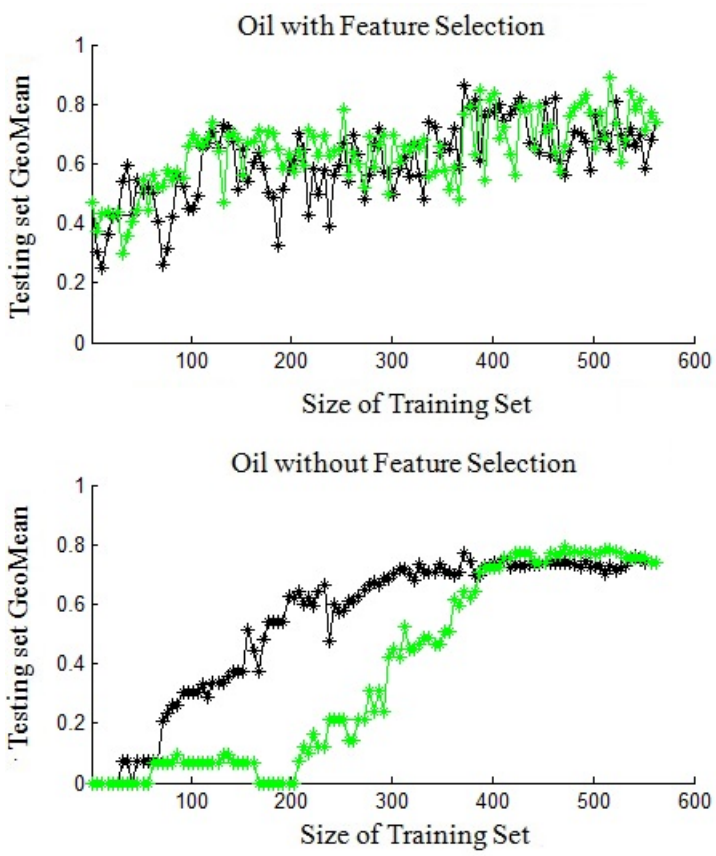


Figure 7.18: Results in terms of the $GeoMean$ for the Oil data set [6] using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

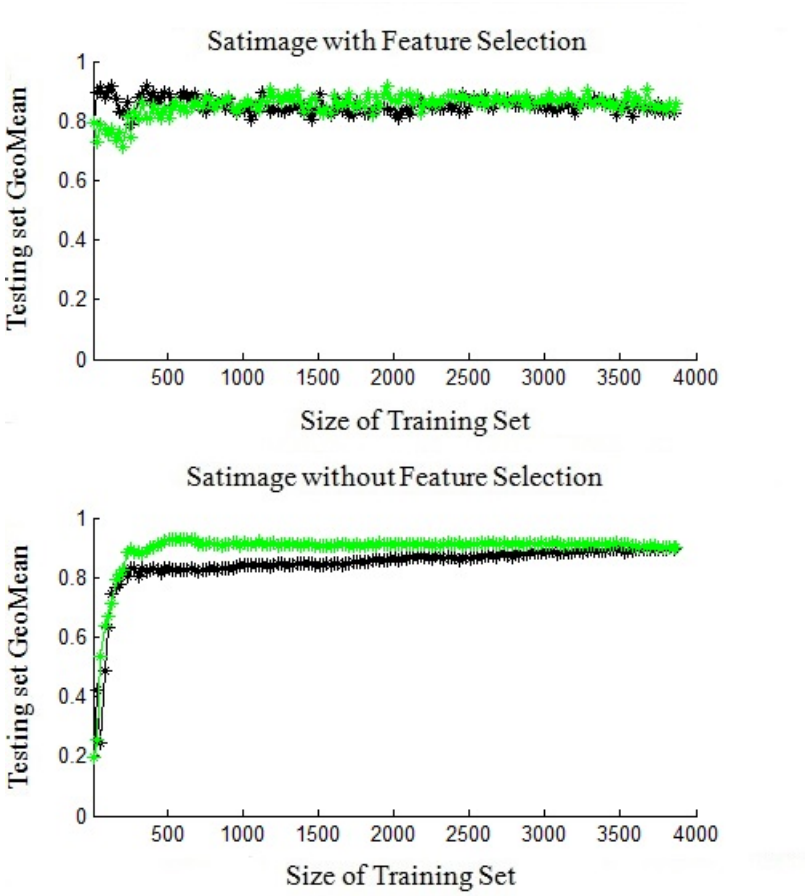


Figure 7.19: Results in terms of the $GeoMean$ for the Satimage data set [7] using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

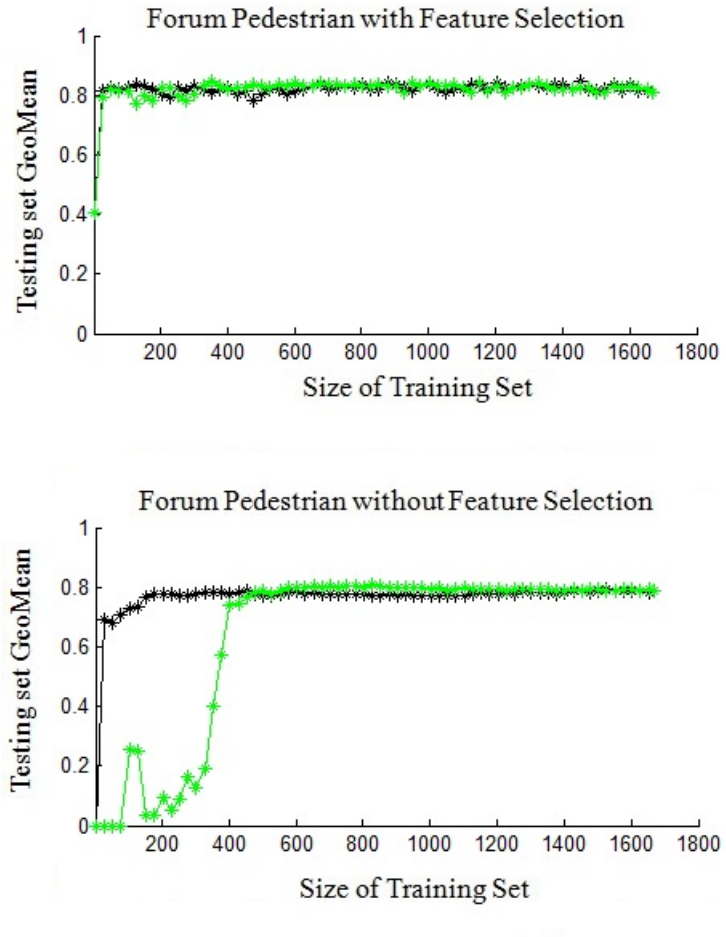


Figure 7.20: Results in terms of the $GeoMean$ for the Forum Pedestrian Database using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

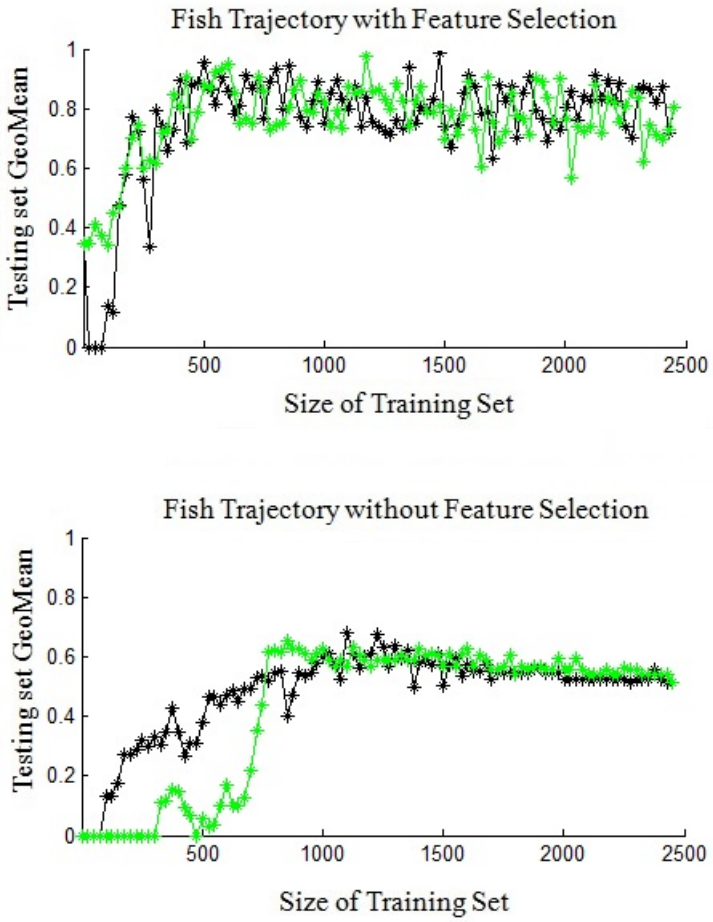


Figure 7.21: Results in terms of the $GeoMean$ for the fish trajectory data set using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

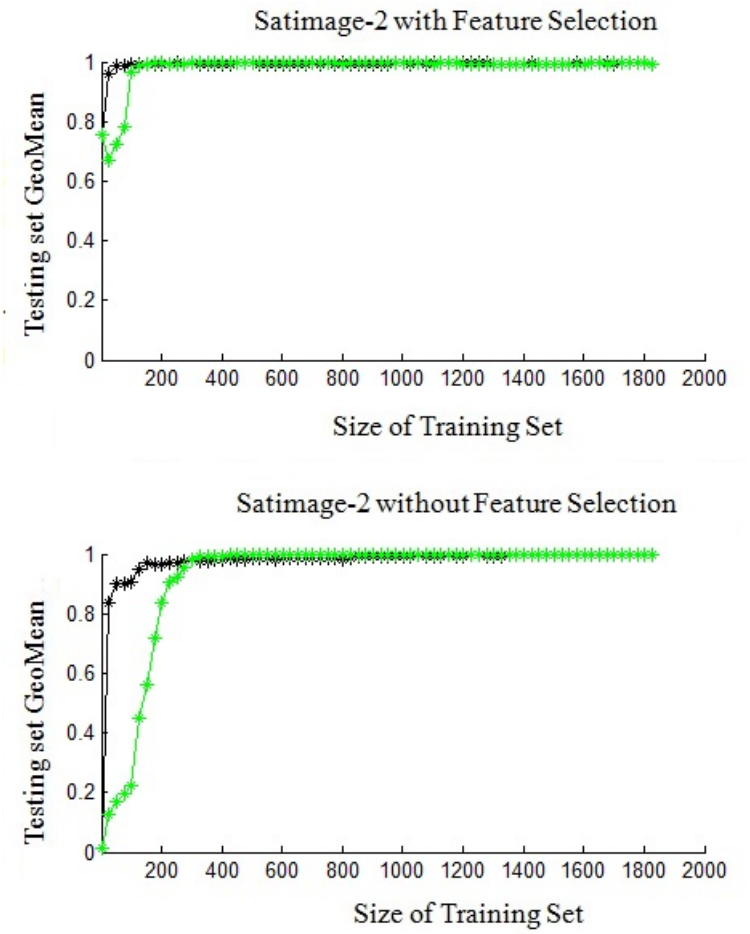


Figure 7.22: Results in terms of the $GeoMean$ for the Satimage_2 data set [7] using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

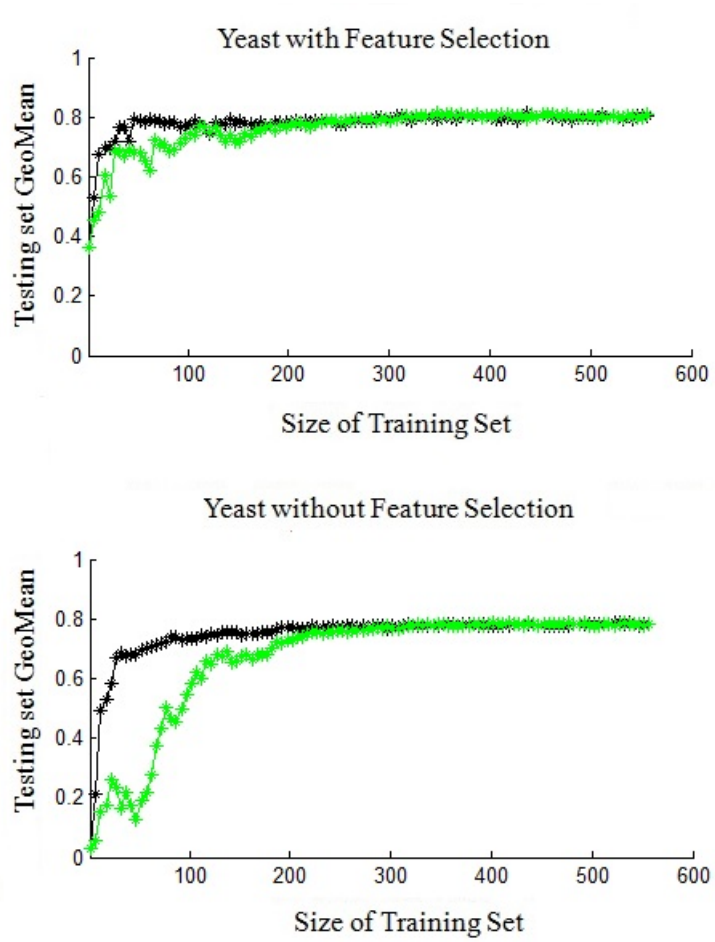


Figure 7.23: Results in terms of the $GeoMean$ for the Yeast data set [7] using SVM. Active learning with feature selection (top) and without feature selection (bottom). Black for random selection and green for information density.

7.2.4.1 How is the performance of active learning (including random selection) affected when it is integrated with feature selection?

The performance comparisons given in Figures 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, and 7.23 shows that:

- For the Pima data set [4, 5], in the early stages (such as until 100 samples exit in the training data) information density with feature selection performed better than without feature selection. However, in the latter stages, information density with feature selection did not perform (even though it is not significantly) as well as without feature selection. On the other hand, the performance of random selection with feature selection generally performed better than random selection only.

- For the Oil data set [6], information density with feature selection performed much better than without feature selection. The performance of random selection with feature selection was also better than without feature selection. But the improvement obtained when feature selection was applied with random selection was not as much as the improvement in information density with feature selection.

- For the Satimage data set [7], the performance of information density with feature selection was better than without feature selection only in the very early stages (until $\approx$ 100 samples in the training data) of *AL*. However, its performance in early stages was not really as good as its performance in the latter stages. Random selection with feature selection, on the other side, performed better than without feature selection until $\approx$ 700 samples in the training set and showed the best performance in those stages as well. However, in the latter stages performance of random selection with feature selection decreased and it did not perform better than random selection only.

- For the Forum Pedestrian data set, integrating feature selection with *AL* and random selection always performed better than *AL* and random selection without feature selection. Especially, information density with feature selection was quicker to reach the best performance compared to information density only. The results showed that feature selection is essential for this data set, particulary in the early stages of learning.

- For the fish trajectory data set, the performance of active learning and random selection with feature selection was much better than without feature selection and the results showed that feature selection is required for this data set.
- For the Satimage_2 data set [7], active learning and random selection with feature selection reached the best performance quicker than without feature selection.
- And lastly, for the Yeast data set [7], the performance of active learning and random selection with feature selection was much better than without feature selection and the results showed that feature selection is required for this data set.

In summary, it can be seen that using feature selection with information density generally achieved better performance than without feature selection (the performance of without feature selection was overall better in Satimage data set [7] and for the later stages of AL in the Pima data set [4, 5]). Moreover, similar to the results with the NB classifier, the performance of random selection improved as well when feature selection was applied (except Satimage data set [7]). However, different than the results with NB, the improvement in information density performance is greater than random selection when feature selection is integrated.

The **paired t-test** (p-value=0.05) was applied to determine the performance significance (if exists) and the results are given in Table 7.7. The pairings are the performance after each iteration (uses different training data but the same testing data).

In Table 7.7, $InfoDen$ means information density, *Random* is used for random selection while $M1$ refers to method with feature selection (wFS) and $M2$ refers to method without feature selection ($woutFS$) for paired t-test. M1 $\gg$ M2 shows the number of active learning iterations where the performance of wFS is significantly better than the performance of $woutFS$ for a specific data set. Similarly, M2 $\gg$ M1 shows the number of active learning iterations that the performance of $woutFS$ is significantly better than the performance of wFS. "no significance" shows the number of iterations where the p-value between the compared methods is above 0.05. For the "no significance" case we showed the number of iterations that a method performed better than the other considering "the average performance" over cross validation folds.

The results of the paired t-test show that the performance of information density generally improves (either significantly or on average) when feature selection is com-

Table 7.7: Paired t-test of the information density and random selection with/without feature selection when SVM is used as the classifier

Data set		**M1: InfoDen_wFS** **M2: InfoDen_woutFS**	**M1: Random_wFS** **M2: Random_woutFS**	**Total #** **of iterations**
Pima	M1 $\gg$ M2	16	**8**	93
	no significance	M1:21 M2:**39**	M1:**52** M2:29	
	M2 $\gg$ M1	**17**	4	
Oil	M1 $\gg$ M2	**45**	**15**	113
	no significance	M1:**45** M2:22	M1:45 M2:**53**	
	M2 $\gg$ M1	1	0	
Satimage	M1 $\gg$ M2	3	10	156
	no significance	M1:5 M2:**99**	M1:45 M2:**85**	
	M2 $\gg$ M1	**49**	**16**	
Forum Pedestrian Database	M1 $\gg$ M2	**15**	**10**	68
	no significance	M1:**53** M2:0	M1:**58** M2:0	
	M2 $\gg$ M1	0	0	
Fish Trajectory	M1 $\gg$ M2	**52**	**49**	99
	no significance	M1:**46** M2:1	M1:**46** M2:1	
	M2 $\gg$ M1	0	0	
Satimage_2	M1 $\gg$ M2	**13**	**16**	74
	no significance	M1:11 M2:**50**	M1:**37** M2:18	
	M2 $\gg$ M1	0	3	
Yeast	M1 $\gg$ M2	**22**	**15**	112
	no significance	M1:**90** M2:0	M1:**96** M2:1	
	M2 $\gg$ M1	0	0	

bined with active learning (except data sets Pima [4, 5], Satimage [7] and Satimage_2 [7]). However, random selection integrated with feature selection performs better than random selection only as well (except Satimage data set [7]). When the improvements of information density is compared with random selection, it is observed that the improvement of information density is more significant (such as for Oil data set [6], information density with feature selection performed significantly better than information density in 45 iterations while random selection with feature selection performed significantly better than random selection in 15 iterations).

What is the best active learning query strategy (including random selection) when it is integrated with feature selection?

The performance of information density and random selection is compared when feature selection is integrated. The **paired t-test** was applied to the evaluation metric (*GeoMean*) for each information density and random selection pair in the 7 data sets (Table 7.8).

The result given in Table 7.8 shows that there is **no significantly better** method between information density and random selection. Only one exception can be seen with the Pima data set [4, 5], where random selection performed significantly better than information density in 18 iterations of 93 iterations. On the other hand, *information density performed better than random selection in 5 of 7 data sets* (the data sets except Pima [4, 5] and Yeast [7]). However, *in the very early stages (such as when 10% of the training data is used) of active learning with feature selection* for the data sets Pima [4, 5], Satimage [7], Satimage_2 [7] and Yeast [7] (4 of 7 data sets) *random selection performed better on average.*

7.2.5 Summary and Discussion for Active Learning with Feature Selection

In this section, we investigated the effect of feature selection on active learning and we applied this methodology to classification of balanced and imbalanced data sets. Even though active learning and feature selection have been examined many times individually, and are both effective for balanced and imbalanced data set classification, fusion of these two has only been investigated in the natural language processing field where the feature space is implicitly changing since features are based on word frequency. In those studies, the best features at each stage of the active learning are determined

Table 7.8: Paired t-test of information density and random selection when feature selection is integrated and SVM is used as the classifier

Data set		**M1: InfoDen** **M2: Random**	**Total #** **of iterations**
Pima	M1 $\gg$ M2 no significance M2 $\gg$ M1	0 M1:16 M2:**57** **18**	93
Oil	M1 $\gg$ M2 no significance M2 $\gg$ M1	**3** M1:**67** M2:40 1	113
Satimage	M1 $\gg$ M2 no significance M2 $\gg$ M1	**4** M1:**94** M2:54 2	156
Forum Pedestrian Database	M1 $\gg$ M2 no significance M2 $\gg$ M1	**2** M1:**36** M2:28 0	68
Fish Trajectory	M1 $\gg$ M2 no significance M2 $\gg$ M1	1 M1:**51** M2:44 1	99
Satimage_2	M1 $\gg$ M2 no significance M2 $\gg$ M1	**6** M1:**40** M2:25 1	74
Yeast	M1 $\gg$ M2 no significance M2 $\gg$ M1	2 M1:46 M2:**60** 2	112

by a human expert and there is no comparison between using all features or only the selected features during active learning (see Section 2.5 for review).

The experiments with *NB* classifier using five imbalanced and two balanced data sets showed that by applying active learning with feature selection better classification performances can be obtained both for query strategies and random selection particularly in the early stages of active learning which results in less labelling. However, random selection is as effective as *AL* query strategies when they are combined with feature selection especially for imbalanced data sets. In this context, with feature selection, random selection performed better than information density and uncertainty in 4 of 7 data sets and performed better than maximum probability in 5 of 5 data sets while there is no single significantly better method.

The experiments with the *SVM* classifier showed that integrating feature selection with information density and random selection generally improved the performance of them. This improvement is important given that *SVM* is effective in high dimensional spaces as well. In contrast to the results obtained when the *NB* classifier was applied, the performance improvement from feature selection is generally greater for information density than random selection. The number of iterations showing the significant performance is also greater (5 of 7 data sets) for information density compared to random selection when methods with/without feature selection are compared (Table 7.7). However, there is no significantly better method when feature selection is integrated. In 5 of 7 data sets, information density with feature selection performed better than random selection with feature selection, although random selection with feature selection performed better in early stages of active learning (in 4 of 7 data sets).

When active learning strategies and random selection are compared in case of feature selection integration, there is no significantly better algorithm no matter which the classifier is. This is because obtaining significant differences is difficult as it requires a method which is always better than the other methods in all folds. It can be even more difficult for active learning since the training data is different (there is a high training set variability between random selection and other active learning query strategies) while methods are evaluated in the same testing data. On the other hand, there can be more important reasons (such as due to the characteristic of the data) which limits the performance of active learning against to random selection as discussed below.

Similar to the results presented here, several studies showed that it can be difficult for active learning query strategies to outperform random selection [146, 148, 157, 184, 185]. For instance, Bilgic [157] proposed a technique which is based on dimension-

ality reduction and determines the number of dimensions at each active learning step. The results of active learning with/without dimensionality reduction showed that active learning with dimensionality reduction performed significantly better than without dimensionality reduction. However, random selection with dimensionality reduction never performed worse than other query strategies with dimensionality reduction as well. Related to pure active learning (without feature selection or dimensionality reduction), in [184], the conditions that might affect the performance of active learning is discovered and the question of "when does active learning work" (compared to random selection) is tried to be answered. However, their analysis showed that for only 6% to 11% of the iterations active learning strategies are better than random selection even with different active learning query strategies, classifiers and different evaluation metrics. In a different study [148], the challenges that results in poor performance of active learning are listed. One of the challenges is having an **imbalanced class distribution** where active learning strategies find few samples from minority class. However, this challenge exist for random selection as well. The more important challenge is having **disjuncts** (sub-varieties of classes with a very small amount and overlapping) which are difficult to find by random selection but active learning strategies actually avoid finding them (such as information density which tries to select the most uncertain but also most similar sample to the other unlabelled samples) [148]. The data sets having **overlapping** also cause unreliable posterior probabilities especially at the early stages of the active learning which might lead the learning insufficiently. Given those challenges and existing studies presenting that the performance of random selection is as good as active learning query strategies, the obtained not significantly better but better in average performance of active learning query strategies (especially when SVM is applied) when AL is integrated with feature selection should considered as important. Moreover, the reason of not having significantly better performance by AL may be due to *i)* the disjuncts and *ii)* overlapping between different classes. To prove this, as future work, novel active learning query strategies that address these challenges should propose and the performances of them should be compared with random selection (with/without feature selection).

Another concern about active learning with feature selection can be determining a stopping criterion. As can be seen from results (also the plots given in [156]), active learning with feature selection results are not as smooth as results only with active learning. This might be because of the change in feature space at each step of active learning (even though it is not very common especially at the later stages of active

learning). However, the maximum performance rate is reached very quickly for all query strategies when feature selection is used especially when *NB* is used as the classifier. This suggests that training can be stopped rather quickly, but it is unclear how to define where to stop precisely. This can be investigated as future research as well.

7.3 Hierarchical Decomposition Method Integrated with Active Learning

In this section, the proposed hierarchical decomposition method is integrated with active learning. To do that, a novel setting which calculates class probabilities as they are needed to select informative samples by the active learning query strategies is proposed. The aim is to obtain good performance using less training data (i.e. the performance close to the performance that is obtained when all training data is used, see Section 7.3.3.1 for exact definition). In pure active learning applications (where the feature space is constant for all iterations of active learning) the performance generally increases as more training data is added unless there is noise in data labels. However, for the hierarchical decomposition method since it is possible that the hierarchy changes with different training samples and feature combinations at different levels, better classification performance can be obtained using less training data compared to using all training data.

Here, we investigate two issues:

1. Is it possible to obtain substantial performance with less training data compared to using all available training data when active learning (including random selection) is integrated with hierarchical decomposition using the proposed setting?

2. What is the best active learning query strategy (including random selection) when active learning is integrated with hierarchical decomposition using the proposed setting?

7.3.1 Proposed Setting

Active learning with hierarchical decomposition can be applied similarly to the description given in Section 7.2. The only different part is the calculation of scores of being from the majority and the minority classes which should be determined for each

unlabelled training sample to select the most informative samples. This difference is because the hierarchical decomposition method is not a probabilistic method.

The following steps are proposed to integrate hierarchical decomposition with active learning:

- Build the hierarchy as defined in Section 5.1.1 using the existing labelled data.
- Using the hierarchy and a query strategy, select informative instances from the unlabelled data. For each unlabelled sample, the probability of being from the majority class and the minority class are found. To do that, for each level of hierarchy including the misclassified clusters and the selected features for that hierarchy level, the closest cluster is found. If the closest cluster does not have enough samples to estimate the probabilities, then, it merges with the closest clusters until a set containing more than one majority and one minority class sample is obtained. Next, a Gaussian Mixture Model (*GMM*) is estimated with two components, one component for the majority class samples and the other component for the minority class samples. Here, we assume that the combined clusters are close to each other. The *GMM* is used to find the probability of being from the majority class and the minority class for each unlabelled data sample. The same steps are repeated at each level of the hierarchy. The final score of being from the majority class is found by the product rule [186] which is by multiplying the probabilities of being from the majority class at each level of the hierarchy (Eq. 7.3).

$$\begin{aligned} S_{final}(Maj|x) &= \prod_{hl=1}^{HL} P_{hl}(Maj|x) \\ S_{final}(Min|x) &= \prod_{hl=1}^{HL} P_{hl}(Min|x) \end{aligned} \tag{7.3}$$

where $P_{hl}(Maj|x)$ means the probability of being from the majority class at hierarchy level hl, $P_{hl}(Min|x)$ means the probability of being from the minority class at hierarchy level hl for the unlabelled training sample x where the total number of hierarchy levels is HL. $S_{final}(Maj|x)$ and $S_{final}(Min|x)$ represent the final scores of sample x which are used by the AL query strategies to determine the informative samples.

Similarly, the final score of being from the minority class is also found. By using the product rule, we assume the decisions of the different levels are independent

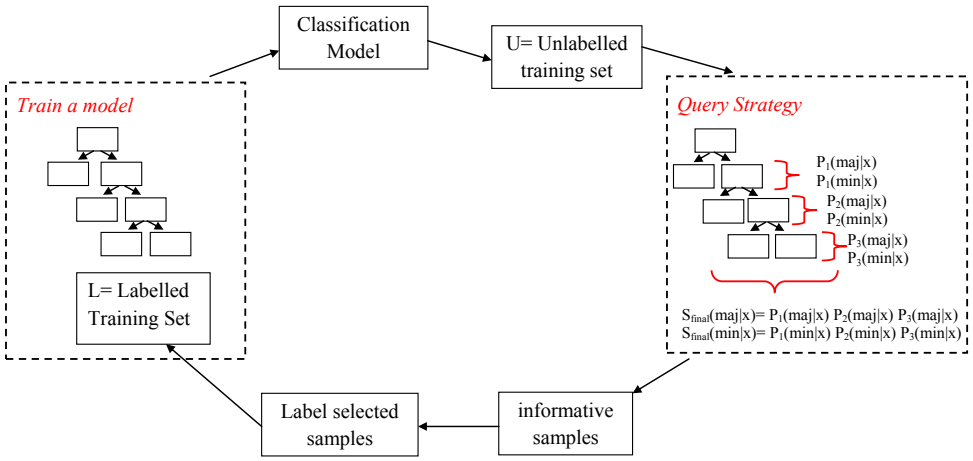


Figure 7.24: Hierarchical Decomposition Method Integrated with Active Learning

from each other. Once the final scores are found any query strategy presented in Section 7.2 is applied and the most informative samples are selected.

- Label the selected informative samples.
- Combine the new labelled instances with the previously labelled training set and repeat all these steps until an active learning stopping criterion is achieved.

The proposed setting is illustrated in Figure 7.24.

7.3.2 Data Sets and Experimental Design

Since the hierarchical decomposition method was proposed for imbalanced data set classification, in this section we only use the imbalanced data sets: Pima [4, 5], Oil [6], Satimage [7], the Forum pedestrian and the fish trajectory data sets (Section 5.2.1) for evaluation. The *AL* query strategies: uncertainty, maximum probability, information density given in Section 7.1 and random selection were compared.

At each cross validation fold (using the same cross validation scheme as given in Section 7.2.2.2), **9 samples from the minority class and 9 samples from the majority class** were randomly chosen as the initial labelled training set. The given query strategy was then used to pick samples from the remainder of the training data set. At each iteration of active learning, **5 samples** were chosen using the given query strategy for the Pima [4, 5], **25 samples** were chosen using the Oil [6] and **50 samples** were chosen using the Satimage [7], the Forum pedestrian and the fish trajectory data sets.

Since the training complexity of the hierarchical decomposition method is more than *NB* and SVM, the numbers of selected samples at each *AL* iteration are more than the numbers of selected samples at each *AL* iteration of active learning with feature selection. An early stopping criterion was not applied and the active learning iterations continued until all training samples were labelled.

As the outlier detection threshold $\{0.6, -0.3, -0.3, 1, 0.3\}$ were taken for the data sets Pima [4, 5], Oil [6], Satimage [7], the fish trajectory and the Forum pedestrian data sets respectively. During testing alternative-1 version (see Table 5.3 for details) of the hierarchical decomposition heuristic was applied to Pima [4, 5], Oil [6] and Satimage [7] data sets while the heuristic given in Figure 5.2 was applied to the Forum pedestrian and the fish trajectory data sets to provide the consistency with the results given in previous chapters.

7.3.3 Results

Figures 7.25, 7.26, 7.27, 7.28 and 7.29 show the testing performances (as the average of the folds) as a function of the number of trained data used at each iteration of active learning with hierarchical decomposition for Pima [4, 5], Oil [6], Satimage [7], the Forum pedestrian and the fish trajectory data sets respectively. In these analysis, the evaluation metric is $GeoMean$ (Eq. 2.7).

7.3.3.1 Is it possible to obtain substantial performance with less training data?

Substantial Performance: The consecutive performances (from any *AL* iteration to the end of all iterations) having $GeoMean \geq 95\%$ of the performance of passive learning (the last iteration of the *AL*) by any *AL* query strategy. The 95% performance threshold is shown with a horizontal cyan line and the starting iteration of the substantial performance is shown with vertical cyan line for each data set (see Figures 7.25, 7.26, 7.27, 7.28 and 7.29).

Best Performance: The highest performance ($GeoMean$) in any iteration by any *AL* query strategy (including random selection). This is the magenta line in the Figures 7.25, 7.26, 7.27, 7.28 and 7.29.

To answer the given question, it is enough to show that there is at least one *AL*

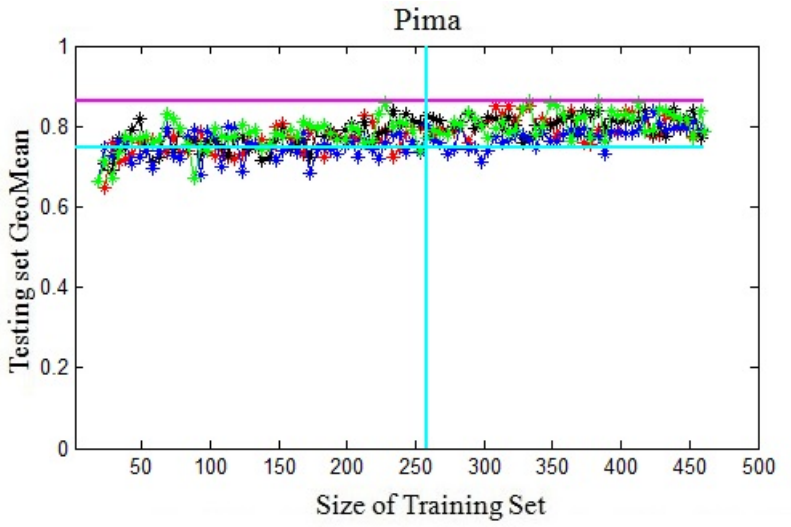


Figure 7.25: Active learning with hierarchical decomposition results in terms of the $GeoMean$ for the Pima data set [4, 5]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail).

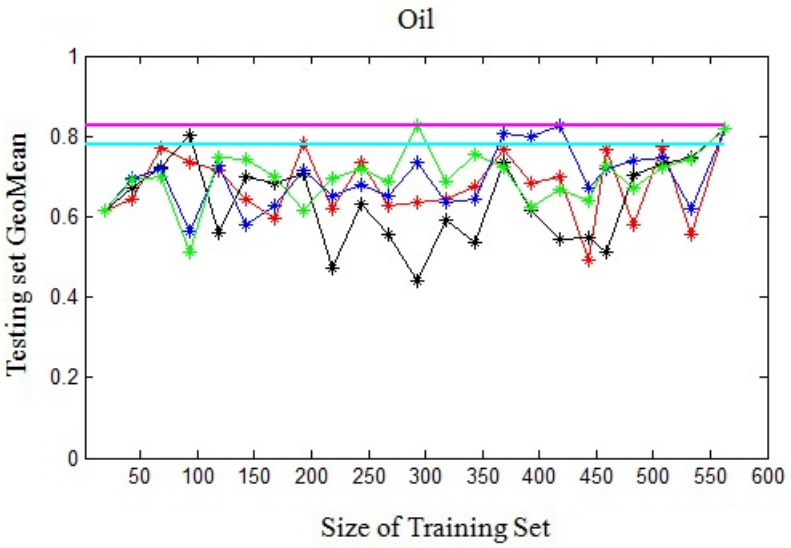


Figure 7.26: Active learning with hierarchical decomposition results in terms of the $GeoMean$ for the Oil data set [6]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan and magenta lines show the 95% of performance that was obtained when all training data is used and best performance, respectively (see text for more detail).

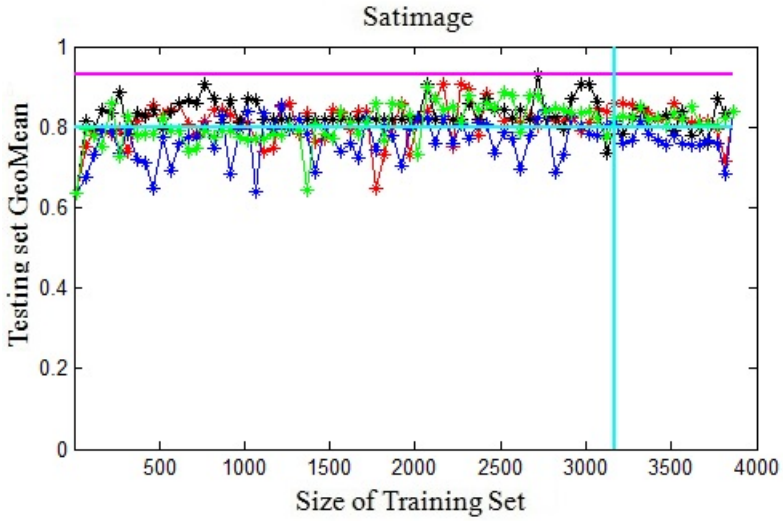


Figure 7.27: Active learning with hierarchical decomposition results in terms of the $GeoMean$ for the Satimage data set [7]. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail).

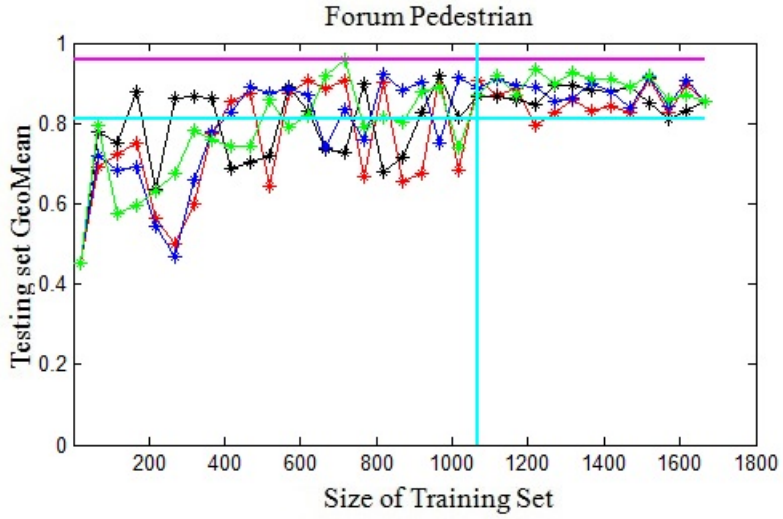


Figure 7.28: Active learning with hierarchical decomposition results in terms of the $GeoMean$ for the Forum pedestrian data set. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail).

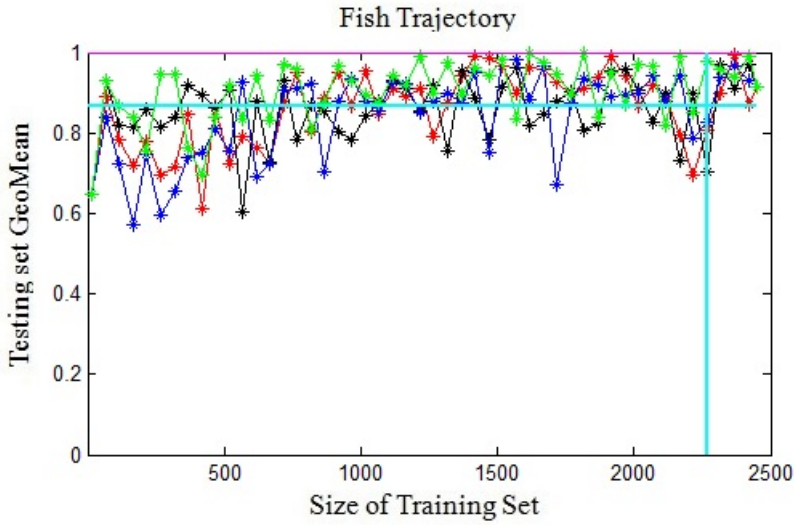


Figure 7.29: Active learning with hierarchical decomposition results in terms of the *GeoMean* for the fish trajectory data set. Red for uncertainty, blue for maximum probability, black for random selection and green for information density. The horizontal cyan, vertical cyan and magenta lines show the 95% of performance that was obtained when all training data is used, the size of training set where the substantial performance starts and best performance, respectively (see text for more detail).

strategy which satisfies the given substantial performance condition. Here, we used the results of information density to satisfy the substantial performance criterion. For all data sets, the best performances were reached before all training samples was used (5 of 5 data set). On the other hand, the substantial performances level was reached before the last *AL* iteration for all except the Oil data set [6] (4 of 5 data set). This shows that it is often possible to obtain a good performance with less data compared to using all training data. In detail:

- For the Pima data set [4, 5] with the given settings the best performance was observed when the model was trained with 383 samples (out of 461) which performed 0.86 *GeoMean* using information density as the query strategy. The *GeoMean* when all training data was used is 0.79. Substantial performance ($GeoMean \geq 0.75$) was achieved after training with 258 samples.

- For the Oil data set [6] with the given settings, the best performance was observed when the model was trained with 293 samples (out of 563) which performed 0.83 *GeoMean* using information density. The *GeoMean* when all training data was used is 0.82 and there is no substantial performance by any *AL*

Table 7.9: The minimum percentage of training data that is needed to obtain substantial performance and the percentage of training data that the best performance is reached with the corresponding AL strategy and random selection. $Random$ is for random selection and $InfoDen$ is for information density.

Performance	**Pima**	**Oil**	**Satimage**	**Forum Pedestrian**	**Fish Trajectory**
Best	83% ($InfoDen$)	52% ($InfoDen$)	70% ($Random$)	43% ($InfoDen$)	66% ($InfoDen$)
Substantial	56% ($InfoDen$)	–	82% ($InfoDen$)	64% ($InfoDen$)	92% ($InfoDen$)

strategy.

- For the Forum data set with the given settings, the best performance was observed when the model was trained with 718 samples (out of 1666) which performed 0.96 $GeoMean$ using information density. The $GeoMean$ when all training data was used is 0.86. Substantial performance ($GeoMean \geq 0.81$) was achieved after training with 1068 samples.

- For the fish trajectory data set with the given settings, the best performance was observed when the model was trained with 1618 samples (out of 2452) which performed 0.998 $GeoMean$ using information density. The $GeoMean$ when all training data was used is 0.91. Substantial performance ($GeoMean \geq 0.87$) was achieved after training with 2268 samples.

- Only for the Satimage data set [7], the best performance was observed using random selection when the model was trained with 2718 samples (out of 3862) which performed 0.93 $GeoMean$. The $GeoMean$ when all training data was used is 0.84. Substantial performance ($GeoMean \geq 0.80$) was achieved after training with 3168 samples.

To summarise, the best and substantial performances are given with the percentage of data used for the each data set in Table 7.9. This shows that generally it is possible to achieve reasonable performance using less training data than available in those data sets. Detecting when one has achieved substantial (or best) performance and can then stop training is a different question that is future work.

Table 7.10: Paired t-test results of the query strategies and random selection for hierarchical decomposition integrated with AL (see text for more detail)

Data set		M1: Uncert M2: Random	M1: Uncert M2: MaxProb	M1: Uncert M2: InfoDen	M1: MaxProb M2: Random	M1: MaxProb M2: InfoDen	M1: InfoDen M2: Random	Total # of iterations
	M1 $\gg$ M2	0	**4**	**3**	0	1	0	
Pima	no significance	M1:36 M2:**52**	M1:**56** M2:27	M1:31 M2:**52**	M1:24 M2:**58**	M1:16 M2:**66**	M1:**52** M2:24	90
	M2 $\gg$ M1	0	1	2	**6**	**5**	**2**	
	M1 $\gg$ M2	0	0	**1**	0	0	**2**	
Oil	no significance	M1:**13** M2:8	M1:9 M2:**12**	M1:8 M2:**12**	M1:**16** M2:4	M1:10 M2:**11**	M1:**12** M2:7	23
	M2 $\gg$ M1	0	0	0	**1**	0	0	
	M1 $\gg$ M2	1	**3**	0	0	0	0	
Satimage	no significance	M1:31 M2:**42**	M1:**55** M2:17	M1:**42** M2:32	M1:11 M2:**60**	M1:19 M2:**54**	M1:29 M2:**47**	78
	M2 $\gg$ M1	**2**	1	**2**	**5**	**3**	0	
Forum	M1 $\gg$ M2	2	1	0	2	**2**	**3**	
Pedestrian	no significance	M1:10 M2:**18**	M1:10 M2:**19**	M1:12 M2:**17**	M1:**17** M2:11	M1:13 M2:**16**	M1:**18** M2:11	34
Database	M2 $\gg$ M1	2	**2**	**3**	2	1	0	
Fish	M1 $\gg$ M2	**1**	1	1	**2**	0	2	
Trajectory	no significance	M1:23 M2:**24**	M1:**24** M2:22	M1:14 M2:**27**	M1:18 M2:**28**	M1:11 M2:**35**	M1:**32** M2:12	50
	M2 $\gg$ M1	0	1	**6**	0	**2**	2	

7.3.3.2 What is the best active learning query strategy (including random selection) when active learning is integrated with hierarchical decomposition?

To determine what the best AL query strategy including random selection for each data set was, **paired t-tests** were applied to the evaluation metric for each AL query strategy and random selection paired with another query strategy and random selection at each iteration of AL (Table 7.10). In Table 7.10, $Uncert$ means uncertainty, $MaxProb$ is used for maximum probability, $InfoDen$ means information density, $Random$ is used for random selection. M1 $\gg$ M2 shows the number of active learning iterations where the performance of M1 is significantly better than the performance of M2 for a specific data set. Similarly, M2 $\gg$ M1 shows the number of active learning iterations that the performance of M2 is significantly better than the performance of M1. "no significance" shows the number of iterations where the p-value between the compared methods is above 0.05.

The paired t-test results given in Table 7.10 show that over all query strategies, random selection and data sets, there is **no significantly better algorithm** (similar to the results obtained in Sections 7.2.3 and 7.2.4). For the Pima data set [4, 5] information density performed the best while random selection was the next best. Similarly, for the Oil data set [6] information density performed the best but this time maximum probability performed better than random selection and uncertainty. For the Satimage data set [7] random selection performed the best in majority of the times while uncertainty was the next best. For the Forum pedestrian data set information density performed the best while the next best method was maximum probability. For the fish trajectory data

set information density performed the best while random selection was the following best.

To conclude, information density performed the best in 4 of 5 data sets even though there is no significantly better AL query strategy including random selection. The second best in overall was random selection which performed better than information density in 29%, 30%, 60%, 32%, and 28% of the iterations for the data sets Pima [4, 5], Oil [6], Satimage [7], Forum pedestrian, and fish trajectory respectively.

7.3.4 Further Analysis

This section is to understand the algorithmic behaviour (not classification performance) of hierarchical decomposition method when it is integrated with active learning. In this context, the number of selected features and the number of hierarchy levels are investigated.

Table 7.11 gives the average and standard deviation (after the $\pm$ sign) of the number of selected features over different levels of the hierarchies for different active learning iterations and passive learning considering cross-validation folds.

The obtained results show that not many features were selected at each level by SFFS [162] compared to the total number of features. However, it was observed that the selected features at different levels of the hierarchy and different iterations of the active learning varied.

In Table 7.12, the average number of hierarchy levels over different active learning iterations and passive learning were given with the standard deviations (after the $\pm$ sign) considering cross validation folds.

As seen, for the Pima [4, 5], Oil [6] and Satimage [7] data sets the constructed hierarchies have less levels compared to the Forum pedestrian and the fish trajectory data sets who use more features compared to other three data sets. For the Pima [4, 5] and Oil [6] data sets the constructed hierarchies had 2 levels for the majority of the active learning iterations while at maximum 6 levels were obtained. For the Satimage data set [7] the numbers of hierarchy level were generally 1. By looking to the number of hierarchy levels it can be observed that the proposed hierarchical decomposition method is not suitable for classification of the Satimage data set [7] where the result given in Table 6.4 also supports this. For the Forum pedestrian data set, the depth of the hierarchy was 2 for the majority of the active learning iterations while at maximum 7 levels were obtained. For the fish trajectory data set, the depth of the hierarchy was

Table 7.11: Number of selected features for hierarchical decomposition integrated with AL and random selection

Data set: Total # of features	**Average and standard deviation (after the $\pm$ sign) of the # of selected features over different levels of hierarchy, AL iterations and cross vallation folds**	**Average and standard deviation (after the $\pm$ sign) of the # of selected features over different levels of hierarchy and cross validation folds in passive learning**
Pima: 8	Uncert: 3.10 ± 2.25 MaxProb: 3.00 ± 1.93 InfoDen: 3.64 ± 2.32 Random: 3.46 ± 2.38	3.6 ± 1.52
Oil: 49	Uncert: 5.26 ± 2.55 MaxProb: 5.08 ± 2.73 InfoDen: 4.21 ± 2.75 Random: 4.55 ± 2.68	8.75 ± 3.30
Satimage: 36	Uncert: 1.19 ± 0.47 MaxProb: 1.33 ± 0.65 InfoDen: 1.25 ± 0.54 Random: 1.20 ± 0.45	1.60 ± 0.55
Forum Pedestrian Database: 57	Uncert: 12.36 ± 7.57 MaxProb: 13.28 ± 8.13 InfoDen: 11.17 ± 6.55 Random: 13.23 ± 7.74	15.00 ± 8.47
Fish Trajectory: 179	Uncert: 13.05 ± 8.63 MaxProb: 19.37 ± 11.21 InfoDen: 14.03 ± 7.57 Random: 10.79 ± 7.53	15.56 ± 10.57

Table 7.12: Number of hierarchy levels for hierarchical decomposition integrated with *AL*

Data set	**Average and standard deviation (after the ± sign) of the # of hierarchy levels over AL iterations and cross validation folds**	**Average and standard deviation (after the ± sign) of the # of hierarchy levels over cross validation folds in passive learning**
Pima	Uncert: 1.51 ± 0.75 MaxProb: 1.55 ± 0.77 InfoDen: 1.70 ± 0.87 Random: 1.65 ± 0.87	1.60 ± 0.55
Oil	Uncert: 1.37 ± 0.71 MaxProb: 1.24 ± 0.60 InfoDen: 1.72 ± 0.82 Random: 1.28 ± 0.58	1.60 ± 0.55
Satimage	Uncert: 1.05 ± 0.30 MaxProb: 1.08 ± 0.30 InfoDen: 1.17 ± 0.43 Random: 1.51 ± 0.14	1.20 ± 0.45
Forum Pedestrian Database	Uncert: 2.50 ± 1.18 MaxProb: 2.60 ± 1.20 InfoDen: 2.37 ± 1.01 Random: 2.40 ± 1.02	3.00 ± 1.00
Fish Trajectory	Uncert: 4.66 ± 2.63 MaxProb: 4.28 ± 3.01 InfoDen: 5.49 ± 2.80 Random: 4.75 ± 2.72	6.11 ± 3.10

5 for the majority of the active learning iterations while at maximum 14 levels were obtained.

7.3.5 Summary and Discussion for Hierarchical Decomposition Method integrated with Active Learning

In this section, we investigated integrating the hierarchical decomposition method (Chapter 5) with active learning. This required estimating the scores of being from the majority and the minority classes which is needed to apply the active learning query strategies to select informative samples. The probabilities were found by a *GMM* belong to the closest cluster in a single hierarchy level to an unlabelled sample. The *GMM* of a cluster was found using the majority class samples and the minority class samples in that cluster as two different components. For each unlabelled data, at each hierarchy level the probabilities of being from majority class and the probabilities of being from minority class were found. Then, those probabilities for each class were combined with the product rule which provides the final scores. The results showed that with such a setting it is possible to obtain substantial performance using different active learning query strategies with less training data compared to using all available training data (passive learning). However, as future work other approaches such as using logistic regression to find the probabilities from unlabelled data and cluster distances (even though this needs labelled validation data as well) can be integrated and also compared with the proposed setting.

The statistical tests showed that there is no significantly better active learning query strategy (including random selection) but information density was the best for the majority of the data sets. The reasons that there are no significantly better results for active learning may be the same as the reasons discussed in Section 7.2.5. Therefore, as future work, the hierarchical decomposition method should be integrated with new query strategies which for instance address the given challenges in Section 7.2.5, and which may provide significantly better results over random selection.

As another future work, this integration might be useful for collecting ground truth data set from a very large data repository (e.g. Fish4Knowledge repository) in a much quicker way as active learning strategies choose the most informative instances to be labelled and implicitly filter out other unlabelled samples.

7.4 Conclusions

In this chapter, we investigated feature selection integrated with active learning and the hierarchical decomposition method integrated with active learning.

The experiments showed that applying active learning with feature selection results in better classification both for active learning query strategies and random selection no matter the classifier used (Naive Bayes and SVM and especially for Naive Bayes). When the Naive Bayes classifier is used, with feature selection, it is possible to obtain better performance even in very early stages of active learning. Additionally, while using the Naive Bayes, random selection generally performed as well as active learning strategies: uncertainty, information density and maximum probability. When SVM is used as the classifier, information density with feature selection performed better (but not significantly) than random selection with feature selection in the majority of the experiments.

For hierarchical decomposition with active learning, the proposed setting (to find the scores of being from the majority and minority classes) was successful to obtain substantial or best performances using different active learning strategies even with fewer training data compared to using all training data. The results showed that information density was the best for the majority of the data sets while statistical tests showed that there was no significantly better active learning query strategy and random selection can sometimes perform just as well.

In conclusion, for both methods (active learning integrated with feature selection and the hierarchical decomposition method integrated with active learning) better active learning query strategies need to be developed to obtain significantly better performance than random selection.

Chapter 8

Conclusions

This thesis has explored three novel supervised learning frameworks for detection of unusual fish trajectories. The hierarchical decomposition framework, which performed the best out of these three, is the main focus of this thesis. This framework is also a general method for imbalanced data classification and uses clustered and labelled data. When applied to classifying fish trajectories, this framework used subsets of features extracted from fish trajectories.
The preceding chapters present the following original contributions:

1. A novel method to filter out large amount of normal fish trajectories with low time complexity.

2. A novel approach to unusual fish trajectory detection using novel trajectory descriptions which have not previously been used for fish behaviour analysis.

3. A novel approach to unusual fish trajectory detection which builds a feature or class taxonomy independent hierarchy. Additionally, this approach is a general method for imbalanced data set classification.

4. A comprehensive evaluation of active learning with feature selection using most popular query strategies and random selection. Results showed that query strategies used and the random selection perform better when feature selection is integrated with active learning. A novel approach to integrate the proposed hierarchical decomposition method with active learning which results in best and substantial performance with less training data.

These contributions are summarised in the reminder of this chapter, together with a discussion of their limitations and future work.

8.1 Main Contributions, Limitations and Future Work

8.1.1 A Filtering Mechanism for Normal Fish Trajectories

Main Findings

The fish trajectories used in this thesis relies on the *Fish4Knowledge* repository which contains tera-scale underwater videos, meaning a much larger number of trajectories. When the number of trajectories is huge like this and the number of normal trajectories is much bigger than the number of unusual trajectories, normal trajectories can dominate unusual trajectories and detecting unusual trajectories become harder. To address this issue and quickly scan large number of trajectories, primitive fish motions were modelled as given in Chapter 3. The proposed rule based filtering method is the first algorithm for filtering normal fish trajectories in an unconstrained open sea environment with a low $FNrate$ (the positive class represents unusual trajectories and negative class represents normal trajectories). It was observed that this method is very efficient and can be used to collect ground truth data (even though the results still need a manual inspection). Additionally, the proposed method was able to distinguish true fish trajectories from the false fish trajectories which arise due to failures of the fish detection and tracking algorithms (which are affected by plant movements, object occlusions, typhoons and murky water, etc).

The results of the proposed filtering mechanism which were obtained using three different data sets (in Chapters 3, 4 and 5) showed that its unusual and normal trajectory detection rates are much better when it was applied to a single fish species from a single camera location.

The experiments presented in Chapter 3 showed that it is hard to distinguish normal and unusual fish trajectories using trajectory points as there are many overlapping descriptions between two classes. This motivated us to define other features which are extracted from trajectories such as velocity and shape based features which is the foundation of the methods presented in Chapters 4 and 5.

Limitations and Future Work

For straight and/or cross motions and being stationary, the pixel parameter was selected from $\{2, 4, 8, 16 \text{ and } 20\}$ to define the search area. The applied setting was reasonable by looking at the performance during training but on the other hand, an evolutionary or other search algorithms can be used to find better pixel parameter values.

As future work, the effectiveness of the proposed filtering method as a preliminary

step of an unusual trajectory detection method could be tested using much larger labelled fish trajectory data sets (such as more than 10 thousands, millions) to see if it is possible to improve the unusual fish trajectory detection rate of that method. Moreover, improved and/or additional rules which might consider the velocity, orientation of the fish etc. can be defined to decrease the false filtering.

8.1.2 The Flat Classifier

Main Findings

Given the challenges of the fish trajectory data (see Section 1.3), it is hard to distinguish normal and unusual fish trajectories using trajectory points alone. Therefore, novel trajectory descriptors were defined to describe trajectories. The trajectories were clustered using the best features chosen by the feature selection algorithm which uses labels of the trajectories. This flat classifier is a novel approach for unusual trajectory detection particulary since it uses labelled and clustered data together. It is based on clustering the data samples and then using several outlier detection rules. Improved performance especially compared to the proposed method in Chapter 3 was obtained for unusual fish trajectory detection in unconstrained underwater conditions. The successful results of the flat classifier motivated us to integrate it into a hierarchical decomposition method as given in Chapter 5.

Limitations and Future Work

Since the proposed method is based on clustering, it was necessary to encode trajectories with a fixed length vector form. To do that, the extracted properties (such as velocity, turn etc.) were used in terms of their statistical properties (mean, variance etc.) which fixed the vector length of the trajectory representation no matter the length of the trajectory.

The outlier detection parameter was taken as $\{$-1, -0.3, 0, 0.3, 0.6, 0.9, 1, 2, 3, and 6$\}$ for the experiments given in this chapter and these values were good enough to obtain good performances. However, as an alternative evolutionary algorithms can be adapted to find the optimal outlier detection threshold.

For feature selection, SFFS was used. This is a wrapper type feature selection method (method uses training and validation data labels to select best features). The comparisons between filter type feature selection methods and SFFS showed that filter based feature selection methods are not as successful as SFFS. SFFS was efficient and

effective with its substantial processing speed and the utility of the selected features. However, other wrapper type feature selection methods can also be adapted to train the proposed flat classifier.

As future work, the proposed method can also be applied to larger labelled fish trajectory data sets which might also include other fish species, camera locations and the time of the day.

8.1.3 Clustering Based Hierarchical Decomposition

Main Findings

A novel hierarchical decomposition method was proposed. This method uses outlier detection in combination with clustering to detect unusual fish trajectories and also to classify imbalanced data sets. This new hierarchical decomposition method does not use any fixed hierarchy based on features and/or classes. By being based on clustering, it is different from common hierarchical methods which use supervised learning. Different feature spaces are used to build the hierarchy. The hierarchy decomposition method significantly improved performance on unusual fish trajectory analysis. Furthermore, results obtained when the proposed method was applied to imbalanced data sets showed that the proposed method is successful especially when the distribution of the minority class is sparser than the majority class. It performs well when the class imbalanced ratio (the number of minority class samples over majority class samples) is low. It is successful if the majority and minority samples are highly overlapping and even when both classes contain varieties (such as having a mixture of distributions or having subclasses). Furthermore, the proposed method does not need the support of any cost function, algorithmic or data level algorithm (see Section 2.3) to handle imbalanced data sets.

The key observation and the justification for using a hierarchy is that some features allow partitioning of some samples, which then allows other features to be useful on the remaining samples. Therefore, this results in more specific features to be used once the data focuses onto specific subclasses. The proposed method comes up with multiple decision boundaries which are equal to the number of clusters in a hierarchy level. Those boundaries help the classification of data especially if the data is highly overlapping, and the imbalanced ratio between minority (unusual) and majority (normal) classes is high.

As presented in Chapter 5, the new trajectories or the new data samples can be

classified using different heuristics. If the aim is rare class detection such as unusual fish trajectory detection, then the heuristic "decision as an unusual trajectory at any level stops classification of the new trajectory while decision as a normal trajectory send the new trajectory to the next level" should be used since this heuristic produce better $TPrate$ (minority class classification) with high $TNrate$ (majority class classification). Other heuristics given in the same chapter usually results in better $TNrate$ as compared to $TPrate$.

The computational complexity of the proposed hierarchy decomposition method during training is high as it includes SFFS which is not very efficient especially for high dimensional data sets. This can be seen as a disadvantage but by implementing it in parallel on a task farming architecture as given in [2] this can be overcome. On the other hand, more importantly, the proposed method's testing complexity is low which only requires a few distance calculations between the closest clusters at each level and the new trajectory (data point).

Limitations and Future Work

The high training computational complexity of the proposed hierarchical decomposition method is one of the biggest limitations. As mentioned above, especially for high dimensional data sets, feature selection part of the proposed method should be parallelised for efficiency.

Being based on a heuristic can be seen as a shortcoming since it might be hard to decide which heuristic should be applied. At least for rare class detection, we propose using the heuristic given in Figure 5.2 and successful performance was shown using different data sets. On the other hand, for the interested readers different heuristics can be proposed and applied.

As future work, the proposed method can be applied to larger labelled fish data sets which might also include other fish species, camera locations and the time of the day. Even though its performance was tested using 20 public imbalanced data sets and 300 synthetic data sets, it can still be tested with more imbalanced data sets where especially the dimensionality of the data sets is higher.

8.1.4 Active Learning with Imbalanced Data Sets

Main Findings

The integration of the proposed hierarchical decomposition method with active learn-

ing was examined. It was observed that with a proper active learning query strategy, best performance can be obtained with less training data. This implies that data collection for ground truth construction can be realised more efficiently where active learning query strategies determine the most informative instances. Moreover, active learning with feature selection was also investigated. To the best of our knowledge, this was the most comprehensive examination of active learning with feature selection. The results showed that by using feature selection the performance of active learning for all query strategies and random selection reached the peak performance earlier by achieving better performance than active learning without feature selection. Additionally, it was observed that the performance of random selection is generally as good as other active learning query selection methods.

Limitations and Future Work

The results of active learning with feature selection and hierarchical decomposition integrated with active learning did not perform significantly better than random selection. Future work should investigate the use of better query strategies.

Active learning with feature selection results are not as smooth as results without feature selection. This might be because of the change in feature space at each step of active learning. However, the maximum performance rate is reached very quickly for all query strategies when feature selection is used (especially when Naive Bayes is used as the classifier). This suggests that training can be stopped rather quickly, but it is unclear how to define when to stop precisely. This can be investigated as future research as well.

The analysis both for active learning with feature selection and also for integration of the hierarchical decomposition method into active learning can be enlarged using more data sets, even though the data sets used in Chapter 7 cover different imbalanced ratios, varied in terms of class overlap, the number of samples, the number of features. To see the improvement that feature selection provides to active learning, data sets having larger dimensionality can also be applied where the feature selection should have even greater benefit. Moreover, other query strategies can also be investigated.

Since the training of the hierarchical decomposition method and the proposed setting for its integration with active learning has high time complexity, at each iteration of the active learning more informative samples were chosen to be labelled compared to the analysis with Naive Bayes and Support Vector Machine. Even with these settings, we were still able to show that substantial performance can be obtained with less

training data. As future work, the same experiments can be repeated by choosing fewer samples at each active learning iteration.

Lastly, it was shown that the proposed approach for integration of the hierarchical decomposition method with active learning which involves calculating a GMM from the closest cluster is useful and successful. However, as future work other approaches such as using logistic regression to estimate the probabilities from distances can be integrated and also compared with the proposed approach.

Bibliography

[1] Liwicki M. and Bunke H. Hmm-based on-line recognition of handwritten whiteboard notes. In *Proceedings of 10th International Workshop on Frontiers in Handwriting*, pages 595–599, 2006.

[2] McDonagh S., Beyan C., Huang P. X., and Fisher R. B. Applying semi-synchronised task farming to large-scale computer vision problems. *The International Journal of High Performance Computing Applications*, pages 1–24, 2014.

[3] Majecka B. Statistical models of pedestrian behaviour in the forum. In *Master Thesis, School of Informatics, University of Edinburgh*, 2009.

[4] Hayashi Y. Neural expert system using fuzzy teaching input and its application to medical diagnosis. *Information Sciences Applications*, 1:47–58, 1994.

[5] Smith J. W., Everhart J. E., Dickson W. C., Knowler W. C., and Johannes R. S. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265, 1988.

[6] Kubat M., Holte R., and Matwin S. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30:195–215, 1998.

[7] Blake C. L and Merz C. J. Uci repository of machine learning databases. *http://archive.ics.uci.edu/ml/ Accessed Oct 28*, 2013.

[8] Brehmer P., Chi T. D., and Mouillot D. Amphidromous fish school migration revealed by combining fixed sonar monitoring (horizontal beaming) with fishing data. *Journal of Experimental Marine Biology and Ecology*, 334(1):139–150, 2006.

[9] Spampinato C. and Palazzo S. Hidden markov models for detecting anomalous fish trajectories in underwater footage. In *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing*, pages 23–26, 2012.

[10] Graham N., Jones E. G., and Reid D. G. Review of technological advances for the study of fish behavior in relation to demersal fishing trawls. *ICES Journal of Marine Science*, 61:1036–1043, 2004.

[11] Boom B. J., He J., Palazzo S., Huang P. X., Beyan C., Chou H., Lin F., Spampinato C., and Fisher R. B. Research tool for the analysis of underwater camera surveillance footage. *Ecological Informatics*, pages 2383–2397, 2013.

[12] Nadarajan G., Chen-Burger Y. H., and Fisher R. B. A knowledge-based planner for processing unconstrained underwater videos. In *IJCAI Workshop on Learning Structural Knowledge From Observations*, 2009.

[13] Huang P. X., Boom B. J., and Fisher B. F. Underwater live fish recognition using a balanced-guaranteed optimized tree. In *Proceedings of the 11th Asian Conference on Computer Vision (ACCV), Lecture Notes in Computer Science*, volume 7724, pages 422–433, 2013.

[14] Piciarelli C., Micheloni C., and Foresti G. L. Trajectory-based anomalous event detection. *IEEE Transactions On Circuits and Systems for Video Technology*, 18(11):1544–1554, November 2008.

[15] Morris B. T. and Trivedi M. M. Trajectory learning for activity understanding: Unsupervised, multilevel and long-term adaptive approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2287–2301, November 2011.

[16] Turaga P., Chellappa R., Subrahmanian V. S., and Udrea O. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[17] Morris B. T. and Trivedi M. M. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1114–1127, August 2008.

[18] Spampinato C., Giordano D., Salvo R. D., Chen-Burger Y., Fisher R. B., and Nadarajan G. Automatic fish classification for underwater species behavior understanding. In *Proceedings of First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams*, pages 45–50, 2010.

[19] Spampinato C., Palazzo S., Giordano D., Kavasidis I., Lin F. P., and Lin Y. T. Covariance based fish tracking in real-life underwater environment. In *Proceedings of International Conference on Computer Vision Theory and Applications*, pages 409–414, 2012.

[20] Thida M., Eng H., and Chew B. F. Automatic analysis of fish behaviors and abnormality detection. In *Proceedings of International Association for Pattern Recognition Conference on Machine Vision Applications*, pages 8–18, 2009.

[21] Nogita S., Baba K., Yahagi H., Watanabe S., and Mori S. Acute toxicant warning system based on a fish movement analysis by use of AI concept. In *International Workshop: Artificial Intelligence for Industrial Applications*, pages 273–276, 1988.

[22] Schalie W. H., Shedd T. R., Knechtges P. L., and Widder M. W. Using higher organisms in biological early warning systems for real-time toxicity detection. *Biosensors and Bioelectronics*, 16(7):457–465, 2001.

[23] Papadakis V. M., Papadakis I. E., Lamprianidou F., Glaroulos A., and Kentouri M. A computer vision system and methodology for the analysis of fish behavior. *Aquacultural Engineering*, 46:53–59, 2012.

[24] Serra-Toro C., Montoliu R., Traver V. J., and Hurtado-Melgar I. M. Assessing water quality by video monitoring fish swimming behaviour. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, pages 428–431, 2010.

[25] Chew B. F., Eng H. L., and Thida M. Vision-based real-time monitoring on the behavior of fish school. In *Proceedings of International Association for Pattern Recognition Conference on Machine Vision Applications*, pages 3–16, 2009.

[26] Mancera J. M., Vargas-Chacoff L., Garcia-Lopez A., Kleszczynska A., Kalamarz H., Martinez Rodriguez G., and Kulczykowska E. High density and

food deprivation affect arginine vasotocin, isotocin and melatonin in gilthead sea bream (sparus auratus). *Comparative Biochemistry and Physiology Part A*, 149:92–97, 2008.

[27] Pinkiewicz T. H., Purser G. J., and Williams R. N. A computer vision system to analyze the swimming behavior of farmed fish in commercial aquaculture facilities: A case study using cage-held atlantic salmon. *Aquacultural Engineering*, 45:20–27, 2011.

[28] Kato S., Nakagawa T., Ohkawa M., Muramoto K., Oyama O., Watanabe A., Nakashima H., Nemoto T., and Sugitani K. A computer image processing system for quantification of zebrafish behavior. *Journal of Neuroscience Methods*, 134:1–7, 2004.

[29] Amer M., Bilgazyev E., Todorovic S., Shah S., Kakadiaris I., and Ciannelli L. Fine-grained categorization of fish motion patterns in underwater videos. In *International Conference on Computer Vision Workshops*, pages 1488–1495, 2011.

[30] Xu J., Liu Y., Cui S., and Miao X. Behavior al responses of tilapia (oreochromis niloticus) to acute fluctuations in dissolved oxygen levels as monitored by computer vision. *Aquacultural Engineering*, 35(3):207–217, 2006.

[31] Sillito R. R. and Fisher R. B. Semi-supervised learning for anomalous trajectory detection. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 227–238, 2008.

[32] Li C., Han Z., Ye Q., and Jiao J. Abnormal behavior detection via sparse reconstruction analysis of trajectory. In *Proceedings of International Conference on Image and Graphics*, pages 807–810, 2011.

[33] Makris D. and Ellis T. J. Spatial and probabilistic modelling of pedestrian behaviour. In *Proceedings of British Machine Vision Conference (BMVC)*, volume 2, pages 557–566, 2002.

[34] Naftel A. and Khalid S. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12:227–238, 2006.

[35] Brand M. and Kettnaker V. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[36] Porikli F. Learning object trajectory patterns by spectral clustering. In *Proceedings of IEEE Conference Multimedia Expo*, volume 2, pages 1171–1174, 2004.

[37] Nair V. and Clark J. J. Automated visual surveillance using hidden markov models. In *Proceedings of 15th Vision Interface Conference*, pages 88–92, 2002.

[38] Bashir F., Wu Q., Khokhar A., and Schonfeld D. Hmm-based motion recognition system using segmented PCA. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2286–2289, 2005.

[39] Sillito R. R. and Fisher R. B. Parametric trajectory representations for behaviour classification. In *Proceedings of British Machine Vision Conference (BMVC)*, 2009.

[40] Zhong H., Shi J., and Visontai M. Detecting unusual activity in video. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 819–826, 2004.

[41] Porikli F. and Haga T. Event detection by eigenvector decomposition using object and frame features. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 114–122, 2004.

[42] Hu W., Xiao X., Fu Z., Xie D., Tan T., and Maybank S. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, 2006.

[43] Zhang D., Gatica-Prez, D., Bengio S., and McCowan I. Semi-supervised adapted HMMs for unusual event detection. In *Proceedings of IEEE Computer Vision Pattern Recognition (CVPR)*, volume 1, pages 611–618, 2005.

[44] Luhr S., Venkatesh S., West G., and Bui H. H. Duration abnormality detection in sequence of human activity. *Technical Report TR-2004/02, Curtin University of Technology*, 2004.

[45] Duong T. V., Bui H. H., Phung D. Q., and Venkatesh S. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 838–845, 2005.

[46] Xiang T. and Gong S. Video behaviour abnormality detection using reliability measure. In *Proceedings of British Machine Vision Conference*, 2005.

[47] Xiang T. and Gong S. Video behaviour profiling and abnormality detection without manual labelling. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1238–1245, 2005.

[48] Xiang T. and Gong S. Incremental visual behaviour modelling. In *IEEE Visual Surveillance Workshop*, pages 65–72, 2006.

[49] Loy C. C., Xiang T., and Gong S. Detecting and discriminating behavioural anomalies. *Pattern Recognition*, 44:117–132, 2011.

[50] Owens J. and Hunter A. Application of the self-organizing map to trajectory classification. In *Proceedings of IEEE International Workshop on Visual Surveillance*, pages 77–83, 2000.

[51] Izo T. and Grimson W. E. L. Unsupervised modeling of object tracks for fast anomaly detection. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 529–532, 2007.

[52] Jung C. R., Hennemann L., and Musse S. R. Event detection using trajectory clustering and 4-D histograms. *IEEE Transactions on Circuit and Systems for Video Technology*, 18(11):1565–1575, 2008.

[53] Figueiredo M. T. A. and Jain A. K. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.

[54] Anjum N. and Cavallaro A. Multifeature object trajectory clustering for video analysis. *IEEE Transactions on Circuits and Systems For Video Technology*, 18(11):1555–1564, November 2008.

[55] Hu H. Y., Qu Z. W., and Li Z. H. Multi-level trajectory learning for traffic behavior detection and analysis. *Journal of the Chinese Institute of Engineers*, 37(8):1–12, 2014.

[56] Choudhary A., Pal M., Banerjee S., and Chaudhury S. Unusual activity analysis using video epitomes and pLSA. In *Proceedings of 6th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 390–397, 2008.

[57] Varadarajan J. and Odobez J. M. Topic models for scene analysis and abnormality detection. In *Proceedings of International Conference on Computer Vision Workshop*, pages 1338–1345, 2009.

[58] Jeong H., Chang H. J., and Choi J. Y. Modelling of moving object trajectories by spatio-temporal learning for abnormal behaviour detection. In *Proceedings of IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 119–123, 2011.

[59] Zhu X., Liu Z., and Zhang J. Human activity clustering for online anomaly detection. *Journal of Computers*, 6(6):1071–1079, 2011.

[60] Ivanov I., Dufaux F., Ha T. M., and Ebrahimi T. Towards generic detection of unusual events in video surveillance. In *Proceedings of Advanced Video and Signal Based Surveillance (AVSS)*, pages 61–66, 2009.

[61] Hospedales T. M., Li J., Gong S., and Xiang T. Identifying rare and subtle behaviours: A weakly supervised joint topic model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2451–2464, 2011.

[62] Junejo I. N., Javed O., and Shah M. Multi feature path modeling for video surveillance. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, pages 716–719, 2004.

[63] Khalid S. and Naftel A. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. In *Proceedings of ACM International Workshop on Video Surveillance and Sensor Networks*, pages 45–52, 2005.

[64] Makris D. and Ellis T. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man. and Cybernetics*, 35(3):397–408, June 2005.

[65] Dahlbom A. and Niklasson L. Trajectory clustering for coastal surveillance. In *IEEE International Conference on Information Fusion*, pages 1–8, 2007.

[66] Basharat A., Gritai A., and Shah M. Learning object motion patterns for anomaly detection and improved object detection. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[67] Wiliem A., Madasu V., Boles W., and Yarlagadda P. Detecting uncommon trajectories. In *Proceedings of Digital Image Computing: Techniques and Applications (DICTA)*, pages 398–404, 2008.

[68] Bouttefroy P. L. M., Bouzerdoum A., Phung S. L., and Beghdadi A. Abnormal behavior detection using a multi-modal stochastic learning approach. In *Proceedings of Intelligent Sensors, Sensor Networks and Information Processing*, pages 121–126, 2008.

[69] Zelniker E. E., Gong S., and Xiang T. Global abnormal behaviour detection using a network of cctv cameras. In *Proceedings of the Eighth International Workshop on Visual Surveillance*, 2008.

[70] Wiliem A., Madasu V., Boles W., and Yarlagadda P. A context-based approach for detecting suspicious behaviours. In *Proceedings of Digital Image Computing: Techniques and Applications (DICTA)*, pages 146–153, 2009.

[71] Jiang F., Wu, Y., and Katsaggelos A. K. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, 18(4):907–913, April 2009.

[72] Espinosa-Isidron D. L. and Garcia-Reyes E. B. A new dissimilarity measure for trajectories with applications in anomaly detection. *Lecture Notes in Computer Science Progress in Pattern Recognition, Image Analysis, Computer Vision and Applications*, 6419:193–201, November 2010.

[73] Al-Khateeb H. and Petrou M. An extended fuzzy SOM for anomalous behaviour detection. In *Proceedings of IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 31–36, 2011.

[74] Shi F., Zhou Z., Xiao J., and Wu W. Robust trajectory clustering for motion segmentation. In *Proceedings of IEEE International Conference on Computer Vision(ICCV)*, pages 3088–3095, 2013.

[75] Morris B. T. and Trivedi M. M. Understanding vehicular traffic behavior from video: a survey of unsupervised approaches. *Journal of Electronic Imaging*, 22(4):1–15, October-December 2013.

[76] Dickinson P. and Hunter A. Using inactivity to detect unusual behaviour. In *Proceedings of IEEE Workshop on Motion and Video Computing*, pages 1–6, 2008.

[77] Yin J. and Meng Y. Abnormal behavior recognition using self-adaptive hidden markov models. In *Proceedings of International Conference on Image Analysis and Recognition*, pages 337–346, 2009.

[78] Nishio S., Okamoto H., and Babaguchi N. Hierarchical anomality detection based on situation. In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR)*, pages 1108–1111, 2010.

[79] Akoz O. and Karsligil M. E. Video-based traffic accident analysis at intersections using partial vehicle trajectories. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 4693–4696, 2010.

[80] Wang Y., Wang D., and Chen F. Abnormal behavior detection using trajectory analysis in camera sensor networks. *International Journal of Distributed Sensor Networks*, pages 1–9, 2014.

[81] Piciarelli C. and Foresti G. L. Anomalous trajectory detection using support vector machines. In *Proceedings of Advanced Video and Signal Based Surveillance*, pages 153–158, 2007.

[82] Ma Y. and Li M. Detection for abnormal event based on trajectory analysis and FSVM. In *Proceedings of International Conference on Intelligent Computing*, pages 1112–1120, 2007.

[83] Lui C., Wang G., Ning W., Lin X., Li L., and Liu Z. Anomaly detection in surveillance video using motion direction statistics. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 717–720, 2010.

[84] Li J., Hospedales T. M., Gong S., and Xiang T. Learning rare behaviours. In *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 292–307, 2010.

[85] Hendel A., Weinshall D., and Peleg S. Identifying surprising events in videos using bayesian topic models. In *Proceedings of Asian Conference on Computer Vision (ACCV), Part 3, LNCS*, volume 6494, pages 448–459, 2011.

[86] Galar M., Fernandez A., Barrenechea E., Bustince H., and Herrera F. A review on ensembles for the class imbalance problem: Bagging, boosting, and hybrid-based approaches. *IEEE Transactions On Systems, Man, and Cybernetics Part C*, 42(4):463–484, July 2012.

[87] Galar M., Fernandez A., Barrenechea E., and Herrera F. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, 46:3460–3471, 2013.

[88] Vajda S. and Fink G. A. Strategies for training robust neural network based digit recognizers on unbalanced data set. In *Proceedings of the 12th Conference on Frontiers in Handwriting Recognition*, pages 148–153, 2010.

[89] Garcia S. and Herrera F. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. *Evolutionary Computation*, 17:275–306, 2009.

[90] Liu W. and Chawla S. Class confidence weighted kNN algorithm for imbalanced data sets. In *Proceedings of Pasific-Asia Conference on Knowledge Discovery and Data Mining, PART 2, LNAI*, pages 345–356, 2011.

[91] Osuna E., Freund R., and Girosi F. Support vector machines: Training and applications. *Technical Report Massachusetts Institute of Technology Cambridge, MA, USA*, 1997.

[92] Veropoulos K., Cristianini N., and Campbell C. Controlling the sensitivity of support vector machines. In *Proceedings of Joint Conference on Artificial Intelligence (IJCAI)*, pages 55–60, 1999.

[93] Wu G. and Chang E. Y. Class-boundary alignment for imbalanced dataset learning. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 49–56, 2003.

[94] Akbani R., Kwek S., and Japkowicz N. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, pages 39–50, 2004.

[95] Chawla V. C., Bowyer K. W., Hall L. O., and Kegelmeyer W. P. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[96] Japkowicz N. and Stephen S. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6:429–449, 2002.

[97] Kubat M. and Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 179–186, 1997.

[98] Japkowicz N. The class imbalance problem: Significance and strategies. In *Proceedings of International Conference on Artificial Intelligence, Special Track on Inductive Learning*, pages 111–117, 2000.

[99] Ling C. and Li C. Data mining for direct marketing problems and solutions. In *Proceedings of International Conference on on Knowledge Discovery and Data Mining*, pages 217–225, 1998.

[100] Tang Y., Zhang Y. Q., Chawla N. V., and Krasser S. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems Man. Cybernetics Part B*, 39(1):281–288, 2009.

[101] Chen C., Liaw A., and Breiman L. Using random forest to learn imbalanced data. *Technical Report 666, Statistics Department, University of California at Berkeley, Available at http://www.stat.berkeley.edu/users/chenchao/666.pdf 2004*, 2004.

[102] Johnson R. A., Chawla N. V., and Hellman J. J. Species distribution modeling and prediction: A class imbalance problem. In *Proceedings of Intelligent Data Understanding (CIDU)*, pages 9–16, 2012.

[103] Han H., Wang W. Y., and Mao B. H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. *Advances in Intelligent Computing*, pages 878–887, 2005.

[104] Chawla N. V., Lazarevic A., Hall L. O., and Bowyer K. W. Smoteboost: Improving prediction of the minority class in boosting. In *Proceedings of the Seventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, 2003.

[105] Hu S., Liang Y., Ma L., and He Y. MSMOTE: Improving classification performance when training data is imbalanced. In *Proceedings of the 2nd International Workshop on Computer Science Engineering*, volume 2, pages 13–17, 2009.

[106] Klement W., Wilk S., Michaowski W., and Matwin S. Classifying severely imbalanced data. In *Proceedings of the 24th Canadian Conference on Advances in Artificial Intelligence*, pages 258–264, 2011.

[107] Fithian W. and Hastie T. Local case-control sampling: Efficient subsampling in imbalanced data sets. *The Annals of Statistics*, 42(5):1693–1724, 2014.

[108] Pazzani M., Merz C., Murphy P., Ali K., Hume T., and Brunk C. Reducing misclassification costs. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 217–225, 1994.

[109] Domingos P. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[110] Tan S. Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert System Applications*, 28(4):667–671, 2005.

[111] Fumera G. and Roli F. Cost-sensitive learning in support vector machines. In *Proceedings of Workshop Machine Learning, Methods and Applications, held in 8th meeting of the Italian Assoc. of Artificial Intelligence*, 2002.

[112] Drummond C. and Holte R. C. Exploiting the cost (in)sensitivity of decision tree splitting criteria. In *Proceedings of the 7th International Conference on Machine Learning (ICML)*, pages 239–246, 2000.

[113] Williams D. P., Myers V., and Silvious M. S. Mine classification with imbalanced data. *IEEE Geosciences And Remote Sensing Letters*, 6(3):528–532, 2009.

[114] Breiman L. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[115] Freund Y. and Schapire R. E. Decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[116] Radivojac P., Chawla N. V., Dunker K., and Obradovic Z. Classification and knowledge discovery in protein databases. *Journal of Biomedical Informatics*, 37(4):224–239, 2004.

[117] Liu X. Y., Wu J., and Zhou Z. H. Exploratory undersampling for class imbalance learning. *IEEE Transactions on Systems Man. Cybernetics Part B*, 39(2):539–550, 2009.

[118] Seiffert C., Khoshgoftaar T. M., Hulse J., and Napolitano A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems Man. Cybernetics Part B*, 20(1):185–197, 2010.

[119] Wang B. X. and Japkowicz N. Boosting support vector machines for imbalanced data sets. *Lecture Notes in Artificial Intelligence*, 4994:38–47, 2008.

[120] Batuwita R. and Palade V. Adjusted geometric-mean: A novel performance measure for imbalanced bioinformatics dataset learning. *Journal of Bioinformatics and Computational Biology*, 10(4):1–23, 2012.

[121] Huang J. and Ling C. X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[122] Barandela R., Sanchez J. S., Garcia V., and Rangel E. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36:849–851, 2003.

[123] Rao K. N., Rao T. V., and Lakshmi D. R. A novel class imbalance learning method using subset filtering. *International Journal of Scientific and Engineering Research*, 3(9):1–9, September 2012.

[124] Silla C. N. and Freitas A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1):31–72, 2010.

[125] Cesa-Bianchi N., Gentile C., and Zaniboni L. Incremental algorithms for hierarchical classification. *The Journal of Machine Learning Research*, 7:31–54, 2006.

[126] Wu F., Zhang J., and Honavar V. Learning classifiers using hierarchically structured class taxonomies. In *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation, Springer 3607*, pages 313–320, 2005.

[127] Li T. and Ogihara M. Music genre classification with taxonomy. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, pages 197–200, 2005.

[128] Bennett P. N. and Nguyen N. Refined experts: Improving classification in large taxonomies. In *Proceedings of the 32th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 11–18, 2009.

[129] Silla C. N. and Freitas A. A. Selecting different protein representations and classification algorithms in hierarchical protein function prediction. *Intelligent Data Analysis Journal*, 15(6):979–999, 2011.

[130] Kumar S., Ghosh J., and Crawford M. M. Hierarchical fusion of multiple classifiers for hyperspectral data analysis. *Pattern Analysis and Applications*, 5:210–220, 2002.

[131] Chen Y., Crawford M. M., and Ghosh J. Integrating support vector machines in a hierarchical output space decomposition framework. In *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing*, volume 2, pages 949–952, 2004.

[132] Hao P. Y., Chiang J. H., and Tu Y. K. Hierarchically SVM classification based on support vector clustering method and its application to document categorization. *Expert Systems with Applications*, 33:627–635, 2007.

[133] Freitas C. O. A., Oliveira L. S., Aires S. B. K., and Bortolozzi F. Metaclasses and zoning mechanism applied to handwriting recognition. *Journal of Universal Computer Science*, 14(2):211–223, 2008.

[134] Epshtein B. and Ullman S. Feature hierarchies for object classification. In *Proceedings of International Conference on Computer Vision (ICCV)*, pages 220–227, 2005.

[135] Peng Y., Lin C., and Sun M. Audio classification using binary hierarchical classifiers with feature selection for healthcare applications. In *Proceedings of IEEE International Symposium on Circuits and Systems*, pages 3238–3241, 2008.

[136] Freeman C., Kulic D., and Basir O. Joint feature selection and hierarchical classifier design. In *Proceedings of IEEE International Conference on Systems Man and Cybernetic*, pages 1728–1734, 2011.

[137] Fu Y., Zhu X., and Li B. A survey on instance selection for active learning. *Knowledge Information Systems*, 35:249–283, 2013.

[138] Settles B. Active learning literature survey. *Computer Sciences Technical Report 1648, University of Wisconsin-Madison*, 2009.

[139] Uguroglu S. Robust learning with highly skewed category distributions. In *PhD Thesis, Carnegie Mellon University, School of Computer Science*, 2013.

[140] Jingrui H. Analysis of rare categories. *Cognitive Technologies, Springer 2012, ISBN 978-3-642-22812-4*, pages 1–128, 2010.

[141] Haines T. and Xiang T. Active learning using dirichlet processes for rare class discovery and classification. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 1–11, 2011.

[142] Zhu J. and Hovy E. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the Joint meeting of the Conference on Empirical Methods in Natural Language Processing and the Conference on Natural Language Learning*, pages 783–790, 2007.

[143] Doyle S., Monaco J., Feldman M., Tomaszewski J., and Madabhushi A. An active learning based classification strategy for the minority class problem: Application to histopathology annotation. *BMC Bioinformatics*, 12:1471–2105, 2011.

[144] Holub A. and Perona P. Entropy-based active learning for object recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition, Workshop on Online Learning for Classification*, pages 1–8, 2008.

[145] Lewis D. and Gale W. A sequential algorithm for training text classifiers. In *Proceedings of SIGIR Conference on Research and Development in Information Retrieval, ACM/Springer*, pages 3–12, 1994.

[146] Ertekin S., Huang J., Bottou L., and Giles C. L. Learning on the border: Active learning in imbalanced data classification. In *Proceedings of ACM Conference on Information and Knowledge Management*, pages 127–136, 2007.

[147] Li S., Ju S., Zhou G., and Li X. Active learning for imbalanced sentiment classification. In *Proceedings of International Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148, 2012.

[148] Attenberg J. and Provost F. Inactive learning?: Difficulties employing active learning in practice. *ACM SIGKDD Explorations Newsletter*, 12(2), December 2010.

[149] Roy N. and McCallum A. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 441–448, 2001.

[150] Zhu X., Lafferty J., and Ghahramani Z. Combining active learning and semisupervised learning using gaussian fields and harmonic functions. In *Proceedings of the ICML Workshop on the Continuum from Labeled to Unlabeled Data*, pages 58–65, 2003.

[151] Moskovitch R., Nissim N., Stopel D., Feher C., Englert R., and Elovici Y. Improving the detection of unknown computer worms activity using active learning. In *Proceedings of the German Conference on Artificial Intelligence*, pages 489–493, 2007.

[152] Settles B. and Craven M. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078, 2008.

[153] Seung H. S., Opper M., and Sompolinsky H. Query by committee. In *Proceedings of the ACM Workshop on Computational Learning Theory*, pages 287–294, 1992.

[154] Raghavan H., Madani O., and Jones R. Active learning with feedback on both features and instances. *Journal of Machine Learning Research*, 7:1655–1686, 2006.

[155] Hu Y., Milios E., and Blustein J. Interactive feature selection for document clustering. In *Proceedings of 20th Symposium On Applied Computing, ACM Special Group on Applied Computing*, pages 1148–1155, 2011.

[156] Okabe M. and Yamada S. Interactive spam filtering with active learning and feature selection. In *Proceedings of WI-IAT Workshop*, volume 3, pages 165–168, 2008.

[157] Bilgic M. Combining active learning and dynamic dimensionality reduction. In *Proceedings of SIAM International Conference on Data Mining (SDM)*, pages 696–707, 2012.

[158] Bashir F. I., Khokhar A. A., and Schonfeld D. View-invariant motion trajectory based activity classification and recognition. In *Proceedings of ACM Multimedia Systems*, pages 45–54, 2006.

[159] Suk T. and Flusser J. Graph method for generating affine moment invariants. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, volume 2, pages 192–195, 2004.

[160] Li X., Hu W., and Hu W. Coarse-to-fine strategy for vehicle motion trajectory clustering. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, pages 591–594, 2006.

[161] Frey B. J. and Dueck D. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[162] Pudil P., Novovicova J., and Kittler J. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[163] He X., Cai D., and Niyogi P. Laplacian score for feature selection. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, volume 18, 2005.

[164] Cai D., Zhang C., and He X. Unsupervised feature selection for multi-cluster data. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 333–342, 2010.

[165] Yao D., Yang J., and Zhan X. An improved random forest algorithm for class-imbalanced data classification and its application in pad risk factors analysis. *The Open Electrical and Electronic Engineering Journal*, 7:62–70, 2013.

[166] Breiman L., Friedman J. H., Olshen R. A., and Stone C. J. Classification and regression trees. *Wadsworth and Brooks, Monterey, CA.*, 1984.

[167] Janssens J. H. M. Outlier detection with one-class classifiers from ML and KDD. In *Proceedings of International Conference on Machine Learning Applications*, pages 147–153, 2009.

[168] Hsiao K., Xu K., Calder J., and Hero A. O. Multi-criteria anomaly detection using pareto depth analysis. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2012.

[169] Das B., Krishnan N. C., and Cook D. J. Handling imbalanced and overlapping classes in smart environments prompting dataset. *Springer Book on Data Mining for Services in Studies in Computational Intelligence*, 2012.

[170] Alcala-Fdez J., Fernandez A., Luengo J., Derrac J., Garcia S., Sanchez L., and Herrera F. Keel data mining software tool: dataset repository, integration of algorithms and experimental analysis framework. *Journal of Multiple Valued Logic and Soft Computing, http://www.keel.es/dataset.php*, 17(2):255–287, 2011.

[171] Sigillito V. G., Wing S. P., Hutton L. V., and Baker K. B. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Technical Digest.*, 10:262–266, 1989.

[172] Weiss S. and Kapouleas I. An empirical comparison of pattern recognition. neural nets and machine learning classification methods,. In *Proceedings of the 11th International Joint Conference of Artificial Intelligence*, pages 781–787, 1989.

[173] Boutell M. R., Luo J., Shen X., and Brown C. M. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[174] Garcia S. and Herrera F. An extension on statistical comparisons of classifiers over multiple datasets for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[175] Garcia S., Fernandez A., Luengo J., and Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[176] Demsar J. Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[177] Wilcoxon F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[178] Holm S. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

[179] Chen Y. and Mani S. Active learning for unbalanced data in the challenge with multiple models and biasing. *Journal of Machine Learning Research*, 6:113–126, 2011.

[180] Xu G., Niu Z., Gao X., Cao Y., and Zhao Y. Active learning algorithm for threshold of decision probability on imbalanced text clasification based on protein-protein interaction documents. In *Proceedings of International Conference on Data Storage and Data Engineering*, pages 78–82, 2010.

[181] Lee M. S., Rhee J-K., Kim B-H., and Zhang B-T. AESNB: Active example selection with naive bayes classifier for learning from imbalanced biomedical data. In *Proceedings of IEEE International Conference on Bioinformatics and Bioengineering*, pages 15–21, 2009.

[182] Sohn S., Comeau D. C., Kim W., and Wilbur W. Term-centric active learning for naive bayes document classification. *Open Information Systems Journal*, 3:54–67, 2009.

[183] MATLAB version 8.3.0.532. Statistics and Machine Learning Toolbox version 9.0. *The MathWorks Inc.*, 2014.

[184] Ewans L. P. G., Adams N. M., and Anagnostopoulos C. When does active learning work? *IDA, Lecure Notes in COmputer Science*, 8207:174–185, 2013.

[185] Cawley G. Baseline methods for active learning. In *Journal of Machine Learning Research Conference and Workshop Proceedings*, volume 15, pages 47–47, 2011.

[186] Kittler J., Hatef M., Duin R. P. W., and Matas J. On combining classifiers. *Transaction of Pattern Analysis and Machine Learning*, 20(3):226–239, 1998.