Licentiate Thesis

Anomaly Detection in Trajectory Data for Surveillance Applications

Rikard Laxhammar
Computer Science

Studies from the School of Science and Technology at Örebro University 19
Örebro 2011

Anomaly Detection in Trajectory Data for Surveillance Applications

Studies from the School of Science and Technology at Örebro University 19

Rikard Laxhammar

Anomaly Detection in Trajectory Data for Surveillance Applications

This research has been supported by:

IN PARTNERSHIP WITH THE
Knowledge Foundation

© Rikard Laxhammar, 2011

Title: Anomaly Detection in Trajectory Data
for Surveillance Applications

Abstract

Abnormal behaviour may indicate important objects and events in a wide variety of domains. One such domain is intelligence and surveillance, where there is a clear trend towards more and more advanced sensor systems producing huge amounts of trajectory data from moving objects, such as people, vehicles, vessels and aircraft. In the maritime domain, for example, abnormal vessel behaviour, such as unexpected stops, deviations from standard routes, speeding, traffic direction violations etc., may indicate threats and dangers related to smuggling, sea drunkenness, collisions, grounding, hijacking, piracy etc. Timely detection of these relatively infrequent events, which is critical for enabling proactive measures, requires constant analysis of all trajectories; this is typically a great challenge to human analysts due to information overload, fatigue and inattention. In the Baltic Sea, for example, there are typically 3000–4000 commercial vessels present that are monitored by only a few human analysts. Thus, there is a need for automated detection of abnormal trajectory patterns.

In this thesis, we investigate algorithms appropriate for automated detection of anomalous trajectories in surveillance applications. We identify and discuss some key theoretical properties of such algorithms, which have not been fully addressed in previous work: sequential anomaly detection in incomplete trajectories, continuous learning based on new data requiring no or limited human feedback, a minimum of parameters and a low and well-calibrated false alarm rate. A number of algorithms based on statistical methods and nearest neighbour methods are proposed that address some or all of these key properties. In particular, a novel algorithm known as the *Similarity-based Nearest Neighbour Conformal Anomaly Detector* (SNN-CAD) is proposed. This algorithm is based on the theory of Conformal prediction and is unique in the sense that it addresses all of the key properties above.

The proposed algorithms are evaluated on real world trajectory data sets, including vessel traffic data, which have been complemented with simulated anomalous data. The experiments demonstrate the type of anomalous behaviour that can be detected at a low overall alarm rate. Quantitative results for learning and classification performance of the algorithms are compared. In particular, results from reproduced experiments on public data sets show

that SNN-CAD, combined with *Hausdorff distance* for measuring dissimilarity between trajectories, achieves excellent classification performance without any parameter tuning. It is concluded that SNN-CAD, due to its general and parameter-light design, is applicable in virtually any anomaly detection application. Directions for future work include investigating sensitivity to noisy data, and investigating long-term learning strategies, which address issues related to changing behaviour patterns and increasing size and complexity of training data.

Keywords: Anomaly detection, trajectory analysis, statistical methods, Conformal prediction, automated surveillance.

Acknowledgements

First and foremost, I would like to thank Göran Falkman, who is my main supervisor. You have shown great commitment to my research project at all time, and your advice and support have been invaluable to me. Many are the times when I have felt discouraged and resigned before our supervision meetings; yet, at each such occasion, I have left our meeting feeling relieved and encouraged. I would also like to express my sincerest gratitude to Klas Wallenius, who is my research mentor at Saab. Without your support and commitment, this research project would never have been realised in the first place. Your advice and feedback have been of high importance to my research and for the writing of this thesis.

This research has been supported by my employer Saab AB, and I am very grateful and proud for the unique opportunity they have offered me. I would like to extend a special thanks to Egils Sviestins at Saab, who is the co-author of one of my papers, and who has given me valuable feedback on my research, including a draft of this thesis and all my published papers. Other persons from Saab, who have given me feedback, and with whom I have had many interesting discussions, include Thomas Kronhamn, Martin Smedberg and Håkan Warston.

I am also very thankful to my current and former colleagues of the GSA research group at the University of Skövde: Christoffer Brax, with whom I have co-authored two papers and had many interesting discussions regarding anomaly detection, Lars Niklasson, who is my co-advisor, Fredrik Johansson, Anders Dahlbom, Maria Riveiro, Tina Erlandsson, who has given extensive feedback on a draft of this thesis, and Tove Helldin. I appreciate not only your feedback and our scientific discussions, but also our social intercourse during lunches, coffee breaks and other social activities. I would also like to acknowledge Stefan Arnborg, at the Royal Institute of Technology in Stockholm, who introduced me to the exciting research area of Conformal prediction.

Lastly, I would like to thank my beloved Kajsa for always being there, and for putting up with an, at times, absent-minded researcher at home.

Contents

1 Introduction **1**
- 1.1 Aim and Objectives . . . 3
- 1.2 Research Methodology . . . 4
- 1.3 Scientific Contribution . . . 5
- 1.4 Publications . . . 9
- 1.5 Thesis Outline . . . 14

2 Background **15**
- 2.1 Anomaly Detection . . . 15
  - 2.1.1 General Aspects of Anomaly Detection . . . 16
  - 2.1.2 Statistical Anomaly Detection . . . 20
  - 2.1.3 Other Anomaly Detection Algorithms . . . 26
- 2.2 Anomaly Detection in Trajectory Data . . . 29
  - 2.2.1 Representing Trajectory Data . . . 29
  - 2.2.2 Anomaly Detection in Video Surveillance . . . 29
  - 2.2.3 Anomaly Detection in Maritime Surveillance . . . 33
- 2.3 Conformal Prediction . . . 35
- 2.4 Hausdorff Distance for Shape Matching . . . 37

3 Conformal Anomaly Detection **41**
- 3.1 Issues with Previous Anomaly Detection Algorithms . . . 41
  - 3.1.1 Assumptions on the Underlying Distribution . . . 41
  - 3.1.2 Parameter-laden Algorithms . . . 42
  - 3.1.3 The Problem of Setting the Anomaly Threshold . . . 42
- 3.2 Conformal Prediction and Anomaly Detection . . . 43
  - 3.2.1 A Nonconformity Measure for Multi-class Anomaly Detection . . . 44
- 3.3 Conformal Anomaly Detection . . . 45
  - 3.3.1 The Conformal Anomaly Detector . . . 45
  - 3.3.2 Interpretation of a Conformal Anomaly . . . 46
  - 3.3.3 Online Semi-supervised Learning . . . 47

3.3.4 The Choice of Nonconformity Measure 47
3.3.5 Similarity-based Nearest Neighbour Conformal Anomaly Detector 48
3.4 Discussion 49
3.5 Summary 51

4 Anomaly Detection in Trajectory Data 53
4.1 Issues with Previous Algorithms 53
4.2 Point-based vs. Trajectory-based Anomaly Detection 55
4.3 Point-based Anomaly Detection 55
4.3.1 Statistical Approaches 55
4.3.2 Conformal Anomaly Detection Approach 60
4.4 Trajectory-based Anomaly Detection 61
4.4.1 A Dissimilarity Measure for Incomplete Trajectories . . 63
4.4.2 A Dissimilarity Measure for Complete Trajectories . . . 63
4.4.3 Considering Location, Speed and Course 63
4.5 Discussion 65
4.6 Summary 67

5 Empirical Investigations 69
5.1 Performance Measures 69
5.2 Overview of Experiments 71
5.2.1 Data Sets 71
5.2.2 Experiments 71
5.3 Anomaly Detection in Unlabelled Vessel Position-Velocity Data . 74
5.3.1 Data Description and Preprocessing 74
5.3.2 Experiment Design 74
5.3.3 Results 75
5.3.4 Analysis 75
5.3.5 Summary and Conclusion 80
5.4 Anomaly Detection in Labelled Vessel Trajectory Data 80
5.4.1 Extraction of Normal Training Data 81
5.4.2 Creation of Normal and Anomalous Test Data 84
5.4.3 General Setup and Parameters 86
5.4.4 Normalcy Learning – GMM vs. KDE 87
5.4.5 Sequential Anomaly Detection Delay – First Experiment 91
5.4.6 Sequential Anomaly Detection Delay – Second Experiment 97
5.4.7 Anomaly Detection – Precision and Recall 98
5.4.8 Anomaly Detection – False Alarm Rate 99
5.4.9 Summary 100
5.5 Anomaly Detection in Synthetic Trajectory Data 101
5.5.1 Data Description 102
5.5.2 Accuracy of Outlier Measure 103
5.5.3 Online Learning and Sequential Anomaly Detection . . . 104

5.6 Anomaly Detection in Real Video Trajectory Data 106
5.7 Discussion 107
5.7.1 Limitations 107
5.8 Summary 108

6 Conclusions **111**
6.1 Contributions 111
6.1.1 Summary of Contributions 116
6.2 Future work 117
6.3 Generalisation to Other Domains 119
6.4 Final Remarks 120

References **122**

List of Publications

I. **Laxhammar, R.** and Falkman, G. (2011) Sequential Conformal Anomaly Detection in Trajectories based on Hausdorff Distance, *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

II. **Laxhammar, R.** and Falkman, G. (2010) Conformal Prediction for Distribution-Independent Anomaly Detection in Streaming Vessel Data, *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques* (ACM), Washington D.C., USA, July 2010.

III. **Laxhammar, R.**, Falkman, G. and Sviestins, E. (2009) Anomaly Detection in Sea Traffic - a Comparison of the Gaussian Mixture Model and the Kernel Density Estimator, *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, July 2009.

IV. Brax, C., Niklasson, L. and **Laxhammar, R.** (2009) An ensemble approach for increased anomaly detection performance in video surveillance data, *Proceedings of the 12th International Conference on Information Fusion, Seattle*, USA, July 2009.

V. Brax, C. and **Laxhammar, R.** and Niklasson, L. (2008) Approaches for detecting behavioural anomalies in public areas using video surveillance data, *Proceedings of SPIE Electro-Optical and Infrared Systems: Technology and Applications V*, Cardiff, Wales, September 2008.

VI. **Laxhammar, R.** (2008) Anomaly detection for sea surveillance, *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

List of Figures

2.1 Illustration of Hausdorff distance between polygonal curves . . 38

3.1 Illustration of the problem with nearest neighbour NCM for anomaly detection 44

4.1 Illustration of route anomaly 62

5.1 First example of vessel anomalies detected by cell-based GMM . 76
5.2 Second example of vessel anomalies detected by cell-based GMM 77
5.3 Third example of vessel anomalies detected by cell-based GMM 78
5.4 Fourth example of vessel anomalies detected by cell-based GMM 79
5.5 Overview of vessel trajectories extracted from AIS database . . . 82
5.6 Plot of vessel trajectories from all vessel classes in port area of Gothenburg 83
5.7 Plot of vessel trajectories from subset of all vessel classes in port area of Gothenburg 83
5.8 Plot of first set of simulated anomalous vessel trajectories 85
5.9 Plot of second set of simulated anomalous vessel trajectories . . 86
5.10 Plot of vessel trajectories for a selected cell 89
5.11 Visualisation of PDF in position space for GMM 90
5.12 Visualisation of PDF in position space for KDE 90
5.13 Illustration of position anomaly detected by cell-based KDE in normal vessel trajectory data 94
5.14 Illustration of velocity vector anomaly detected by cell-based KDE in normal vessel trajectory data. 95
5.15 Plot of synthetic trajectories from public data set 102
5.16 Histogram over false negatives based on training data size for SNN-CAD during online learning and sequential anomaly detection 105
5.17 Plot of video trajectories from public data set 106

List of Algorithms

3.1 The Conformal Anomaly Detector (CAD) 46
3.2 Similarity-based Nearest Neighbour Conformal Anomaly Detector (SNN-CAD) 50
4.1 Iterative EM for estimating GMM with unknown number of components 57
4.2 Single Point Trajectory Nonconformity Measure (SPT-NCM) . . 60
4.3 Single Point Trajectory Conformal Anomaly Detector (SPT-CAD) 61

List of Tables

1.1 Research objectives, publications and thesis chapters. 13

5.1 Overview of experiments 72
5.2 Results normalcy modelling experiment 89
5.3 Detection delay on the anomalous segments of the first test set of vessel trajectories 92
5.4 Detection delay on the anomalous segments of the second test set of vessel trajectories 98
5.5 Classification performance on third test set of labelled vessel trajectories 99
5.6 Empirical false alarm rate for SNN-CAD on vessel trajectories . 100
5.7 Average accuracy for different outlier measures on a public set of simulated trajectories 103

List of Symbols

δ_H	Undirected Hausdorff distance
$\overrightarrow{\delta_H}$	Directed Hausdorff distance
S	Dissimilarity measure
A	Nonconformity measure
B	Multi-set
ϵ	Significance level
p	P-value
α	Nonconformity score
k	Number of nearest neighbours
P	Probability distribution
P_θ	Parametrised probability distribution
$p(x)$	Probability distribution for a discrete or continuos variable x
$Pr(\cdot)$	Probability of a specified event

List of Acronyms

AIS	Automatic Identification System
CAD	Conformal Anomaly Detector
CP	Conformal Prediction
DTW	Dynamic Time Warping
ED	Euclidean Distance
EM	Expectation-Maximization
FAR	False Alarm Rate
GMM	Gaussian Mixture Model
HD	Hausdorff Distance
HMM	Hidden Markov Model
IID	Independent Identically Distributed
KDE	Kernel Density Estimation
LCSS	Longest Common Sub-Sequence
LOF	Local Outlier Factor
MAP	Maximum a Posterior
ML	Maximum Likelihood
NCM	Non-Conformity Measure
PDF	Probability Density Function
SNN-NCM	Similarity-based Nearest Neighbour Non-Conformity Measure
SNN-CAD	Similarity-based Nearest Neighbour Conformal Anomaly Detector

SOM	Self-Organising Map
SPT-CAD	Single Point Trajectory Conformal Anomaly Detector
SPT-NCM	Single Point Trajectory Non-Conformity Measure
SVM	Supper Vector Machine

Chapter 1
Introduction

Abnormal behaviour may indicate important objects and events in a wide variety of domains. One such domain is intelligence and surveillance where there is a clear trend towards more and more advanced sensor systems producing huge amounts of *trajectory data* from moving objects, such as people, vehicles, vessels and animals. In the maritime domain, for example, abnormal vessel behaviour, such as unexpected stops, deviations from standard routes, speeding, traffic direction violations etc., may indicate threats and dangers related to smuggling, sea drunkenness[1], collisions (Danish Maritime Authority, 2003), grounding (Swedish Maritime Safety Inspectorate, 2004), terrorism[2], hijacking[3], piracy[4] etc. According to Rhodes (2009), "timely identification and assessment of anomalous activity within an area of interest is an increasingly important capability — one that falls under the enhanced situation awareness objective of higher-level fusion" . Timely detection of these relatively infrequent events, which is critical for enabling pro-active measures, requires constant analysis of all trajectories; this is typically a great challenge to a human analysts due to information overload, fatigue and inattention. In the Baltic sea, for example, there are typically 3000–4000 commercial vessels present that are monitored by a few human analysts[5]. Thus, there is a need for automated trajectory analysis.

In this thesis, we are mainly concerned with algorithms for automated learning and detection of abnormal trajectories in surveillance applications. The main contribution of our work is the proposal and evaluation of algorithms appropriate for sequential anomaly detection in trajectory data.

In the research fields *information fusion* and *data mining*, various computational methods have been proposed for supporting surveillance analysts in detecting abnormal and interesting behaviour. *Signature-based methods* as-

[1] http://www.swedishwire.com/economy/5738-drunk-captain-runs-aground-in-sweden

[2] http://www.globalsecurity.org/security/profiles/uss_cole_bombing.htm

[3] http://news.bbc.co.uk/2/hi/uk_news/8196640.stm

[4] http://en.wikipedia.org/wiki/Piracy_in_Somalia

[5] http://www.idg.se/2.1085/1.376546/sjobasis-battre-an-stalmannens-rontgensyn?utm_source=tip-friend&utm_medium=email

sume that specific models, such as rules and templates, for interesting behaviour can be defined a priori and used for automated pattern recognition in new data (Patcha and Park, 2007). Such models are constructed according to two main knowledge extraction strategies, which are adopted to various extents: incorporation of human *expert knowledge* regarding suspicious and interesting behaviours (Edlund et al., 2006; Dahlbom et al., 2009) and *supervised learning* from historical data of interesting behaviour (Fooladvandi et al., 2009). However, it has been argued that signature-based methods are not sufficient since accurate models of all possible behaviour of interest cannot be acquired in practise (Patcha and Park, 2007). The main reasons for this are lack of expert knowledge and data that cover the full spectrum of interesting behaviours, and practical knowledge engineering difficulties in encoding expert knowledge. According to Kraiman et al. (2002), "there is a need for robust, non-template-based processing techniques that monitor large tracking and surveillance data sets" . It has further been argued that the analysis should be focused towards detecting "strange" and abnormal patterns that deviate from the expected or "normal" patterns. Such approaches, typically refereed to as *anomaly*, *novelty* or *outlier detection* methods (Chandola et al., 2009), benefit from the fact that there are usually large amounts of historical data which can be exploited for learning normal behaviour.

A key issue when designing an anomaly detector is how to represent the data in which anomalies are to be found. Some anomaly detection methods assume that data is represented as points in a fixed *feature space*. This implies that a fixed number of features, e.g., the location at a fixed number of points in time, have to be extracted from each complete trajectory or trajectory segment. Other methods only assume that a *similarity* or *dissimilarity measure* is defined for pairs of trajectories or trajectory segments. The choice of features or similarity/dissimilarity measure essentially determines the type of anomalies that can be detected and is therefore of high importance.

Most of the proposed algorithms are essentially designed for *offline anomaly detectio*n in the sense that they assume that the *complete* trajectory has been observed before classifying it as anomalous or not. This is a limitation in, e.g., surveillance applications since it delays anomaly alarms and thus the ability to react to impending events. In contrast, algorithms for *online* or *sequential anomaly detection* allow detection in *incomplete* trajectories (Morris and Trivedi, 2008a), e.g., real-time detection of anomalous trajectories as they evolve.

With a few exceptions, *learning* in previously proposed algorithms for trajectory anomaly detection is *offline;* fixed model parameters and thresholds are typically estimated or tuned once based on a batch of historical data. But in many domains, normal behaviour keeps evolving and a current notion of normal behaviour might not be sufficiently representative in the future (Chandola et al., 2009). The advantage of *online learning* in the domain of surveillance has been discussed by Piciarelli and Foresti (2006) and Rhodes et al. (2007).

Anomaly detection algorithms typically require careful setting of multiple application specific parameters in order to achieve (near) optimal performance; trajectory anomaly detection is no exception to this. Indeed, Keogh et al. (2007) argue that most data mining algorithms are more or less *parameter-laden*, which is undesirable for several reasons. According to Markou and Singh (2003a), "an [anomaly] detection method should aim to minimise the number of parameters that are user set" .

The *anomaly threshold* is a central parameter in all anomaly detection algorithms, since it regulates the sensitivity to true anomalies and the rate of *false alarms*. Many algorithms rely on a *distance* or *density threshold* for deciding whether new data is anomalous or not. These distances or densities are typically not normalised and the procedures for setting the thresholds seem to be more or less ad-hoc and not very intuitive to, e.g., an operator of a surveillance system. The difficulty of tuning the anomaly threshold has consequences regarding the effectiveness and usefulness of an anomaly detection system. According to Axelsson (2000), "the false alarm rate is the limiting factor for the performance of the [anomaly] detection system". Indeed, Riveiro (2011) argues that "the primary and most important challenge that needs to be met for using [an anomaly detection] approach is the development of strategies to reduce the high false alarm rate". Hence, maintaining a *well-calibrated false alarm rate* is of critical importance in anomaly detection applications.

Different models and algorithms have previously been proposed for trajectory anomaly detection in, e.g., the domains of video surveillance and maritime surveillance. Yet, it may be argued that these algorithms typically suffer from drawbacks related to one or more of the issues discussed above: offline anomaly detection, offline learning, many parameters, tuning of the anomaly threshold and its relation to the false alarm rate. Moreover, most of the algorithms are only demonstrated or evaluated on simulated data sets and/or relatively small real world data sets with few anomalies; there is generally a lack of empirical results on fairly large real world data sets. Thus, it is unclear to what extent the proposed algorithms are appropriate for real surveillance applications.

1.1 Aim and Objectives

Following the discussion in the last paragraph above, we formulate the overall research aim of this thesis as follows:

Aim *Investigate properties and performance of algorithms for anomaly detection in trajectory data for surveillance applications, and propose new or updated algorithms that are better suited for this task.*

In order to address this aim, a number objectives are identified:

Objective 1: Identify important and desirable theoretical properties of algorithms for anomaly detection in surveillance applications.

Objective 2: Review and analyse previously proposed algorithms for anomaly detection in trajectory data.

Objective 3: Propose algorithms that are well-suited for anomaly detection in trajectory data.

Objective 4: Demonstrate feasibility and validity of proposed algorithms on real world data sets.

Objective 5: Identify suitable performance measures for evaluating algorithms for anomaly detection in trajectory data.

Objective 6: Evaluate proposed algorithms according to identified performance measures.

1.2 Research Methodology

In order to address the objectives stated in Section 1.1, we adopt a number of different research methods.

Starting with Objective 1 and 2, we perform two *literature reviews* and *literature analyses* (Berndtsson et al., 2002). The first review and analysis is focused on algorithms for anomaly detection in general. The second review and analysis is focused on algorithms for anomaly detection in trajectory data in surveillance applications. An important issue when undertaking a literature analysis is how to systematically search for previously published work that is relevant for the current research (Berndtsson et al., 2002). We adopt a number of different search strategies based on:

- Searching the internet in general and scientific databases in particular, using different combinations of selected keywords, such as "anomaly detection", "surveillance", "trajectory data" etc.
- Browsing annual proceedings for selected conferences, selecting a subset of papers for further reading based on title, abstract and/or keywords.
- Backwards citation chaining from relevant papers previously found.

In order to demonstrate the feasibility and validity of the proposed algorithms (Objective 4), and enable the evaluation of the algorithms (Objective 6), we develop an *implementation* (Berndtsson et al., 2002) for each of them. More specifically, we implement the proposed algorithms in MATLAB[6]. An important issue when developing an implementation of an algorithm is to ensure its *reliability* (Berndtsson et al., 2002), i.e., the robustness and correctness of the implementation. Most of the implemented algorithms are based on functions and subroutines from the standard MATLAB library and the official Statistics

[6] http://www.mathworks.com

toolbox[7] by MathWorks, which strengthens the reliability of the implementations. Moreover, for one of the proposed algorithms, a publicly available implementation of the corresponding core algorithm is used, which was developed and implemented by the original authors themselves.

In order to evaluate the proposed algorithms according to the identified performance measures (Objective 6) , we perform a series of *experiments* (Berndtsson et al., 2002) using the implementations of the corresponding algorithms and different trajectory data sets. Data mining algorithms are typically evaluated by measuring their performance on a *labelled* data set, which is often publicly available[8]. However, public availability of real world surveillance data is very limited due to proprietary issues etc. Moreover, real world surveillance data set is typically *unlabelled*, i.e., there is no information regarding which part of the data is actually normal or anomalous, and includes very few, if any, true anomalies; this may threaten validity and reliability of experimental results and conclusions. An alternative to real world data is to *simulate* data, which has the advantages that 1) the resulting data is labelled and 2) *reproducibility* of experiments is enhanced, under the assumption that the data can more easily be made publicly available (downloaded) or accurately re-created given the parameters and details of the simulation process. The main drawback of using simulated data is that validity or generalisability may be questioned if, e.g., the simulated anomalies do not reflect the actual anomalies encountered in a real surveillance application. That is, we are actually measuring performance on a different problem (detect simulated anomalies) than the one we aim to measure (detect true anomalies). In this thesis, we evaluate performance of the proposed algorithms using both real and simulated vessel trajectory data. Moreover, we reproduce two previously published experiments, which involve a labelled set of simulated trajectories and real video trajectories, respectively.

1.3 Scientific Contribution

This thesis is based on previously published work by the author. The published work has also been updated and extended with new theoretical and empirical results in this thesis. The main scientific contributions, listed below, are organised according to three research areas:

Conformal Prediction and Anomaly Detection

- **Description and discussion of how *Conformal prediction* (CP) (Vovk et al., 2005) can be adopted for multi-class anomaly detection applications** (Section 3.2).

  CP is a recent machine learning theory proposed for supervised learning

[7] http://www.mathworks.com/products/statistics/

[8] see, e.g., the UCI Machine Learning Repository: http://archive.ics.uci.edu/ml/

and prediction with valid confidence (Vovk et al., 2005). A *multi-class anomaly detector* is an algorithm that, in addition to detecting anomalies, is able to distinguish between multiple normal classes in data (Chandola et al., 2009). In this thesis, we discuss a novel application of CP for multi-class anomaly detection. This includes identification and discussion of key theoretical properties of an anomaly detector based on a conformal predictor. The main design parameter in CP is known as the *nonconformity measure* (NCM) (Vovk et al., 2005). Various NCMs have previously been proposed for supervised classification applications. We adapt one such NCM, which is based on distance to k-nearest neighbours in feature space, making it more suitable for multi-class anomaly detection applications.

- **Proposal of the *Conformal Anomaly Detector* (CAD), an general algorithm for anomaly detection with well-calibrated false alarm rate** (Section 3.3.1).

  In many applications, we are only interested in detecting anomalies and not determining which (if any) of the normal classes that best fits a new example. We refine our initial work on applying CP for anomaly detection, resulting in CAD, which is a *one-class anomaly detector* that is computationally more efficient than the corresponding multi-class conformal predictor. We identify and discuss key theoretical properties of CAD, including its well-calibrated false alarm rate and application independent anomaly threshold.

- **Proposal of the *Similarity-based Nearest Neighbour Conformal Anomaly Detector* (SNN-CAD), which is an instance of CAD that does not require that input data is represented in a fixed-dimensional feature space** (Section 3.3.5).

  Analogously to a conformal predictor, the main design parameter in CAD is the NCM. Previously proposed NCMs require that examples are represented as data points in a feature space with fixed dimensions. This may be a problem in some applications, such as trajectory anomaly detection, where input examples are represented as sequences or sets of data points of variable length or size, respectively. Hence, we propose the *Similarity-based Nearest Neighbour Non-Conformity Measure* (SNN-NCM) which only requires that a dissimilarity measure between pairs of examples is specified. Based on SNN-NCM, we propose SNN-CAD which is an online learning and anomaly detection algorithm that scales linearly with the size of training data.

Algorithms for Anomaly Detection in Trajectory Data

- **Proposal of algorithms for learning and sequential anomaly detection in incomplete trajectories** (Section 4.3 and 4.4.1).

  Four different algorithms for learning and sequential anomaly detection in trajectory data are proposed in this thesis. These can be categorised according to the underlying learning and anomaly detection algorithm and the type of feature model adopted. The first and the second of the proposed algorithms, known as *cell-based GMM* and *cell-based KDE*, are founded on statistical modelling of local trajectory *point features*, such as current position-velocity vector, using *Gaussian Mixture Models* (GMM) and *Kernel Density Estimation* (KDE), respectively (Section 4.3.1). The main novelties of these algorithms are two-fold. First, a grid-based approach to suppress model complexity is introduced, where a separate model is estimated for each cell of the grid based on the local training data. Second, a novel approach to point-based statistical anomaly detection is proposed that involves the combination of a two separate detectors based on to the position probability density function (PDF) and position-conditional velocity vector PDF, respectively. The third algorithm, known as *Single Point Trajectory Conformal Anomaly Detector* (SPT-CAD), is based on CAD and a point-based NCM that considers momentary position and velocity vector of trajectories (Section 4.3.2). The fourth algorithm is an adoption of SNN-CAD where directed *Hausdorff distance* (HD) is proposed as a parameter-free dissimilarity measure for trajectories (Section 4.4.1). The main novelty of SPT-CAD and SNN-CAD based on HD is that learning and anomaly detection is based on CAD.

- **Qualitative results for anomaly detection in a large set of real vessel tracks** (Section 5.3).

  Experiments are carried out where the validity of the cell-based GMM algorithm and the point-based feature model are demonstrated on a large real world data set. These experiments show the type of anomalous vessel behaviour that can be detected by the proposed algorithm.

- **Quantitative results from comparative evaluation of proposed algorithms for sequential anomaly detection in vessel trajectory data** (Section 5.4).

  A number of different algorithms for anomaly detection in vessel trajectory data have previously been published. However, there seems to be no published results regarding their relative performance. In this thesis, we investigate the relative performance of cell-based GMM and KDE and SPT-CAD for sequential anomaly detection in vessel trajectory data. Experiments are carried out using real vessel trajectories assumed to be normal and simulated trajectories considered anomalous. Results are related

to learning performance and anomaly detection delay of the proposed algorithms.

- **Results from empirical investigations of fundamental properties related to learning and anomaly detection performance of SNN-CAD based on HD** (Section 5.5 and 5.6).

  These investigations include reproduced experiments on a non-public data set of vessel trajectories (Section 5.4.7) and two public data sets of simulated (Section 5.5) and real video trajectories (Section 5.6), respectively. Results related to anomaly detection accuracy are compared to those previously published for other algorithms. Moreover, we demonstrate the ability of SNN-CAD to detect labelled anomalies in incomplete trajectories, and that sensitivity to true anomalies gradually improves during online learning.

Evaluation of Algorithms for Anomaly Detection in Trajectory Data

- **Proposal of *normalcy modelling performance measure* for measuring learning accuracy of statistical models for anomaly detection** (Section 5.4.4).

  Previous work on trajectory anomaly detection is focused on evaluating classification accuracy on a test set of trajectories labelled normal and anomalous. Yet, acquiring a representative set of labelled anomalies is problematic, since anomalies occur (very) rarely and may appear very different from each other. However, it may be argued that obtaining an accurate normalcy model is a prerequisite for good accuracy of any anomaly detector. To complement classification accuracy on test data, we therefore introduce normalcy modelling performance which, in contrast to classification accuracy, only requires data labelled as normal.

- **Proposal of *detection delay* as a performance measure in the domain of trajectory anomaly detection** (Section 5.1).

  Previous work on anomaly detection in trajectories is concerned with evaluating classification accuracy on (complete) trajectories. Yet, in case of sequential anomaly detection, we are also interested in minimising the time, i.e., the number of data points, required for accurately classifying incomplete trajectories. Detection delay, which is a well-known performance measure within the domain of *change-detection* (Ho and Wechsler, 2010), is therefore introduced for evaluating sequential trajectory anomaly detectors.

Summary of Contributions

To summarise, the main contributions of this thesis involve the proposal and evaluation of a number of algorithms appropriate for sequential anomaly detection in trajectory data. Two of these algorithms, SPT-CAD and SNN-CAD, are based on CAD which is a novel algorithm for online learning and anomaly detection proposed in this thesis. CAD is founded on the theory of CP and a key property that follows from this is that the false alarm rate is well-calibrated. The only design parameter in SNN-CAD is the dissimilarity measure; we propose the use of directed and undirected HD, which are both parameter-free dissimilarity measures, for anomaly detection in incomplete and complete trajectories, respectively.

All the proposed algorithms are evaluated on one or more real world data sets, including different sets of vessel trajectories and a set of video trajectories. A number of relevant performance measures are identified and discussed, of which two are novel in the context of trajectory anomaly detection. Qualitative results indicate the type of anomalous behaviour that can be detected by the algorithms. Quantitative results are related to learning performance and anomaly detection delay. In case of SNN-CAD, experiments previously published by other authors are reproduced and classification performance results compared to those previously reported for other algorithms.

1.4 Publications

The following publication list provides a short summary of the author's publications, and a description of how these contribute to the thesis. The publications are divided into those of high relevance and those of less relevance for the thesis.

Publications of High Relevance for the Thesis

1. **Laxhammar, R.** (2008) Anomaly detection for sea surveillance, *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

   This paper introduces cell-based GMM based on Expectation-Maximisation for sequential anomaly detection in vessel tracks. A point-based feature model based on momentary vessel position and velocity vector is proposed. The validity of the proposed algorithm and feature model is empirically investigated using a large data set of real vessel tracks that are unlabelled. Qualitative results demonstrate the type of anomalous vessel behaviour that can be detected by the proposed algorithm. This paper contributes to Objective 2, 3 and 4.

2. **Laxhammar, R.**, Falkman, G. and Sviestins, E. (2009) Anomaly Detection in Sea Traffic - a Comparison of the Gaussian Mixture Model and the Kernel Density Estimator, *Proceedings of the 12th International Conference on Information Fusion*, Seattle, USA, July 2009.

   The aim of this paper is to investigate the relative performance of cell-based GMM vs. cell-based KDE for sequential anomaly detection in vessel trajectories. To this end, two performance measures, which are novel in the context of trajectory anomaly detection, are introduced. The first, known as normalcy modelling performance, aims to measure the accuracy of the estimated PDF for normal data when the true PDF is unknown but normal sample data is available. The second performance measure, known as detection delay, aims to measure the sensitivity and reactivity of a sequential anomaly detector. The normalcy modelling performance of cell-based GMM and KDE is evaluated using a large data set of normal vessel trajectories extracted from an AIS database of recorded vessel traffic. Quantitative results from this experiment are complemented by qualitative results visualising differences between the PDFs estimated by GMM/KDE. Detection delay is evaluated on a set of simulated anomalous trajectories, where detector thresholds are tuned to generate a low rate of false alarms on a subset of the normal trajectories. This paper contributes to Objective 4, 5 and 6.

3. **Laxhammar, R.** and Falkman, G. (2010) Conformal Prediction for Distribution-Independent Anomaly Detection in Streaming Vessel Data, *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques* (ACM), Washington D.C., USA, July 2010.

   Conformal prediction (CP) is a recent machine learning theory proposed for supervised learning and prediction with valid confidence (Vovk et al., 2005). Given a specified significance level $\epsilon \in (0,1)$, a conformal predictor outputs a prediction set that includes the true label or value with probability at least $1-\epsilon$. In this paper, we present a novel application of CP for multi-class anomaly detection. The key idea of interpreting the empty and erroneous prediction sets as anomalies is discussed. Theoretical properties of an anomaly detector based on a conformal predictor are identified, including its distribution independence and application independent anomaly threshold ϵ; the expected rate of false alarms is bounded by ϵ, under the assumption that training data and new normal data are IID. The main design parameter in CP is the NCM. We adapt a previously proposed NCM based on distance to k-nearest neighbours, making it more suitable for multi-class anomaly detection applications. As an application, we consider anomaly detection in vessel trajectories, where vessel class is predicted based on the current position-velocity vector. If the prediction set does not include the (later) observed class (reported by

the vessel itself), the vessel is classified as anomalous. Experiments are performed on a subset of the normal vessel trajectories used in paper 2 above (Laxhammar et al., 2009), and a set of simulated anomalous trajectories. Results include detection delay on the anomalous trajectories for the proposed conformal predictor and the previously proposed cell-based GMM and KDE algorithms. This paper contributes to Objective 1, 3, 4 and 6.

4. **Laxhammar, R.** and Falkman, G. (2011) Sequential Conformal Anomaly Detection in Trajectories based on Hausdorff Distance, *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, July 2011.

   In this paper, we further refine our previous work on CP and anomaly detection in paper 3 above (Laxhammar and Falkman, 2010). Based on the concept of *smoothed p-values* from CP, we formalise CAD which is a general algorithm for anomaly detection. One of the main theoretical properties of CAD is that the false alarm rate is *well-calibrated*; if the training set and new normal data are IID, the rate of normal examples erroneously classified as anomalous will be close to the specified anomaly threshold ϵ. Analogously to a conformal predictor, the NCM is a central parameter in CAD. We propose SNN-CAD, which is based on a new NCM that, in contrast to previously proposed NCM, allows input data to be represented as sets or sequences of different sizes or lengths, respectively. The only design parameter in SNN-CAD is the specified dissimilarity measure S. We propose two parameter-free dissimilarity measures based on HD for comparing multi-dimensional trajectories of arbitrary lengths. One of these measures is designed for sequential anomaly detection in incomplete trajectories. One aim of SNN-CAD and the proposed trajectory dissimilarity measures is to detect anomalous trajectories with high accuracy without having to optimise any particular parameters. To this end, we reproduce two previously published experiments on two public data sets, and compare anomaly detection accuracy for SNN-CAD and previously published algorithms. There seems to be no results published for online learning and sequential anomaly detection on public trajectory data sets. Therefore, we carry out new experiments on one of the public data sets, investigating detection delay and how sensitivity to true anomalies increases as more training data is accumulated (online learning). This paper contributes to Objective 1, 2, 3, 4 and 6.

Publications of Less Relevance for the Thesis

5. Brax, C., **Laxhammar, R.** and Niklasson, L. (2008) Approaches for detecting behavioural anomalies in public areas using video surveillance data, *Proceedings of SPIE Electro-Optical and Infrared Systems: Technology and Applications V*, Cardiff, Wales, September 2008.

   In this paper, two different algorithms are evaluated for learning and anomaly detection in labelled trajectories, extracted from real video data. One of evaluated algorithms is an extended version of the cell-based GMM algorithm, which was originally proposed in paper 1 (Laxhammar, 2008). The extension is two-fold: Firstly, the point-based feature model is extended with a new feature corresponding to the accumulated time that the object has remained in the video frame. Secondly, a hierarchical grid at multiple spatial scales is introduced. The second algorithm adopts a histogram-based approach to anomaly detection and was originally proposed by Brax et al. (2008). Results show that both of the proposed algorithms can detect labelled anomalies while maintaining a low false alarm rate. The main contribution of the author is the development, implementation and evaluation of the extended cell-based GMM algorithm. The paper contributes mainly to Objective 4 and to less extent Objective 3.

6. Brax, C., Niklasson, L. and **Laxhammar, R.** (2009) An ensemble approach for increased anomaly detection performance in video surveillance data, *Proceedings of the 12th International Conference on Information Fusion, Seattle*, USA, July 2009.

   This paper extends previous work in paper 5 above by considering a more crowded scene that involves more complex behaviour. Similar to the previous paper, an updated version of the cell-based GMM (Laxhammar, 2008) and the histogram-based algorithm (Brax et al., 2008) are evaluated on another data set of labelled trajectories, extracted from recorded video data. For cell-based GMM, the extended feature model from paper 5 is further extended with an additional feature corresponding to the accumulated time that an object has remained stationary. In addition to evaluating the classification performance of each individual anomaly detector, the combination of the two detectors is also evaluated. Results show that a simple combination achieves better classification performance than any of the two detectors by themselves. Similar to paper 5, the main contribution of the author is the development, implementation and evaluation of the extended cell-based GMM algorithm. The paper contributes to mainly to Objective 4 and to less extent Objective 3.

Table 1.1: Research objectives, publications and thesis chapters.

Objectives	Publications	Chapters
Objective 1: Identify important and desirable theoretical properties of algorithms for anomaly detection in surveillance applications.	Paper 3 and 4	Chapter 3
Objective 2: Review and analyse previously proposed algorithms for anomaly detection in trajectory data.	Paper 1 and 4	Chapter 2 and 4
Objective 3: Propose algorithms that are well-suited for anomaly detection in trajectory data.	Paper 1 and 3–6	Chapter 3 and 4
Objective 4: Demonstrate feasibility and validity of proposed algorithms on real world data sets.	Paper 1–6	Chapter 5
Objective 5: Identify suitable performance measures for evaluating algorithms for anomaly detection in trajectory data.	Paper 2 and 4	Chapter 5
Objective 6: Evaluate proposed algorithms according to identified performance measures.	Paper 2, 3 and 4	Chapter 5

1.5 Thesis Outline

This thesis is organised as follows. After the introductory chapter, we present the background to the subjects of this thesis in Chapter 2. This consists mainly of a review of anomaly detection in general and anomaly detection in trajectory data in particular. We will also briefly review Conformal prediction and Hausdorff distance.

In Chapter 3, we theoretically investigate algorithms for anomaly detection. We start off by discussing various issues and limitations of previously proposed algorithms. This is followed by a discussion of a novel application of CP for multi-class anomaly detection. The remaining part of the chapter is dedicated to the CAD, which is a general algorithm for anomaly detection proposed in this thesis. We identify and discuss key properties of CAD. We also propose SNN-CAD, which is appropriate for anomaly detection in data represented as sets or sequences of varying size or length, such as trajectories.

In Chapter 4, we investigate algorithms for sequential anomaly detection in trajectory data. Two types of algorithms are considered: *point-based* algorithms that consider representations of single trajectory points, and *trajectory-based* algorithms that consider representations of complete trajectories or trajectory segments. Two types of algorithms for point-based sequential anomaly detection are proposed and discussed. The first is cell-based statistical modelling of point features using GMM or KDE. Traditional and novel point feature models, based on the position and velocity vector, are considered. The second point-based anomaly detector proposed is SPT-CAD. For trajectory-based sequential anomaly detection, SNN-CAD based on HD is proposed.

In Chapter 5, we empirically investigate the algorithms proposed in this thesis. We start by introducing and discussing the performance measures used in the experiments. A number of experiments are then carried out, organised according to the different data sets used. In the first experiment, we investigate cell-based GMM for anomaly detection in a relatively large data set of unlabelled vessel tracks. This is followed by a series of experiments on other data sets of labelled vessel trajectories, where relative performance of all the anomaly detection algorithms proposed in this thesis is evaluated. In the final part of this chapter, we reproduce two experiments previously published by other authors on two public data sets of synthetic and real video trajectories, respectively. Here, classification performance for SNN-CAD is compared to previously published algorithms. One of the data sets is also used for investigating some fundamental properties of SNN-CAD related to learning and anomaly detection.

Finally, in Chapter 6, the main conclusions that can be drawn from the thesis are discussed. This includes the main scientific contributions and possible directions for future work.

Chapter 2
Background

This chapter gives a background to the subject of the thesis and introduce basic concepts and theory that are needed. The first part, Section 2.1, introduces the problem of anomaly detection and gives a survey over different aspects of it and various algorithms for solving it. This is followed by a presentation of previous work related to anomaly detection in trajectory data (Section 2.2), which is the central topic of the thesis. In Section 2.3, we introduce the theory of Conformal prediction which underpins the Conformal Anomaly Detector, one of the the main contributions of the thesis. The last section introduces the Hausdorff distance which serves as basis for the proposed trajectory dissimilarity measures, another contribution of the thesis.

2.1 Anomaly Detection

Anomaly detection has been identified as an important technique for detecting critical events in a wide range of data rich domains where a majority of the data is considered "normal" and uninteresting (Latecki et al., 2007; Chandola et al., 2009). Yet, it is a rather fuzzy concept and domain experts may have different notions of what constitutes an *anomaly* (cf. Roy (2008)). Common synonyms to anomaly include *outlier, novelty, rare*, *abnormal*, *deviating*, *unexpected, suspicious*, *interesting* etc. In the academic world, more or less similar definitions of anomaly detection and the closely related concepts *outlier detection* and *novelty detection* have been proposed by different authors with various backgrounds and application areas. However, the methods and algorithms used in practise are often the same (Chandola et al., 2009).

In the statistical community, the concepts *outlier* and *outlier detection* have been known for quite a long time. Barnett and Lewis (1994) defined an outlier in a data set to be "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data". A similar definition of an outlier was given by Hawkins (1980):

> [An outlier is] an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism.

This mechanism is usually assumed to follow a stationary probability distribution. Hence, outlier detection essentially involves determining whether or not a particular observation has been generated by the same distribution as the rest of the observations. Traditionally, outlier detection in the statistical community has been used for cleaning data sets by removing *noise* or *contaminants* before fitting statistical models; outliers are considered noise and are removed in order to improve the quality of the statistical models.

In the data mining community, "anomalies are patterns in data that do not conform to a well defined notion of normal behaviour" (Chandola et al., 2009). Often, the notion of normal behaviour is captured by a *normalcy model*, which is induced from training data. According to Portnoy et al. (2001), "anomaly detection approaches build models of normal data and then attempts to detect deviations from the normal model in observed data". In contrast to traditional statistical applications, data mining applications are usually interested in the anomalous observations *per se*, since they may correspond to interesting and important events. Traditional applications of anomaly detection in data mining include *fraud detection* in commercial domains (Chandola et al., 2009), *intrusion detection* in network security (Portnoy et al., 2001) and *fault detection* in industrial domains (Chandola et al., 2009). These applications all have in common that patterns of the interesting behaviour is difficult, if not impossible, to explicitly define a priori because of limited knowledge and lack of data.

According to Ekman and Holst (2008), "anomaly detection says nothing about the detection approach and it actually says nothing about what to detect". Indeed, anomaly detection, as it is usually defined, refers to a process that aims to *detect something*; yet it says nothing in particular about *what* to detect. This means that the *interpretation* and *impact* of an anomaly is undefined within the scope of the anomaly detector. One could argue that the definition of an anomaly is always relative a specific model or data set and therefore it is a subjective concept rather than an objective truth; something that appears to be deviating relative a statistical model, or strange to a human with limited domain knowledge, may be fully understandable and predictable by, e.g., some other model or human domain expert. Therefore, great care should be taken when selecting a suitable and representative domain model or data set for anomaly detection applications.

2.1.1 General Aspects of Anomaly Detection

In a recent survey by Chandola et al. (2009), a number of general aspects of the anomaly detection problem is discussed: the nature of input data, the availability of labelled data, the type of the anomalies to be detected and the nature

of the output. This section reviews these and other aspects related to anomaly detection, such as offline vs. online learning.

Nature of Input

Similar to other data mining algorithms, most anomaly detection algorithms assume that the basic input is in the form of a *data point*, also referred to as *data instance*, *feature vector, observation, pattern, example*, *object* etc. The data point can be univariate or multivariate, but has usually a fixed number of *feature*s, also referred to as *attributes* or *variable*s. These can be a mix of binary, categorical or continuous values. However, some methods do not require explicit data points as input; instead, pairwise distances or similarities between data points are provided in the form of a similarity or distance matrix (Chandola et al., 2009). Another aspect of the input is the relationship between different data points, which can be of spatial and temporal nature. Trajectory data is an example of time-series where data points are temporally ordered. Yet, most anomaly detection techniques explicitly or implicitly assume that there is no relationship between different data points, i.e., that they are independent of each other (Chandola et al., 2009).

Most applications of anomaly detection involve a *feature extraction* processes; this corresponds to preprocessing the raw input data and extracting relevant features. In the context of moving object surveillance, such features may be current speed and location of an object, its size and previously visited locations. The choice of an appropriate *feature model* is critical in anomaly detection applications, since it essentially determines the character of detected anomalies. If, for example, we only consider the position feature of an object, it will be hard, if not impossible, to detect anomalies related to low or high speed of the object, assuming that speed is more or less independent of position. If inappropriate features are selected, the resulting anomalies may be of little or no interest. Thus, features should be selected carefully based on available domain knowledge of how interesting anomalies manifest themselves.

Types of Anomalies

Considering the anomalies that are to be detected, Chandola et al. (2009) categorise them as *point anomalies*, *contextual anomalies* and *collective anomalies*. Point anomalies correspond to individual data points that are anomalous relative *all* other data points; this type of anomalies is captured by, e.g., the definition of an outlier given by Hawkins (1980). Point anomalies are the focus of most research within anomaly detection algorithms (Chandola et al., 2009). Contextual anomalies, also known as conditional anomalies, are data points that are considered anomalous in a particular context. To formalise the notion of context, each data point is defined by *contextual attributes* and *behaviour attributes* (Chandola et al., 2009). If the behaviour attributes of a data point

are anomalous relative the behaviour attributes of the subset of data points having the same or similar contextual attributes, the corresponding data point is considered a contextual anomaly. Examples of contextual attributes could be time of day, season and geographical location. Lastly, collective anomalies consist of a set or sequence of *related* data points that are anomalous relative the rest of the data points. In this case, the individual data points may not be anomalous by themselves; it is the aggregation of the data points that is anomalous. Examples of collective anomalies can found in sequence data, graph data and spatial data (Chandola et al., 2009).

Availability of Data Labels

In some anomaly detection applications, historical data may be annotated by a *label* telling whether a particular data point is considered normal or anomalous. This annotation is typically based on human expert knowledge regarding normalcy and what constitutes an anomaly in the current domain. Since annotation is often done manually, the available amount of labelled data is usually very limited. In particular, labelled anomalies are usually hard to acquire due to the fact that such data points are rare and that anomalies may be dynamic in nature, i.e., new types of anomalies may arise for which there is no labelled training data (Chandola et al., 2009).

Based on the extent to which labels are available, anomaly detection algorithms can be categorised as *supervised*, *semi-supervised* or *unsupervised* (Chandola et al., 2009). Supervised algorithms assume that the training set contains labelled data points of both classes, i.e., normal and anomalous. They typically learn a predictive model for classifying new unlabelled data points as either normal or anomalous (Chandola et al., 2009; Latecki et al., 2007). However, they are considered out of the scope in most anomaly detection applications, since availability of labelled anomalies is very limited (Latecki et al., 2007). Indeed, most definitions of anomaly and outlier detection, including those presented earlier in this chapter, suggest that no labelled anomalies are required for normalcy modelling.

In contrast, semi-supervised techniques only assume that data points labelled as normal are available. They typically learn a normalcy model from a data set assumed to reflect normalcy. This model is then used for detecting anomalies in new data. Unsupervised techniques are even more flexible, since they learn a normal model from an unlabelled data set which may include anomalies. These techniques do, however, make the implicit assumption that normal data points are (far) more frequent than anomalous in the data set; if this is not the case, such algorithms may suffer from high false alarm rates (Chandola et al., 2009).

Online vs. Offline Learning

Learning in most anomaly detection algorithms is essentially *offline*; static model parameters are learnt from a batch of training data and then used repeatedly when classifying new data. In order to accurately model normalcy, a fairly large training set may be required, which is representative of all possible normal behaviour. But such a data set might not be available from the outset. Moreover, "in many domains normal behaviour keeps evolving and a current notion of normal behaviour might not be sufficiently representative in the future" (Chandola et al., 2009). In contrast, *online learning* may account for this by incrementally refining and updating model parameters based on new data points.

Output

There are generally two types of output from an anomaly detector; *scores* and *labels* (Chandola et al., 2009). Scoring techniques assign an anomaly or outlier score to each input data point, where the score value reflects the degree to which the corresponding data point is considered anomalous. Output is usually a list of anomalous data points that are sorted according to their anomaly score. Such a list may include the top-k anomalies, or a variable number of anomalies having a score above a predefined threshold. Labelling techniques, on the other hand, output a label for each input data point, usually *normal* or *anomalous*. Such techniques may also output the corresponding anomaly score, confidence or probability associated with the label. More details regarding the output of different algorithms will be discussed in Section 2.1.2 and 2.1.3 below.

Algorithms for Anomaly Detection

A number of surveys attempting to structure different algorithms for anomaly detection have been published during the last years (e.g., Chandola et al. (2009); Patcha and Park (2007)). Chandola et al. (2009) categorise algorithms as belonging to one or more of the following classes: *classification based techniques*, *parametric or non-parametric statistical techniques*, *nearest neighbour based techniques*, *clustering based techniques*, *spectral techniques* and *information theoretic techniques*. In their survey, various advantages and disadvantages of algorithms from each category are discussed at length.

In this thesis background, we will focus on statistical techniques, since the anomaly detection algorithms we propose and evaluate fall into this category. But we will also present the general principles of classification, nearest neighbour and clustering based techniques, since they are commonly applied algorithms for anomaly detection in trajectory data.

2.1.2 Statistical Anomaly Detection

Statistical methods for anomaly detection are based on the assumption that "normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model" (Chandola et al., 2009). It is usually assumed that normal data points constitute independent and identically distributed (IID) samples from a stationary probability distribution, P, which can be estimated from sample data, D. Thus, statistical methods are based on semi-supervised learning. Given a new data point, z, the goal is to determine whether it can be assumed to have been generated by P or not, i.e., if it is anomalous or not relative the sample data D. Hence, there are two practical problems: how to estimate P based on D, and how to decide whether z can be assumed to be a random sample from P.

Parametric Methods and GMM

Statistical methods for estimating probability distributions can broadly be categorised as either *parametric-* or *non-parametric* models (Markou and Singh, 2003a). Starting with the parametric models, they assume that the underlying distribution belongs to a parameterised family of distributions, i.e., $P_\theta : \theta \in \Theta$, where the parameters θ belong to a parameter space Θ and can be estimated from available sample data D. A common and simple parameterised model in anomaly detection applications is the Gaussian distribution (Chandola et al., 2009; Markou and Singh, 2003a). Another example is the Poisson distribution (Holst et al., 2006). More complex parameterised models in anomaly detection applications include various graphical models, such as Bayesian networks(Johansson and Falkman, 2007) and Hidden Markov Models (HMM) (Urban et al., 2010), and mixture models, such as as univariate and multivariate Gaussian Mixture Models (GMM) (Laxhammar, 2008; Ekman and Holst, 2008) and mixtures of other parameterised distributions, such as the Poisson and Gamma distributions (Ekman and Holst, 2008).

GMM is a common model for approximating continuous multi-modal distributions when knowledge regarding the structure is limited; it has been used in numerous anomaly detection applications (Chandola et al., 2009). A GMM consist of C multivariate Gaussian densities known as mixture components. Each Gaussian component $c_i, i = 1, \ldots, C$, has it own parameter values $\theta_i = (\mu_i, \Sigma_i)$ and weight w_i, where μ_i is the mean value vector, Σ_i is the covariance matrix and w_i is a non-negative normalised mixing weight where all weights sum to one. The total set of parameters for the GMM is denoted $\theta = \{\theta_1, \ldots, \theta_C, w_1, \ldots, w_C\}$. The probability density function for the multivariate GMM is given by:

$$p(x) = \sum_{i=1}^{C} w_i \frac{1}{(2\pi)^{d/2} \sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(x - \mu_i)^\top \Sigma_i^{-1} (x - \mu_i)\right). \qquad (2.1)$$

A common and relatively simple way to estimate the parameters θ of a distribution P_θ based on a data sample D is to use a *Maximum Likelihood* (ML) *estimator*. The Expectation-Maximisation (EM) algorithm (Dempster et al., 1977) is a widely used ML estimator when D is incomplete and data points may have missing values, also known as latent variables. One example of missing values is when it is unknown which of the components of a mixture model that generated a data point. Typically, the EM algorithm starts by randomising initial values for the parameters and then incrementally estimate the values $\hat{\theta}$ that yield maximum likelihood for the sample data D:

$$\hat{\theta} = \underset{\theta \in \Theta}{argmax} \left(D|\theta \right). \tag{2.2}$$

More specifically, the algorithm consist of two steps, *Expectation* and *Maximisation*, that are repeated until a certain end condition, usually a convergence condition, is fulfilled. In case of a GMM with a predefined number of components C, the Expectation step involves updating the posterior probabilities $p\left(c_i \left| x_j \right.\right)$ for each data point $x_j \in D, j = 1, \ldots, n$, belonging to each component $c_i, i = 1, \ldots, C$, according to Bayes' rule (Verbeek, 2003):

$$p\left(c_i \left| x_j \right.\right) = \frac{p\left(x_j; c_i\right) w_i}{\sum\limits_{q=1,\ldots,C} p\left(x_j; c_q\right) w_q}, \tag{2.3}$$

where

$$p\left(x_j; c_q\right) = \frac{1}{\left(2\pi\right)^{d/2} \sqrt{\left|\Sigma_q\right|}} \exp\left(-\frac{1}{2} \left(x_j - \mu_q\right)^\top \Sigma_q^{-1} \left(x_j - \mu_q\right) \right) \tag{2.4}$$

corresponds to the qth component distribution. This expectation of point to component correspondence is then used in the Maximisation step where the parameters of each component are updated based on ML estimation. The Maximisation step involves adjusting the parameters of each component in such a way that the component better fits the data points, taking the updated posterior probabilities into account. More specifically, component parameters are updated according to Equation 2.5 to 2.7 (Verbeek, 2003):

$$w_i = \frac{1}{n} \sum_{j=1}^{n} p\left(c_i \left| x_j \right.\right), \tag{2.5}$$

$$\mu_i = \frac{1}{n w_i} \sum_{j=1}^{n} p\left(c_i \left| x_j \right.\right) x_j, \tag{2.6}$$

$$\Sigma_i = \frac{1}{n w_i} \sum_{j=1}^{n} p\left(c_i \left| x_j \right.\right) \left(x_j - \mu_i\right) \left(x_j - \mu_i\right)^\top. \tag{2.7}$$

The updated model is then used for calculating new posterior probabilities in the Expectation step, and so on.

The popularity of the standard EM algorithm is probably due to its relatively simplicity and fast convergence. But it has some disadvantages. To start with, it is not guaranteed to converge to a global optimum; the algorithm is more or less sensitive to the parameter initialisation and may converge to different ML estimates depending on the start values (Verbeek, 2003). The standard procedure to overcome this initialisation dependence is to start the EM algorithm from several random initialisations and retain the best obtained result (Verbeek, 2003). An extension of the standard EM algorithm calculates the *maximum a posteriori* (MAP) *estimate* based on a prior distribution, $p(\theta)$, on the parameters, thus incorporating prior knowledge and making the algorithm less sensitive to initialisation and noisy data. Another issue is how to determine the optimal number of components, a problem which is not solved by the standard EM algorithm. Verbeek proposed an efficient and greedy version of the EM algorithm that determines the optimal number of components of a GMM and avoids the need for multiple runs with random parameter initialisation (Verbeek et al., 2003).

ML and MAP estimators do not include any uncertainty in the parameter estimates; they simply calculate the most likely parameter values for a given data set, regardless of the size of the data set. Hence there is no information on how confident we can be in the estimates. In the case of a small sample, the estimates are susceptible to random variations in the data; this is a bad property of an anomaly detector since it will give a lot of false alarms by focusing too much on the peculiarities of the data (Holst et al., 2006). Moreover, in many applications, including anomaly detection, we are not interested in the model's parameter values *per se*; rather, we are interested in getting an accurate and reliable estimate of the predictive data distribution, $p(x)$, based on the sample data D. In this case, an alternative to ML or MAP is a fully Bayesian parameter estimation, where the posterior distribution for the parameter values, $p(\theta|D)$, is estimated based on the prior distribution, $p(\theta)$, and available sample data, D, according to Bayes' theorem (Gelman et al., 2003):

$$p(\theta|D) = \frac{p(D|\theta)\,p(\theta)}{\int_\theta p(D|\theta)\,p(\theta)}. \tag{2.8}$$

The posterior distribution for the parameters can then be used for estimating the predictive distribution for normal data (Holst et al., 2006):

$$p(x|D) = \int_\theta p(x|\theta)\,p(\theta|D)\,p(\theta)\,. \tag{2.9}$$

Non-Parametric Methods and KDE

A general drawback of the parametric techniques is that they assume that a parametrised model exists and that it can be accurately estimated; this is doubt-

ful in many applications where the true structure of the probability distribution is unknown and may be highly complex (Markou and Singh, 2003a). One approach to mitigate this problem is to resort to non-parametric methods, such as histograms or Kernel Density Estimators (KDE), which essentially assume nothing about the structural form of the distribution; it is instead determined from the given data (Chandola et al., 2009; Markou and Singh, 2003a).

In histogram based methods, the feature space is assumed to be discrete and each possible value corresponds to a bin of the histogram. The histogram is constructed by simply counting the number of data points that fall within each of the bins. The unknown data distribution is then approximated by the discrete distribution of observed data points from the different bins, i.e., the relative frequency. A drawback of histogram methods is that continuous input data has to be discretised. It may not be obvious how the discretisation should be done and subtle, yet significant, information may be lost in the process. Determining the optimal size of the bins is a key challenge for maintaining low false alarm (false positive) rate and low false negative rate in anomaly detection applications (Chandola et al., 2009). Histogram methods for anomaly detection in multivariate data typically construct separate univariate histogram for each data feature (Chandola et al., 2009). Thus, they do not capture potential correlation between different features. However, there is at least one exception to this, in which correlation between different attributes are accounted for by considering a variant of a *multi-dimensional histogram* (Brax et al., 2008).

In contrast, KDE, also known as *Parzen window estimation,* allows for non-parametric estimation of a multivariate continuous distribution based on a *kernel function* (Chandola et al., 2009). The unknown distribution is approximated by placing a kernel function K on each and every data point x_i of the data sample:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{(h_i)^d}K\left(\frac{x - x_i}{h_i}\right), \tag{2.10}$$

where n is the size of the sample data, d is the number of dimensions of the data points and h_i is a kernel width parameter known as *bandwidth* (Latecki et al., 2007), *window width* (Ristic et al., 2008) or *smoothing parameter* (Markou and Singh, 2003a). The kernel function is usually a symmetric probability density function, being non-negative over its domain and should integrate to unity over the defined range (Markou and Singh, 2003a). A common kernel for anomaly detection applications is the multivariate Gaussian function with zero mean and covariance matrix Σ (Ristic et al., 2008; Latecki et al., 2007):

$$K(x) = \frac{1}{(2\pi)^{d/2}\sqrt{|\Sigma|}}\exp\left(-\frac{1}{2}x^{\top}\Sigma^{-1}x\right). \tag{2.11}$$

The covariance matrix Σ can be set to, e.g., the identity matrix (Latecki et al., 2007) or estimated from data as sample covariance (Ristic et al., 2008). The

kernel width can either be fixed, i.e., the same for all data points, or adaptive based on the local (Latecki et al., 2007) or global (Ristic et al., 2008) distribution of the data points. A formula for computing the optimal fixed width for the Gaussian kernel (under the assumption that the underlying PDF is Gaussian) was originally proposed by Silverman (1986). Yet, a fixed kernel width is inappropriate in applications where local data density varies (Latecki et al., 2007). In particular, a fixed kernel width is unable to deal satisfactorily with the tails of distributions where local density is low (Ristic et al., 2008). Therefore, kernel widths should be adaptive based on the local data density, i.e., broader windows in low density areas and narrower windows in high density areas. A two stage approach known as *adaptive KDE* for computing adaptive window widths from the fixed optimal width was proposed by Silverman (1986). Another approach where local window widths are based on distance to the *k*th nearest neighbour was presented by Latecki et al. (2007). There are, however, at least two general drawbacks of KDE methods: they are relatively sensitive to noise in data (Markou and Singh, 2003a) and they require a relatively large sample set for accurate density estimation.

Anomaly Detection in New Data

Assuming that P has been estimated based on a parametric or non-parametric method as discussed above, various anomaly tests for a new data point z have been proposed (Chandola et al., 2009). Some of these compute an anomaly score, which depends on the probability (discrete data) or probability density value (continuos data), and classify the data point as anomalous if the score is above a particular threshold. Often, the anomaly score is simply the inverse or negative of the data likelihood or its logarithm. Other approaches are based on statistical hypothesis testing where the null hypothesis, H_0, that z constitutes a random sample from the estimated distribution $p(x)$ is tested (Chandola et al., 2009). These tests are based on a specified *test statistic* and *significance level*, which corresponds to the confidence in classifying a data point as anomalous.

For Gaussian models, such as the simple Gaussian distribution or mixtures of Gaussian distributions, a large number of anomaly measures and test statistics have been proposed for classifying new data points as normal or anomalous. A simple technique is to calculate the number of standard deviations (σ) from the corresponding mean μ and compare to a threshold (Markou and Singh, 2003a). For example, data points located 3σ or further away more from μ may be classified as anomalous, where the region $\mu \pm 3\sigma$ contains 99.7% of all data points from the corresponding distribution (Chandola et al., 2009). An equivalent and relatively simple anomaly measure for univariate or multivariate data is the *box plot rule,* which defines a region where 99.3% of the normal data points lie (Chandola et al., 2009). A well known outlier test is *Grubb's test*, where the z-score for a univariate data point is calculated and compared to an adaptive threshold based on data size and the value of a *t*-distribution taken

at a specified significance level (Chandola et al., 2009). Multiple variants of Grubb's test have been proposed for multivariate data (Chandola et al., 2009). A number of other test statistics for anomaly detection have also been proposed for univariate and multivariate data, including *Student's t*-test, *Hotelling's* t^2-test and χ^2-tests. All these test statistics have in common that they assume that the underlying data distribution is Gaussian.

Ekman and Holst (2008) proposed *principal anomaly* as "the true anomaly measure". Assuming that X is a random variable from the probability distribution $p(x)$, the principal anomaly of a data point z with respect to $p(x)$ is defined as $Pr\{p(X) \geq p(z)\}$, i.e., the probability that a random sample X has a density equal to or larger than the density of z (Ekman and Holst, 2008). This measure is attractive from a theoretical perspective, since it is normalised and does not assume that $p(x)$ is Gaussian. However, it is typically not easy to estimate the principal anomaly for multivariate distributions and distributions other than the Gaussian (Ekman and Holst, 2008). Therefore, Ekman and Holst (2008) proposed *deviation* as an alternative anomaly measure, which is defined as:

$$\frac{E[\log p(X)] - \log p(z)}{S[\log p(X)]}, \tag{2.12}$$

where E and S correspond to the expectation (mean) and standard deviation, respectively. They argue that this measure is strongly related to principal anomaly and show how it can be easily calculated for multivariate distributions with independent variables (Ekman and Holst, 2008).

In histograms based methods, there are generally two approaches to anomaly detection. The first is to simply check if new data points fall in any one of the bins of the histogram. If it does, the data points is classified as normal; otherwise it is anomalous (Chandola et al., 2009). The second approach assigns an anomaly score to each data point based on the frequency of the corresponding bin (Chandola et al., 2009). If frequency is below a specified threshold, the data point is classified as anomalous.

For other non-Gaussian distributions, such as Bayesian networks, HMM, KDE etc., anomaly detection is usually done by checking whether the likelihood $p(z)$, or the corresponding logarithmic likelihood, $\log(p(z))$, for the new data point is below an application specific anomaly threshold. A low likelihood indicates an improbable data point and, hence, evidence against the hypothesis that it is normal. More generally, a higher anomaly likelihood threshold increases the probability of detecting true anomalies but also the probability of false alarms.

2.1.3 Other Anomaly Detection Algorithms

Classification Based Methods

Classification based methods essentially assume that a decision boundary in feature space, which separates normal and anomalous data points, can be learnt from a training data set. Similar to statistical methods, classification based methods are usually based on semi-supervised learning. Depending on the number of different normal labels, classification based methods can be grouped into two broad categories: *multi-class* and *one-class* anomaly detection techniques (Chandola et al., 2009). Multi-class techniques assume that training data includes labels from multiple normal classes. Typically, new data points are either classified as belonging to one of the normal classes, or classified as anomalous if it they do not fit any of the normal classes. One-class techniques assume that all data points in the training set have the same label; they typically learn a discriminative boundary around the training data which is used to check if new data points are anomalous or not.

Most of the classification based methods for anomaly detection are based on neural networks, Support Vector Machines (SVM) or rule-based methods (Chandola et al., 2009). Considering neural networks, Multi-Layer Perceptrons (MLP) are the best known and most widely used (Markou and Singh, 2003b). Other neural networks used for anomaly detection include Auto-associative networks, Adaptive resonance theory, Radial basis functions and Hopfield networks (Chandola et al., 2009). Neural networks are powerful classifiers, since they can approximate arbitrary complex decision boundaries in feature space. However, they are designed for *discriminating* between multiple known classes rather than *detecting* new classes (Markou and Singh, 2003b). Since they do not generate closed class boundaries, adapting them to anomaly (novelty) detection and ensuring that the generalisation property of the network does not interfere with its anomaly detection ability are fairly challenging tasks (Markou and Singh, 2003b). In other words, there is typically a risk of overfitting to the training data.

A number of different SVM based methods have been applied to anomaly detection in the one-class setting (Chandola et al., 2009). Similar to neural networks, SVM methods are capable of learning arbitrary complex regions in input feature space. However, they do not have the same problem of overfitting to training data, since they estimate optimal wide-margin hyperplanes in a high-dimensional feature space. A key parameter of any SVM is the *kernel function*, e.g., a polynomial or Gaussian function, which maps input data to a high-dimensional feature space where data is perfectly, or close to, linearly separable. Schölkopf et al. (2000) proposed a one-class SVM for novelty detection based on quantile estimation of arbitrary distributions. The idea is to learn the boundaries in input data feature space for a specified quantile of the underlying distribution, based on available training data. For example, estimating the

boundaries for the first percentile results in a region containing approximately 99% of randomly generated data points from the underlying distribution. The specified quantile corresponds to the anomaly threshold; if a new data point falls outside of the region, it is considered anomalous. The region is found by estimating the hyperplane in the high-dimensional feature space that maximises the distance to the origin where only a small fraction of the training data points fall between the hyperplane and the origin (Schölkopf et al., 2000).

Various methods that learn rules from normal data based on, e.g., decision trees have been proposed for anomaly detection (Chandola et al., 2009). Some of these methods associate a confidence score to each rule, which is proportional to the ratio of the number of training instances correctly classified by the rule and the total number of training data points covered by the rule (Chandola et al., 2009). For new data points, the rule that best captures the data points is searched for, and the inverse of the corresponding rule confidence is returned as the anomaly score. If no rule can be found that captures the new data points, it is classified as anomalous.

Nearest-neighbour Methods

Nearest-neighbour methods for anomaly detection are based on the assumption that "normal data [points] occur in dense neighbourhoods, while anomalies occur far from their closest neighbours" (Chandola et al., 2009). This assumption suggest that a distance or similarity measure between data points is needed. For continuous features, the Euclidean distance (ED) is a popular distance measure, but other measures have also been proposed (Chandola et al., 2009). For categorical attributes, simple matching coefficient is often used, or other more complex measures (Chandola et al., 2009). Generally, distance measures used in nearest neighbour methods are required to be *positive-definite* and *symmetric*; but they are usually not required to satisfy the *triangle inequality* and, hence, not required to be strictly *metric* (Chandola et al., 2009).

A number of different algorithms for calculating anomaly scores based on the nearest neighbour principle have been proposed. A common algorithm is based on the following principle: "The anomaly score of a data [point] is defined as its distance to its k^{th} nearest neighbour in a given data set" (Chandola et al., 2009). Other variants of this principle calculate the anomaly score as the *sum* of the distances to the k nearest neighbours (Eskin et al., 2002), or as the inverse of the number of nearest neighbours that are no more than d distance apart from the corresponding data point (Chandola et al., 2009). The latter algorithm has the flavour of kernel density estimation (Section 2.1.2) where the kernel function corresponds to counting the number of neighbouring data points within the hypersphere of radius d. However, analogously to statistical methods, the nearest neighbour techniques discussed above may be suboptimal if the data has regions of varying densities (Chandola et al., 2009). To handle this, techniques have been proposed for computing the local density

of data points, i.e., the density of data points relative their closest neighbours. One such technique is *Local Outlier Factor* (LOF), in which the anomaly score for a particular data point is equal to the ratio of the average local density of the k nearest neighbours, and the local density of the data point itself (Chandola et al., 2009). The local densities in LOF are inversely proportional to the radius of the minimum hypersphere that contains the k nearest neighbours. Several variants and extension of the basic LOF algorithm have been proposed, e.g., for handling different data types and improving computational efficiency.

In most cases, a threshold is applied to the anomaly score during anomaly detection. Alternatively, the n data points with the largest anomaly scores may be returned (Chandola et al., 2009).

Clustering Methods

Clustering is essentially an unsupervised learning technique that groups similar data points into clusters. Chandola et al. (2009) categorise clustering based techniques for anomaly detection into three groups, depending on what assumptions are made regarding normal and anomalous data. The first group assumes that "normal data [points] belong to a cluster in the data, while anomalies do not belong to any cluster". These techniques are based on an algorithm that clusters the data while not forcing all data points to belong to a cluster; data points that are not found to belong to any cluster are classified as anomalous. According to Chandola et al. (2009), these methods have the disadvantage that they are designed for finding clusters rather than anomalies.

The second group of clustering techniques are more focused on finding anomalies; they are based on the assumption that "normal data [points] lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid" (Chandola et al., 2009). Typically, these methods first cluster the data and then assign an anomaly score to each data point based on the distance to its nearest cluster centroid. Common clustering algorithms include *k-means clustering, Self-Organising Maps* (SOM) and Expectation-Maximisation (EM) (Chandola et al., 2009). In EM, clusters are represented as Gaussian components in a mixture model, i.e., similar to estimation of a GMM (Section 2.1.2). Techniques from the second group can also operate in a semi-supervised mode, where new test data is compared to a cluster model learnt from previous training data (Chandola et al., 2009).

If anomalies form clusters by themselves, algorithms based on the two principles presented above will obviously not be able to detect those anomalies. To address this issue, algorithms have been proposed that rely on the following assumption: "Normal data [points] belong to large and dense clusters, while anomalies either belong to small or sparse clusters" (Chandola et al., 2009). These algorithms assign an anomaly score, which reflects the size or density of the cluster to which the corresponding data points belongs. An example of such method is *Cluster-Based Local Outlier Factor* (Chandola et al., 2009).

2.2 Anomaly Detection in Trajectory Data

Research related to automated anomaly detection in trajectory data has attracted a lot of attention lately, much due to the increasing amounts of historical and real-time trajectory data; there is a clear trend towards more and more advanced sensor systems producing huge amounts of trajectory data from moving objects, such as people, vehicles, vessels and animals. For example, the use of video cameras for public surveillance has increased dramatically since the beginning of the 21th century, and a significant research effort has been done towards developing object tracking algorithms that produce trajectories from raw video data. In the maritime surveillance domain, regulations regarding the mandatory use of AIS transponders for vessels of 300 gross tonnage or greater have resulted in a significant increase of vessel tracking data, which complements existing radar tracking systems (Chang, 2004). In the domain of intelligent transportation, widespread use of GPS has facilitated tracking of land-based transports.

2.2.1 Representing Trajectory Data

In its raw form, a trajectory is often represented as a finite sequence $T = \big((\mathbf{x_1}, t_1), (\mathbf{x}_2, t_2), \ldots, (\mathbf{x}_m, t_m)\big)$. Each data point $\mathbf{x}_i$, sometimes referred to as a *flow vector* (Morris and Trivedi, 2008a), corresponds to a multi-dimensional feature vector of a moving object at time point t_i, where $t_i < t_{i+1}$ for $i = 1, \ldots, m-1$. The feature vectors typically correspond to processed measurements of the moving object at successive time points. In the simple case, $\mathbf{x}_i \in \mathbb{R}^2$ represents an object's (estimated) location in the 2-dimensional plane at time point t_i. Depending on the application, the point feature space is extended by, e.g., a third spatial dimension, and/or the current velocity vector (Ristic et al., 2008) or speed and course (Johansson and Falkman, 2007).

2.2.2 Anomaly Detection in Video Surveillance

A significant amount of work related to automated anomaly detection in trajectory data has been published in the video surveillance domain (Morris and Trivedi, 2008a; Hu et al., 2006; Dee and Velastin, 2008). Most of these methods involve *trajectory clustering*, where cluster models corresponding to normal paths/routes are learnt from historical trajectories; this process is referred to as *trajectory learning* (Morris and Trivedi, 2008a) and will be presented in the first part of this subsection. New trajectories are typically assigned an anomaly score based on the distance to the closest cluster model, or likelihood of the most probable cluster model; this will be described in the second part of this subsection. Finally, methods that do not involve trajectory clustering will be presented in the third and last part of this subsection.

Trajectory Learning

Trajectory learning typically consists of three steps that are sometimes blended: preprocessing of raw trajectories, clustering of preprocessed trajectories and modelling of trajectory clusters (Morris and Trivedi, 2008a).

The purpose of the preprocessing is to produce trajectory representations that are suitable for clustering. A typical problem with raw trajectories is that they are not of equal length. According to Morris and Trivedi (2008a), "steps must be taken to ensure a meaningful comparison between differing sized inputs" and "trajectory representations should retain the intuitive notion of similarity present in raw trajectories for meaningful clusters". Two main techniques are used for preprocessing trajectories: *normalisation* and *dimensionality reduction* (Morris and Trivedi, 2008a). Normalisation techniques ensure that trajectories are of equal length by, e.g., extending short trajectories by zero padding (Hu et al., 2006) or re-sampling trajectories. Dimensionality reduction techniques map trajectories into a lower dimensional feature space which is computationally more manageable and enables more robust clustering given less training data (Morris and Trivedi, 2008a). Examples of dimensionality reduction techniques include vector quantisation, various polynomial approximations (Naftel and Khalid, 2006), parametrised spline models (Sillito and Fisher, 2008), wavelets, HMM (Porikli, 2004), principal component analysis and spectral methods (Atev et al., 2010).

The aim of the trajectory clustering is to learn the underlying structure and routes in data by grouping similar trajectories. All clustering algorithms require that an appropriate *similarity measure*, also known as *distance measure*, is defined which constitutes a valid metric. Euclidian distance (ED) is perhaps the most simple and intuitive similarity measure (e.g., Piciarelli et al. (2008)), but it requires that the preprocessed trajectories are of equal length and ED performs poorly if they are not properly aligned (Morris and Trivedi, 2008a). Other similarity measures and modifications of ED have been proposed for relaxing alignment and length constraints, such as Dynamic Time Warping (DTW) (Morris and Trivedi, 2008a). Longest Common Sub-Sequence (LCSS) is another similarity measure appropriate for trajectories that are of unequal length and/or are not well-aligned (Vlachos et al., 2002). In contrast to ED and DTW, LCSS is more robust to noise in trajectories since it focuses on parts of the trajectories that are similar rather than dissimilar (Vlachos et al., 2002). Other alternatives to ED include HMM-based trajectory distance (Porikli, 2004) and variants of the Hausdorff distance (HD) (Atev et al., 2010) (HD will be presented in Section 2.4). Different similarity measures, including those presented above, were evaluated for trajectory clustering on a real world data set by Zhang et al. (2006).

A number of different algorithms have been proposed for creating and updating trajectory clusters based on a specified similarity measure. Morris and Trivedi (2008a) identified five main categories based on fundamental properties

of the clustering algorithm: *iterative optimisation* (e.g., K-means), *online adaptive*, *hierarchical methods* (agglomerate and divisive), *neural networks* (e.g., Self-organising maps (SOM)) and *co-occurrence methods*. The strengths and weaknesses of each category are discussed in their survey (Morris and Trivedi, 2008a). Of particular interest are clustering algorithms that are online adaptive, since they enable continuous and efficient update and adaption to changes in normal trajectory behaviour. Piciarelli and Foresti (2006) proposed such an algorithm for online clustering of trajectories, including a novel similarity measure tailored for the clustering algorithm.

Once trajectories have been clustered, appropriate cluster models, sometimes referred to as *path models* (Morris and Trivedi, 2008a) or *motion patterns* (Hu et al., 2006), are defined. The path models correspond to compact representations of the clusters which enable efficient inference during, e.g., trajectory classification or anomaly detection. Two main types of path models have been adopted: The first considers a complete path, from starting point to endpoint. The second decomposes a path into smaller atomic parts called *subpaths* (Morris and Trivedi, 2008a). Complete path models are typically based on a *centroid* representation of the cluster, corresponding to the "average" trajectory (Morris and Trivedi, 2008a). The centroid is sometimes complemented by an *envelope*, which captures the extension and variance of the trajectories (Morris and Trivedi, 2008a). Hu et al. (2006) model each path model (referred to as motion pattern) as a chain of Gaussian distributions, where the sequence of mean values and covariance matrices define the centroid and envelope, respectively. Morris and Trivedi (2008b) proposed the modelling of each path by a HMM where each (hidden) state is modelled by GMM. Model parameters for each HMM are initially estimated based on the training trajectories belonging to the corresponding cluster. Each HMM can also be updated online as new training data arrives. In case of subpath models, these are further defined by their connections to other subpaths, which can be probabilistically modelled (Morris and Trivedi, 2008a). An example of a subpath structure was proposed by Piciarelli and Foresti (2006), where each subpath is modelled as a node in a tree-like structure augmented with probabilities for node transitions. A subpath in their approach is represented as a sequence of points, where each point has an associated variance, i.e., similar to Hu et al. (2006).

Anomaly Detection

For complete trajectories, anomaly detection is typically carried out by first determining the path model that best explains the new trajectory, i.e., having the minimal distance to the new trajectory. In case of probabilistic models, this can be done based on a MAP or ML analysis (Morris and Trivedi, 2008a). The corresponding distance or likelihood is then usually compared to an anomaly threshold, analogously to general anomaly detection algorithms discussed in Section 2.1. Some algorithms, but far from all, support *sequential,* also known

as *online* (Morris and Trivedi, 2008a) or *incremental* (Hu et al., 2006), anomaly detection in *incomplete* trajectories. Morris and Trivedi (2008b) proposed an algorithm that monitors the likelihood for the part of the current trajectory that is within a sliding window of fixed size; if this likelihood drops below a specified threshold, the trajectory is classified as anomalous. Hu et al. (2006) proposed an algorithm for incremental anomaly detection in incomplete trajectories; for each new data point, the algorithm first updates the most probable motion pattern (trajectory cluster model) given the part of the trajectory observed so far. The likelihood for the new point is then calculated relative the most probable motion pattern and the point is classified as anomalous if its likelihood is below a specified threshold. If a series of points are classified as anomalous, the trajectory is classified as anomalous. Fu et al. (2005) proposed a sequential anomaly detection algorithm that incrementally determines the MAP trajectory cluster as more data points are observed for an incomplete trajectory. The incomplete trajectory is classified as anomalous if: it leaves the envelope of the current MAP cluster, the MAP cluster varies as more data points are observed, or the velocity of the trajectory deviates from the corresponding velocity of the current MAP cluster.

Non-clustering Based Methods

Other approaches to trajectory anomaly detection in video surveillance have been proposed that do not involve clustering of trajectories as presented above (Piciarelli et al., 2008; Yankov et al., 2008; Brax et al., 2008; Owens and Hunter, 2000). Piciarelli et al. (2008) proposed a trajectory learning and anomaly detection algorithm based on one-class SVM (Section 2.1.3) where each trajectory is represented by a fixed-dimensional feature vector corresponding to evenly sampled points from the raw trajectory. One of the main novelties of the algorithm is its ability to automatically detect and remove anomalies in the training data.

Yankov et al. (2008) proposed the application of *time-series discords* (Keogh et al., 2005) for detecting anomalous trajectories in a database of trajectories. Assuming preprocessed trajectories (or sub-trajectories) of equal length, a discord is defined as the trajectory (or sub-trajectory) maximising the ED to its nearest neighbour in the set. This definition can be extend by considering the distance to the kth nearest neighbour. Moreover, if the data set is assumed to include more than one anomalous trajectory, anomaly detection can be carried out by calculating the top-K discords according the distance to nearest neighbour.

Brax et al. (2008, 2009) proposed an algorithm based on multi-dimensional histograms for detecting anomalous trajectory behaviour in video surveillance. Each data point from a trajectory is represented in a discrete multi-dimensional feature space, referred to as a *composite state space*, which encompasses: current kinematic state (position, speed and course), object's position relative the

currently closest other object, object's size, state of local environment and time of the day. A multi-dimensional histogram is constructed for the composite states of all the data points in the training set. Moreover, a separate histogram for each possible transition from a composite state to another is also constructed. Sequential anomaly detection is based on the relative frequency of each new data point according to the histograms. If the average frequency of the most recent data points within a sliding window is below a predefined threshold, the trajectory is classified as anomalous. One of the main features of this approach is that contextual information, i.e., time of day and state of local environment, is embedded in the state (feature) space. Moreover, the proposed method differs from traditional multi-histograms in that the state boundaries are set based on expert knowledge rather than generated from the data itself (Brax et al., 2009).

Owens and Hunter (2000) proposed an algorithm based on SOM appropriate for learning and sequential (online) anomaly detection in trajectory data. Each data point from a trajectory is represented by a fixed-length feature vector encompassing the current location, velocity, and acceleration together with information on the recent position. The SOM is trained using a set of feature vectors as input. During sequential anomaly detection, the feature vector corresponding to each new data point is submitted to the SOM and the ED to the winning neuron is determined. If this distance exceeds a predefined threshold, the corresponding trajectory is classified as anomalous.

Lee et al. (2008) proposed a *partition-and-detect* framework for detection of anomalous sub-trajectories in a trajectory database. A two-step anomaly (outlier) detection algorithm is proposed, which first partitions each trajectory into a number of line segments. Next, anomalous trajectory partitions, i.e., line segments, are detected according to a combination of distance-based and density-based analysis. In case of distance, a combination of spatial distance and angular distance between line segments is used.

2.2.3 Anomaly Detection in Maritime Surveillance

Maritime surveillance is another domain where substantial work related to anomaly detection in trajectory data has been published. In contrast to the video surveillance domain, most of the algorithms proposed for anomaly detection in the maritime domain do not involve explicit clustering of trajectories as part of the learning phase. The papers published in this domain typically give less information on parameter values and implementation details of the proposed algorithms. Moreover, experiments are often described in less detail and data sets (real and simulated) are exclusively non-public. Hence, it is generally more difficult to actually implement the proposed algorithms and reproduce experiments.

A number of methods have been proposed where learning is based on estimation of a statistical model for the feature values of individual data points from vessel trajectories, e.g., the current location and velocity vector (Brax et al.,

2010; Johansson and Falkman, 2007; Ristic et al., 2008; Kraiman et al., 2002). In the work by Ristic et al. (2008), vessel trajectories are first clustered based on their origin, resulting in a number of motion patterns. The PDF for feature values of data points belonging to the same motion pattern are then estimated using KDE. Data points from new trajectories are sequentially classified as normal or anomalous based on their likelihood, which is calculated from the PDF of the corresponding motion pattern. Kraiman et al. (2002) discuss a two-step learning algorithm where feature values of individual data points are first clustered using SOM and then modelled using GMM. Anomaly detection in new data is carried out based on a Bayesian analysis of the probability output of the GMM, which can be accumulated within a sliding time window. A prototypical scenario is presented where properties of the proposed algorithm are discussed. However, no details regarding the learning and anomaly detection algorithm are given.

Other statistical approaches have also been proposed based on Bayesian networks (Johansson and Falkman, 2007) or multi-dimensional histograms (Brax et al., 2010), where point features are discretisized and anomaly scores for new data points calculated based on their likelihood or frequency in training data, respectively. Johansson and Falkman (2007) argue that one of the main advantages of the Bayesian network is that it enables incorporation of human expert knowledge during learning. The multi-histogram method proposed by Brax et al. (2010) has the advantage that it, in contrast to the previous algorithms, captures the relationship between successive data points by modelling the frequency of different state transitions. In order to suppress the effect of noise during anomaly detection, Johansson and Falkman (2007) and Brax et al. (2010) propose a sliding window approach where anomaly scores for individual data points are integrated, i.e., similar to Kraiman et al. (2002) described above.

Urban et al. (2010) proposed a two level approach based on a combination of GMM and HMM for learning and anomaly detection in trajectory data. On the first level, a GMM is estimated for the position feature of individual trajectory points based on EM. One the second level, each component of the GMM is considered a discrete state in a HMM which is estimated from the trajectory data using the *Baum-Welch algorithm*. Anomaly detection in new trajectories is carried out by thresholding the likelihood according to the HMM. Classification performance of the algorithm is evaluated on a non-public simulated data set. A similar two level approach was proposed by Tun et al. (2007), in which clusters, referred to as regions, in the position feature space for vessel trajectories are first discovered using a combination of *density maps* and *Linear scale space*. A HMM is then estimated for the transitions between different regions. Moreover, additional feature values, such as momentary course and speed, are estimated for vessel trajectories within the same region. A short extract from an experimental evaluation with real normal data and simulated anomalous data show that the method can accurately detect the anomalies.

Rhodes et al. (2005, 2007); Garagic et al. (2009); Bomberger et al. (2006) have proposed a number of algorithms for learning and anomaly detection in vessel traffic based on different variants of neural networks. The first algorithm is based on a combination of online unsupervised and supervised learning, implemented by a Fuzzy ARTMAP neural network, where clusters in feature space are incrementally detected and optionally labelled by a human operator (Rhodes et al., 2005). The features considered in their application are momentary location and speed of vessels. New data points are classified as anomalous depending on their distance to nearest cluster. The authors present prototypical scenarios and discuss qualitative properties related to online performance of the algorithm; yet, no details are given regarding parameters and implementation. In a subsequent paper, the authors proposed replacing the Fuzzy ARTMAP algorithm with another mixture-based neural network algorithm with similar properties, including unsupervised and incremental learning (Garagic et al., 2009). Performance of the new algorithm was compared to that of a standard mixture model based on batch learning.

Bomberger et al. (2006) proposed an algorithm based on associative neural network that learns to predict the future location of vessels based on their current location, speed and course. Similar to their previously proposed algorithms, learning is incremental and unsupervised. However, features are also discretisized, where location is specified by the corresponding cell coordinates in a grid over the surveillance area. Anomalous vessel behaviour is detected when current location (cell) of a vessel is inconsistent with the corresponding prediction. This algorithm is considered to be a complement to the previous algorithms, since it is able to detect anomalous behaviour that develops over time, such as anomalous routes. Results for learning and prediction performance of the algorithm based on real vessel data were presented. These results indicated that performance is sensitive to local grid resolution. Hence, the authors investigated an extended approach in a subsequent paper, where improvement in prediction performance was achieved by using multiple spatial scales to represent position (Rhodes et al., 2007).

The online clustering algorithm initially proposed by Piciarelli and Foresti (2006) for video surveillance (Section 2.2.2) was evaluated by Dahlbom and Niklasson (2007) in the domain of maritime surveillance. Dahlbom and Niklasson (2007) discuss some practical problems and issues that were discovered during experiments on a simulated vessel trajectories. The paper is concluded with a discussion regarding an alternative spline-based clustering approach.

2.3 Conformal Prediction

In this section, we describe Conformal prediction, which underpins the algorithms for anomaly detection proposed in Chapter 3.

Conformal prediction is a technique for "hedging" individual predictions made by machine learning algorithms with valid measures of confidence (Gam-

merman and Vovk, 2007). Given a specified *confidence level*, say 95%, conformal predictors output a *prediction set* that contains the true label (value) with probability at least 95%. In the case of classification, conformal predictors may also output the single most likely label and the corresponding confidence, i.e., the confidence in the prediction set of size one.

Assume an example space of the form $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$ where $\mathbf{X}$ is the feature space and $\mathbf{Y}$ is the label space. Having observed the sequence of examples $(x_1, y_1), \ldots, (x_n, y_n)$ and the features x_{n+1} of the next example, the conformal predictor predicts the next label y_{n+1} by producing a prediction set $\Gamma^\epsilon\big((x_1, y_1), \ldots, (x_n, y_n), x_{n+1}\big) \subseteq \mathbf{Y}$ that is *valid* at the specified *significance level* $\epsilon \in (0, 1)$. The validity of the prediction set means that for each successive n, the probability of the event $y_{n+1} \in \Gamma^\epsilon\big((x_1, y_1), \ldots, (x_n, y_n), x_{n+1}\big)$ is at least $1 - \epsilon$ (Gammerman and Vovk, 2007). Thus, the successive predictions made will each be correct with a probability equal to or larger than the corresponding confidence level, even though they are based on an accumulating data set (Gammerman and Vovk, 2007). For example, in case of 95% confidence level, prediction sets will be correct at least 95% of the time. The only assumption is that the data sequence is IID or *exchangeable*, which is a slightly weaker assumption than IID (Shafer and Vovk, 2008).

The basic idea of conformal prediction is to estimate the *p-value*, p_y, for each possible label $y \in \mathbf{Y}$ for the new example and exclude from the prediction set those labels having $p_y < \epsilon$, i.e., analogously with statistical hypothesis testing, the most unlikely labels are rejected at significance level ε (Shafer and Vovk, 2008). In order to estimate these *p*-values, the concept of a *nonconformity measure* (NCM) is introduced, which measures how "different" an example is relative a set of examples. Formally, a NCM is real-valued function $A\big(B, (x, y)\big)$ that returns a nonconformity score α measuring how different the particular example (x, y) is from the set of examples in the *bag* (multi-set) B (Shafer and Vovk, 2008). Since conformal predictors only assume that data is IID or exchangeable, the information of the order is irrelevant and omitted by representing the examples as a bag. By calculating the nonconformity score $\alpha_i = A\big(B_i, (x_i, y_i)\big)$ for each example $(x_i, y_i) \in \big\{(x_1, y_1), \ldots, (x_n, y_n)\big\}$ relative the rest in $B_i = \big\{(x_j, y_j) \in \{(x_1, y_1), \ldots, (x_n, y_n)\} \,|\, j \neq i\big\}$, we can estimate the *p*-value for a particular example (x_i, y_i) as the ratio of nonconformity scores $\alpha_1, \ldots, \alpha_n$ that are at least as large as α_i:

$$p_i = \frac{\big|\{j = 1, \ldots, n \mid \alpha_j \geq \alpha_i\}\big|}{n}. \tag{2.13}$$

Now, including the new example (x_{n+1}, y) with observed features x_{n+1} and hypothetical label y, we expect that any label $y \neq y_{n+1}$, i.e., any label other than the true label, would result in the corresponding α_{n+1} being relatively large compared to $\alpha_1, \ldots, \alpha_n$. This would imply a low *p*-value p_y and that the corresponding example constitutes an outlier. The prediction set Γ^ϵ is formed

by estimating the *p*-value p_y for each $y \in \mathbf{Y}$ and including in the prediction set those y having a *p*-value greater than the significance level ϵ.

The conformal predictor presented above is *conservatively valid* in the sense that the probability of error is bounded by ϵ, i.e., $Pr\big(y_{n+1} \notin \Gamma^{\epsilon}\big) \leq \epsilon$ (Gammerman and Vovk, 2007). It can easily be extended to a *smoothed conformal predictor* that produces prediction sets that are *exactly valid*, which means that probability of error equals ϵ, i.e., $Pr\big(y_{n+1} \notin \Gamma^{\epsilon}\big) = \epsilon$ (Gammerman and Vovk, 2007). The smoothed conformal predictor is obtained by simply updating equation 2.13 to:

$$p_i = \frac{\big|\{j \mid \alpha_j > \alpha_i\}\big| + \tau_i \big|\{j \mid \alpha_j = \alpha_i\}\big|}{n}, \tag{2.14}$$

where $\tau_i \in [0, 1]$ is a random variable sampled from the uniform distribution on the unit interval (Gammerman and Vovk, 2007).

A (smoothed) conformal predictor will produce valid and nested prediction sets using *any* real-valued function $A\big(B, (x, y)\big)$ as the NCM (Shafer and Vovk, 2008). However, the prediction sets will only be small if an appropriate NCM is chosen that measures well how different (x, y) is from B, i.e., precision is highly dependent on the chosen NCM. Therefore any available domain knowledge regarding the structure of the underlying process should be exploited when defining NCMs for specific applications. Yet several general NCMs for both classification and regression based on different modern machine learning algorithms have been proposed, including the k-nearest neighbours algorithm, (kernel) ridge regression, support vector machines and neural networks (Vovk et al., 2005).

2.4 Hausdorff Distance for Shape Matching

This section presents the *Hausdorff distance* (HD), which later will be used for anomaly detection in trajectory data (Chapter 4).

Generally speaking, HD is a dissimilarity measure for two sets of points in a metric space. It is a well known distance measure in the field of computational geometry and image processing, where it has been applied for shape matching and shape recognition (Alt, 2009). Given two sets of points $A \subseteq \mathbb{R}^D$ and $B \subseteq \mathbb{R}^D$, the *directed* HD from A to B, $\overrightarrow{\delta_H}(A, B)$, corresponds to the maximum distance from a point $a \in A$ to the closest point of B, where distance between points is measured by some metric $d(a, b)$, typically ED:

$$\overrightarrow{\delta}_H(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\}. \tag{2.15}$$

Assuming that the point sets represent different shapes, the directed HD captures the degree to which shape *A* resembles some *part* of shape *B*. It should be noted that this measure is not a metric since it is not symmetric; the reverse HD

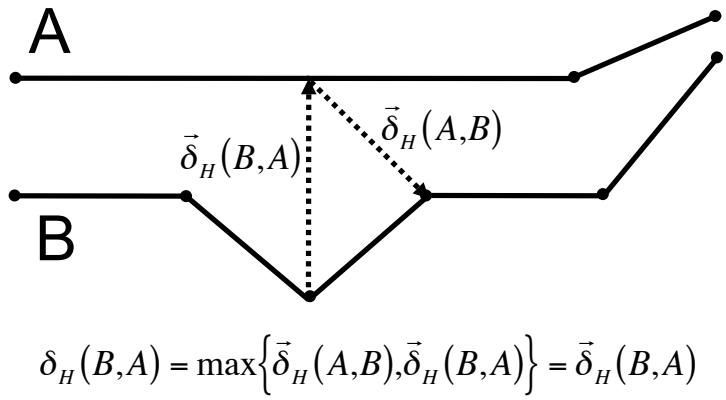

Figure 2.1: Illustration of the directed Hausdorff distance (HD) $\overrightarrow{\delta_H}$ and undirected HD δ_H between two polygonal curves A and B.

$\overrightarrow{\delta_H}(B, A)$ is in general different from $\overrightarrow{\delta_H}(A, B)$. In order to compare complete shapes, the *undirected* HD between A and B, $\delta_H(A, B)$, can be calculated as the maximum of the two directed distances:

$$\delta_H(A, B) = max\left\{\overrightarrow{\delta}_H(A, B), \overrightarrow{\delta}_H(B, A)\right\}. \tag{2.16}$$

This distance measure is symmetric and constitutes a valid metric for all closed, bounded sets A and B (Huttenlocher et al., 1993). The directed and undirected HD between two polygonal curves are illustrated in Figure 2.1.

If A and B are represented as finite sets of points, there is a naive algorithm for calculating the (directed or undirected) HD in time $O(nm)$, where n and m are the number of points in A and B, respectively (Alt, 2009). However, there is no similar straightforward algorithm for calculating the exact HD when A and B are represented as polygons, polygonal curves or other parametric curves, which are common in shape matching applications (Alt, 2009). In the case of polygons and polygonal curves, it is not enough to consider the distance from each vertex of A (B) to each line segment of B (A), since HD can occur between two interior points of two line segments of A and B (Alt, 2009). Yet an efficient algorithm for calculating the exact HD between two polygons in time $O\big((n+m)+log(n+m)\big)$ has been proposed, where n and m are the number of line segments of the corresponding polygons (Alt et al., 1995). The algorithm is based on the fact that the HD between two sets of disjoint line segments A and B can only occur at points that are either endpoints of line segments or intersection points of the Voronoi diagram of one of the sets with a segment of the other (Alt, 2009). This property, together with further observations, reduce

the number of possible points along each line segment where HD may occur, from infinitely many to $O(n+m)$ (Alt, 2009).

Chapter 3
Conformal Anomaly Detection

In this chapter, we theoretically investigate algorithms for anomaly detection. We start off by discussing various issues and limitations of previously proposed algorithms (Section 3.1). This is followed by a discussion of a novel application of Conformal prediction (CP) for multi-class anomaly detection (Section 3.2.1). The remaining part of the chapter (Section 3.3) is dedicated to the proposed *Conformal Anomaly Detector* (CAD), which is a general algorithm for anomaly detection with well-calibrated false alarm rate. We identify and discuss key properties of CAD. We also present the *Similarity-based Nearest Neighbour Conformal Anomaly Detector* (SNN-CAD), which is well-suited for anomaly detection in data represented as sets or sequences of varying size or length, such as trajectories.

3.1 Issues with Previous Anomaly Detection Algorithms

3.1.1 Assumptions on the Underlying Distribution

Simple parametrised statistical models, such as the Gaussian distribution, are attractive because parameter estimation is rather straightforward, and because there exists a number of general tests that provide well-founded confidence for anomaly detection (Section 2.1.2). However, accuracy of estimated models, and robustness of anomaly detection, depend on whether the structural assumptions are valid. For example, fitting a Gaussian model to data generated from an approximately uniform distribution will result in overestimation and underestimation of the probability density at the centre (mean) and the tails of the distribution, respectively. Generally, in case of underestimation, the false alarm rate will increase since normal data points will be assigned a lower likelihood. And in case of overestimation, the sensitivity to anomalies will decrease, since there is an increased risk that subtle anomalies will be erroneously classified as normal. In some real world applications, where behaviour dynamics

are governed by social systems and structures rather than by physical laws, the Gaussian assumption may indeed be questioned (cf. Taleb (2004)). Moreover, traditional statistical methods may encounter serious conceptual and computational difficulties when applied to high-dimensional data sets (Gammerman and Vovk, 2007).

3.1.2 Parameter-laden Algorithms

It seems that most anomaly detection algorithms require careful preprocessing and setting of multiple application specific parameters and detection thresholds in order to achieve (near) optimal performance; trajectory anomaly detection is no exception to this. In fact, Keogh et al. (2007) argue that most data mining algorithms are more or less *parameter-laden*, which is undesirable for several reasons. In an extensive empirical study, they showed that "in case of anomaly detection, parameter-laden algorithms are particularly vulnerable to overfitting" (Keogh et al., 2007). The risks of overfitting parameter-laden models in real world applications were also discussed by Hand (2006), who showed empirically that "the marginal gain from complicated models is typically small compared to the predictive power of the simpler models". According to Markou and Singh (2003a), "an [anomaly] detection method should aim to minimise the number of parameters that are user set". Indeed, one could argue that use of few parameters suppresses bias towards particular types of anomalies and makes the method easier to implement for different applications.

3.1.3 The Problem of Setting the Anomaly Threshold

The anomaly threshold is a central parameter in all anomaly detection algorithms, since it regulates the sensitivity to true anomalies and the rate of false alarms (false anomalies). Nearest-neighbour methods and clustering methods (Section 2.1.3) typically rely on a distance or density threshold for deciding whether a data point is anomalous or not. Many statistical methods, in particular non-Gaussian and non-parametric methods, rely on a likelihood (density) threshold (Section 2.1.2). These distances or densities are typically not normalised and the procedures for setting the thresholds seem to be more or less ad-hoc. Typically, thresholds are static and set once using, e.g., the training set or a separate tuning set. The interpretation of the distances, densities and thresholds are not very intuitive to, e.g., an operator of a surveillance system. In case of modern classification-based methods, such as one-class SVM, it has been argued that "while these approaches provide impressive computationally efficient solutions on real data, it is generally difficult to precisely relate tuning parameter choices to desired false alarm probability" (Zhao and Saligrama, 2009). The difficulty of tuning the anomaly threshold has consequences regarding the effectiveness and usefulness of an anomaly detection system; according

to Axelsson (2000), "the false alarm rate is the limiting factor for the performance of the [anomaly] detection system". Moreover, Riveiro (2011) argues that "the primary and most important challenge that needs to be met for using [an anomaly detection] approach is the development of strategies to reduce the high false alarm rate". Riveiro (2011) goes on arguing that "the inability of suppressing false alarms is a perennial problem that prevents the widespread adoption of anomaly detection capabilities" and that "this obstacle may become a nuisance for operators that might turn them off".

3.2 Conformal Prediction and Anomaly Detection

The theory of Conformal prediction was developed for supervised learning and prediction applications (Section 2.3). However, we argue that theoretical properties of Conformal prediction also make it well-suited for one-class and multi-class anomaly detection. The key observation is that the *p*-value estimated for an observed example is a general and useful anomaly measure. Based on the concepts of NCM and *p*-values, we will in Section 3.3.1 define the Conformal Anomaly Detector (CAD), which is a general one-class anomaly detection algorithm. But before we introduce CAD, let us discuss how a conformal predictor operating in the supervised online learning and prediction setting can be used for multi-class anomaly detection.

Assume an example space $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$ where $\mathbf{X}$ is the feature space and $\mathbf{Y}$ is a finite set of normal classes. Based on the set of previously observed examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and observed features x_{n+1} of the next example, a conformal predictor will output a prediction set $\Gamma^\epsilon \subseteq \mathbf{Y}$ for the class y_{n+1} at the specified significance level ϵ. Assuming that the significance level is fairly low (e.g., 1% or less), an interesting situation arises if the prediction set is empty; this happens if all *p*-values (one for each possible class) are below ϵ. An empty prediction set indicates that the new example has novel features that do not match well any of the known classes. According to Gammerman and Vovk (2007), "[an empty prediction set at a low significance level] means that either the training set is non-random or the test [example] is not representative of the training set". As an example, the authors discuss an Optical Character Recognition (OCR) application, where an empty prediction set may arise if the new image corresponds to a letter while the training set consists of images of digits (Gammerman and Vovk, 2007). Another interesting situation arises if a non-empty prediction set does not include the true label (which is revealed after each prediction during online learning). This means that features were recognised but the (later observed) class was unexpected. An example of such situation would be an badly written "7" that resembles a "1" (or vice verse). In both situations, the prediction set is incorrect and we may suspect that the corresponding example is an anomaly, i.e., it was not generated from the same distribution as the training set. In general, if the new example and the training set are IID, we know that the probability of error, i.e., that the prediction

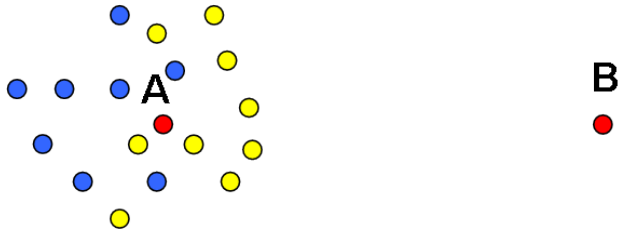

Figure 3.1: Illustrating the problem with the NCM based on distances to nearest neighbours of the same class and nearest neighbours of some other class, for anomaly detection applications.

set is incorrect, is bounded by ϵ (Section 2.3). thus, if we classify examples as anomalous if the corresponding prediction set is incorrect, we expect that the frequency of false alarms will be less than ϵ. Hence, we can bound false alarm rate by the specified significance level ϵ.

3.2.1 A Nonconformity Measure for Multi-class Anomaly Detection

A nearest neighbour NCM suitable for classification applications has previously been proposed (Gammerman and Vovk, 2007). It is based on ED in feature space to the *k*-nearest examples with the same label and the *k*-nearest examples with a different label:

$$\alpha_i := \frac{\sum_{j=1}^{k} d_{ij}^{+}}{\sum_{j=1}^{k} d_{ij}^{-}}, \tag{3.1}$$

where d_{ij}^{+} is the *j*th shortest distance from x_i to other examples labelled in the same way as x_i and d_{ij}^{-} is the *j*th shortest distance from x_i to examples labelled differently from x_i. The intuition behind this NCM is rather simple: It is assumed that examples of the same (normal) class are close to each other in feature space, while examples of different (normal) classes are further away from each other. However, we argue that this NCM is suboptimal in multi-class anomaly detection applications when there are reasons to believe that normal classes overlap in feature space. We will show this point with the following principle example, illustrated by Figure 3.1: Assume that we have a binary classification problem where the two class distributions (blue and yellow) overlap in a high-density area in feature space. Considering the red point A, we expect its distance to the closest examples of each class to be approximately the same. So its nonconformity score using Equation (3.1) would be close to 1, regardless of which class it actually belongs to. Moving away from both distributions and considering the red point B, the ratio of the distances to the closest examples

of each class will approach 1. Hence, the p-value of a normal point located in the overlapping high density area will be close to the p-value of an obviously anomalous point located far away from both class distributions. Therefore, we propose a modified NCM in the case of partially overlapping class distributions that simply omits the distances to other classes:

$$\alpha_i := \sum_{j=1}^{k} d_{ij}^{+}. \tag{3.2}$$

We expect this NCM to be more sensitive to anomalous points located far away from all known classes.

3.3 Conformal Anomaly Detection

In some applications, we may only be interested in detecting anomalies and not determining which, if any, of the normal classes that fits the example best. In this case, it makes more sense to only check the p-value for the *observed* label, rather than calculating p-values for *all* possible labels $\mathbf{Y}$ as done during label prediction. And in case of one-class anomaly detection, i.e., when we are dealing with only a single normal class, the feature-label space $\mathbf{X} \times \mathbf{Y}$ does not make any sense since $|\mathbf{Y}| = 1$. Moreover, by calculating smoothed p-values according to Equation (2.14), we get a more precise notion of expected false alarm rate. Based on these observations and Hawkins' definition of an outlier (Section 2.1), we formalise the Conformal Anomaly Detector (Algorithm 3.1) in Section 3.3.1 below.

3.3.1 The Conformal Anomaly Detector

Assume that we are observing a stream of examples corresponding to observed behaviour in some domain of interest. For each new example z_n, the task of the Conformal Anomaly Detector (CAD) is to decide whether z_n is anomalous relative the training set of previously observed examples $\{z_1, \ldots, z_{n-1}\}$, given a NCM A and an anomaly threshold $\epsilon \in (0, 1)$. If the smoothed p-value p_n for z_n is below the anomaly threshold ϵ, z_n is classified as a *conformal anomaly* at significance level ϵ. The p-value can be interpreted as the probability of erroneously rejecting the null hypothesis that z_n was independently generated from the same distribution as the previous examples (Shafer and Vovk, 2008). In other words, the p-value corresponds to the probability of classifying z_n as anomalous when it is in fact not; this is known as a *false alarm*. This observation leads to the following theorem:

Theorem 3.1. *If the new example z_n and the training set $\{z_1, \ldots, z_{n-1}\}$ are independent and identically distributed, the probability of false alarm of a Conformal Anomaly Detector is equal to ϵ.*

Algorithm 3.1 The Conformal Anomaly Detector (CAD)

Input: Nonconformity measure A, anomaly threshold ϵ, old examples $z_1, \ldots, z_{n-1}$ and new example z_n.
Output: Boolean variable $Anomaly$.

1: $D = \{z_1, \ldots, z_n\}$
2: **for** $i \leftarrow 1$ to n **do**
3: $\quad \alpha_i \leftarrow A\left(D \backslash z_i, z_i\right)$
4: **end for**
5: $\tau \leftarrow U\left(0, 1\right)$
6: $p_n \leftarrow \frac{|\{i:\alpha_i > \alpha_n\}| + \tau |\{i:\alpha_i = \alpha_n\}|}{n}$
7: **if** $p_n < \epsilon$ **then**
8: $\quad Anomaly \leftarrow$ **true**
9: **else**
10: $\quad Anomaly \leftarrow$ **false**
11: **end if**

Assuming online learning with an accumulating training set of IID examples corresponding to normalcy, we expect from Theorem 3.1 that the *false alarm rate*, i.e., the frequency of normal examples erroneously classified as anomalous, will be close to ϵ over time. We refer to this property of CAD as *well-calibrated false alarm rate*. The parameter ϵ regulates the sensitivity to true anomalies and should be set depending on the rate of false alarms that is acceptable in the current application. A higher value of ϵ increases probability of detecting true anomalies but also the frequency of false alarms.

3.3.2 Interpretation of a Conformal Anomaly

There are at least three different explanations for a conformal anomaly. Firstly, it may correspond to a rare or previously unseen, yet normal, example; such examples happen with probability ϵ and are false alarms. Secondly and thirdly, it may be a true anomaly in the sense that it was not generated according to the same distribution as the training data; either the example is a true novelty, or the training data itself is in fact not IID. A non-IID training set may be explained by incomplete or biased data collection. In a surveillance application, for example, observed behaviour during early morning or late afternoon may appear anomalous if the training set is based on data recorded during a limited time of the day, e.g., 10 a.m. to 2 p.m. Another possible reason for a non-IID training set is that the underlying distribution has actually changed. In the context of surveillance, new trajectories may arise as a result of, e.g., new traffic regulations. Indeed, if the (false) alarm rate starts to deteriorate from ϵ, there are reasons to suspect that the distribution for normal behaviour has changed recently.

3.3.3 Online Semi-supervised Learning

New examples classified as normal by CAD should automatically be added to the training set. New examples classified as anomalous would ideally be confirmed or rejected by an *oracle*, which is typically a human expert in the context of surveillance. If rejected, the anomaly is considered a false alarm and the corresponding example should be added to the training set; this corresponds to the case of a rare yet normal event. If confirmed, there would be one or two possible responses depending on whether we are concerned with one-class or multi-class anomaly detection. In case of one-class anomaly detection, a confirmed anomaly should simply *not* be added to the training data. But in case of multi-class anomaly detection, an additional option is to let an oracle (human expert) either re-label the new example as one of the existing normal classes, or labelled it as a new normal class; the latter would correspond to the discovery of a new relevant class.

In case where no (immediate) feedback is available, examples classified as anomalous by CAD could be stored in a separate log for future scrutiny and action. However, this would result in a biased training set, since extreme, yet normal, examples are left out when updating the training set online. Hence, the training set would not be IID relative new normal data and, consequently, the false alarm rate would no longer be guaranteed to be well-calibrated (cf. Theorem 3.1). Conversely, if *all* new examples, regardless of their classification, were added to the training set, sensitivity to true anomalies may decrease if the training set becomes corrupted with true anomalies.

3.3.4 The Choice of Nonconformity Measure

Analogously to a conformal predictor, the choice of NCM is of central importance to the performance of CAD. The more sophisticated the NCM, the more sensitive the algorithm is to subtle anomalies. Yet, computational efficiency is also important, since CAD is based on online learning. In this thesis, we will focus on NCMs based on the nearest neighbour algorithm. According to Russel and Norvig (2003), nearest neighbour methods are very easy to implement, require little in the way of tuning, and often performs quite well. In particular, nearest neighbour methods are well-suited for online learning since they do not require extensive update of a prediction model when new training data is added (compared to, e.g., neural networks and support vector methods). Moreover, nearest neighbour algorithms have some strong consistency results: The asymptotic error rate is less than twice the Bayes error rate, which corresponds the minimum achievable error rate given the distribution of the data (Duda and Hart, 1973).

Previously proposed NCMs, including the one proposed in Section 3.2.1, require that examples are represented as data points in a feature space with fixed dimensions. This may be a problem in applications where input examples

are represented as sequences or sets of data points of variable length or size, respectively. Trajectories is one such example.

3.3.5 Similarity-based Nearest Neighbour Conformal Anomaly Detector

Based on the observations in Section 3.3.4 above, we propose the *Similarity-based Nearest Neighbour Nonconformity Measure* (SNN-NCM):

$$\alpha_i := \sum_{j=1}^{k} S(z_i, z_j). \tag{3.3}$$

This NCM is a generalisation of Equation 3.2, where ED is replaced by a dissimilarity measure S and z_j corresponds to the jth most similar neighbour to z_i. It allows for a more flexible comparison since examples are not necessarily required to have the same size or length. Moreover, S is not required to be *symmetric*, i.e., $S(z_i, z_j)$ can be different from $S(z_j, z_i)$. This property will be exploited in Section 4.4.1 where we define an asymmetric trajectory dissimilarity measure based on directed HD.

Now, let us introduce the *Similarity-based Nearest Neighbour Conformal Anomaly Detector* (SNN-CAD, Algorithm 3.2), which classifies a new example z_n as normal or anomalous based on:

- previously observed examples $z_1, \ldots, z_{n-1}$,
- a dissimilarity measure S,
- an anomaly threshold ϵ,
- a sorted dissimilarity matrix M^{n-1} of size $(n-1) \times k$, where each row i correspond to the dissimilarity values of the k-most similar examples to z_i, sorted in ascending order.

When classifying the new example z_n, SNN-CAD first updates the sorted dissimilarity matrix M^n based on the previous matrix M^{n-1} as follows: For each old example z_i, $i = 1, \ldots, n-1$, the algorithm updates the dissimilarity values of its k most similar neighbours by inserting the new dissimilarity value $S(z_i, z_n)$ at the appropriate column index of the sorted row i of M (lines 3–8 of Algorithm 3.2). The $(k+1)$th most similar neighbour to z_i is simply discarded after the insertion. In a similar manner, the algorithm calculates the reverse dissimilarity $S(z_n, z_i)$ and updates the last row n of M (lines 9–18).

Having updated the dissimilarity matrix, SNN-CAD then calculates the nonconformity score α_i for each example z_i, $i = 1, \ldots, n$, based on Eq. 3.3 (lines 20–22). Finally, the smoothed p-value for the new example is calculated and used for classification (lines 23–29). The output of the algorithm is the classification of z_n and the updated dissimilarity matrix M^n. If the new example

z_n is added to the training set (see Section 3.3.3), M^n is used as input when classifying the next example z_{n+1}. Hence, the algorithm has a recursive flavour since M^n is calculated based on M^{n-1}.

Complexity Analysis

The two inner loops (lines 4–7 and lines 13–16) each has worst-case complexity $O(k)$. Hence, the worst-case complexity of the first outer loop (lines 2–19) is $O(nk)$. The second outer loop (lines 20–22) and line 24 have complexity $O(nk)$ and $O(n)$, respectively. Thus, the overall complexity of Algorithm 3.2 is $O(nk)$. Assuming that k is a (relatively small) constant, the algorithm scales linearly in the size of the training data.

The space complexity of SNN-CAD is also $O(nk)$, since the size of the dissimilarity matrix M^n is $n \times k$. Thus, space complexity also scales linearly with the size of training set.

3.4 Discussion

Compared to most other anomaly detection algorithms, we argue that SNN-CAD is a parameter-light algorithm. In the simple case of $k = 1$ and dissimilarity calculated as ED in feature space, the only parameter left is the anomaly threshold ϵ, which is application independent and regulates sensitivity to anomalies and rate of false alarms. This simple, yet general, NCM is applicable in many domains. The idea of calculating the anomaly score based on ED to k-nearest neighbours has indeed been investigated by other authors (Eskin et al., 2002; Angiulli and Pizzuti, 2002). Yet, other more domain specific dissimilarity measures than ED may be used in SNN-CAD; one such example is HD, which will be presented in Section 4.4.

As is the case with nearest neighbour methods in general, the optimal value of k in terms of classification performance is application and data dependant. However, it may be argued that a higher value of k increases robustness to spurious anomalies and other noise in training data, since their impact will effectively be smoothed out. On the other hand, a larger value of k implies higher computational complexity. Moreover, in case of multi-class anomaly detection, precision decreases with higher k since the boundaries between different normal classes are also smoothed out.

The NCM used in CAD can, analogously with a conformal predictor, be chosen quite freely. In particular, we may use any previously proposed anomaly detection algorithm that outputs an anomaly score for an individual example (data point). As an example, the inverse or negative of the data likelihood for a statistical model could serve as a nonconformity score. Most, if not all, variants of nearest neighbour and cluster methods may potentially be used as a NCM, since the corresponding distance to nearest neighbour or cluster can be interpreted as a nonconformity score. Hence, CAD encompasses a multitude of the

Algorithm 3.2 Similarity-based Nearest Neighbour Conformal Anomaly Detector (SNN-CAD)

Input: Dissimilarity measure S, anomaly threshold ϵ, number of most similar neighbours k, old examples $z_1, \ldots, z_{n-1}$, dissimilarity matrix M^{n-1} and new example z_n.

Output: Dissimilarity matrix M^n, boolean $Anomaly$.

$M \leftarrow M^{n-1}$
for $i \leftarrow 1$ to $n-1$ **do**
  $j \leftarrow k$
  while $j > 0$ **and** $S(z_i, z_n) < M_{i,j}$ **do**
    $M_{i,j+1} \leftarrow M_{i,j}$
    $j \leftarrow j - 1$
  end while
  $M_{i,j+1} \leftarrow S(z_i, z_n)$
  if $i \leq k$ **then**
    $M_{n,i} \leftarrow S(z_n, z_i)$
  else
    $j \leftarrow k$
    while $j > 0$ **and** $S(z_n, z_i) < M_{n,j}$ **do**
      $M_{n,j+1} \leftarrow M_{n,j}$
      $j \leftarrow j - 1$
    end while
    $M_{n,j+1} \leftarrow S(z_n, z_i)$
  end if
end for
for $i \leftarrow 1$ to n **do**
  $\alpha_i \leftarrow sum(M_{i,1}, \ldots, M_{i,k})$
end for
$\tau \leftarrow U(0, 1)$
$p_n \leftarrow \frac{|\{i:\alpha_i > \alpha_n\}| + \tau |\{i:\alpha_i = \alpha_n\}|}{n}$
if $p_n < \epsilon$ **then**
  $Anomaly \leftarrow$ **true**
else
  $Anomaly \leftarrow$ **false**
end if
$M^n \leftarrow M$

algorithm categories discussed by Chandola et al. (2009), including statistical, nearest neighbour and cluster methods.

Yet, it may be argued that CAD is essentially a statistical approach to anomaly detection, since it is based on Hawkins' definition of an outlier (Section 2.1); it is assumed that normal data, i.e., the training set, is generated from a stationary probability distribution, and that anomalies correspond to low probability events. Similar to statistical hypothesis testing, CAD is based on thresholding p-values, which is convenient since the threshold is associated with a significance level that corresponds to the expected false alarm rate. However, unlike the traditional methods, CAD does not require that the underlying distribution belongs to a parametric distribution, such as the Gaussian. While non-parametric methods also avoid making such assumptions, they are, in contrast to CAD, not truly distribution-independent, since they still require that the underlying distribution is estimated. This estimation is typically associated with some error, in particular when dealing with complex, high-dimensional and continuous distributions, and may require large amounts of data for accurate density estimates. Moreover, thresholding is typically less straightforward when using non-parametric methods, as discussed in Section 3.1.3. Classification-based methods, such as SVM, share the property of distribution-independence and are appropriate for high-dimensional data sets. However, as discussed in Section 3.1, they involve setting of various application specific thresholds and it may not be obvious how to relate these to desired false alarm rate.

Similar to classification-based and statistical anomaly detection algorithms (Section 2.1.3), CAD can be categorised as a semi-supervised anomaly detection algorithm, since it assumes that the training set consists of only normal data. Moreover, it can also be classified as an online and semi-supervised learning algorithm; the training set is incrementally updated with new normal examples that have either been classified as normal by CAD, or first classified as anomalous by CAD and then corrected by a human.

3.5 Summary

In this chapter, we have proposed and discussed CAD, which is a general algorithm for online learning and anomaly detection. CAD is based on the theory of Conformal prediction; a central property that follows from this is that the false alarm rate is well-calibrated, i.e., the expected frequency of normal examples misclassified as anomalous equals the specified anomaly threshold $\epsilon \in (0, 1)$, under the assumption that training data and new normal data are IID. Thus, no application specific anomaly threshold is required. Analogously to Conformal prediction, the main design parameter in CAD is the NCM. We have proposed a new NCM, SNN-NCM, which is appropriate in applications where data is represented as sets or sequences of different size or length. Apart from the number of most similar neighbours k, the only design parameter in SNN-NCM is the dissimilarity measure S which can be chosen quite freely. A

simple dissimilarity measure applicable in many domains is ED in feature space. Yet, other more application specific dissimilarity measures may be used, which may be asymmetric and do not necessarily require that data points are of equal size or length. Based on CAD and SNN-NCM, we have proposed and described the SNN-CAD algorithm which scales linearly in the size of the training data.

Chapter 4
Anomaly Detection in Trajectory Data

In this chapter, we theoretically investigate different algorithms for sequential anomaly detection in trajectory data. We start by discussing some limitations of previously proposed algorithms for anomaly detection in trajectory data (Section 4.1). Next, two main approaches to trajectory anomaly detection are considered, which differ in the nature of the feature model adopted: point-based and trajectory-based anomaly detection. For point-based anomaly detection, three different algorithms are proposed: The first and the second are based on parametric (GMM) and non-parametric (KDE) statistical models, respectively. The third algorithm proposed is a specialised version of CAD called the Single Point Trajectory Conformal Anomaly Detector (SPT-CAD), which is based on the Single Point Trajectory Nonconformity Measure (SPT-NCM). For trajectory-based anomaly detection, we adopt SNN-CAD (Section 3.3.5) and propose two parameter-free dissimilarity measures based on HD for comparing multi-dimensional trajectories of arbitrary length. One of these measures is appropriate for sequential anomaly detection in incomplete trajectories.

4.1 Issues with Previous Algorithms

A number of previously proposed algorithms for learning and anomaly detection in trajectory data were briefly presented in Section 2.2. No details were given there regarding the implementation of the algorithms; yet, it seems that most of the algorithms involve setting of multiple parameters prior to learning and anomaly detection. Some of the algorithms also require substantial preprocessing of input data, which may imply even more parameters. For example, some algorithms require that each continuous feature is appropriately discretised by a human expert (e.g., Brax et al., 2008). Hence, they are, similar to data mining and anomaly detection algorithms in other domains, more or less parameter-laden, which is undesirable for many reasons (Section 3.1).

Moreover, most algorithms seem to lack intuitive principles for setting the anomaly threshold, and properties related to well-calibrated false alarm rate seem to be weak or nonexistent. Thus, the performance, generalisability and usability of the proposed algorithms in real world surveillance applications may be questioned, particularly for those algorithms that have only been evaluated on a single synthetic data set, or a relatively small real world data set.

In addition to the general observations and discussion above, a few more or less domain specific observations can be made regarding previously proposed algorithms. To start with, it seems that most of the previous work within video surveillance is focused towards *offline anomaly detection;* it is explicitly or implicitly assumed that the algorithm has observed the *complete* trajectory before it is classified as anomalous or not. This is quite obvious when considering algorithms that require preprocessed trajectories prior to clustering or anomaly detection, such as those based on dimensionality reduction and other normalisation techniques discussed in Section 2.2.2. The focus towards offline anomaly detection is further emphasised by the fact that many algorithms are defined for anomaly detection in trajectory *databases* (e.g., Yankov et al. (2008); Lee et al. (2008)). Moreover, most similarity measures, including ED, DTW and LCSS, are designed for complete trajectories, even though some of them (e.g., LCSS) are more robust to missing data than others. As discussed in the introduction to this thesis, offline anomaly detection is a serious limitation in surveillance applications, since it delays anomaly alarms and thus the ability to react to impending events. Hence, focus should be on developing algorithms that are designed for *online* or *sequential* anomaly detection in *incomplete trajectories*, supporting real-time detection of anomalous trajectories as they evolve.

With a few exceptions (notably Piciarelli and Foresti (2006); Bomberger et al. (2006); Bu et al. (2009)), it seems that learning in previous algorithms for trajectory anomaly detection is *offline*; fixed model parameters and thresholds are typically estimated or tuned once, based on a batch of historical data. The advantage of efficient and continuous *online learning*, also referred to as *incremental learning* (Ekman and Holst, 2008), based on new trajectories, was pointed out by Piciarelli and Foresti (2006), who proposed an online trajectory clustering algorithm. Rhodes et al. (2007) went as far as arguing that it is "essential that learning occurs incrementally in order to allow the system to take advantage of increasing amounts of data without having to take the system offline" and that "an additional benefit [of online learning] is that the system will be able to adapt to changing behaviour patterns automatically". Yet, it may be counter-argued that taking the system offline is not strictly necessary in case of an offline learning process that runs in parallel with the anomaly detector, where updated model parameters are introduced at specific time points. For example, some authors discuss regular batch-learning repetitions using an updated training set (Morris and Trivedi, 2008b). However, it is not clear how to determine the appropriate time point for such an update. On the one hand,

minimising the learning delay, i.e., the time between successive model updates, may be desirable in order to maintain a more timely and accurate model. But on the other hand, computational complexity of offline learning algorithms, which is typically higher than for online algorithms, may impose restrictions on how low delay that can be achieved in a practical application.

An algorithm specifically developed for online learning and anomaly detection in a single continuous trajectory stream was recently proposed by Bu et al. (2009). Yet, the authors do not discuss how or if this algorithm can be applied for anomaly detection in *multiple* trajectories, i.e., when the current trajectory (stream) is compared to multiple previous trajectories.

4.2 Point-based vs. Trajectory-based Anomaly Detection

A key challenge when designing an anomaly detector is how to represent the data in which anomalies are to be found, i.e., how to find an appropriate feature model that captures the "right" type of anomalies. In case of trajectories, we may consider a low-dimensional feature space corresponding to individual trajectory points; we refer to such approaches as *point-based anomaly detection*. Alternatively, we may consider a high-dimensional feature space that embeds the complete trajectory or segments of the trajectory. Such approaches are referred to as *trajectory-based anomaly detection*.

4.3 Point-based Anomaly Detection

A simple approach to anomaly detection in trajectories is to estimate a model for the feature vector, $\mathbf{x} \in \mathbb{R}^d$, of an arbitrary data point of a trajectory, and use this model for detecting anomalous data points in new trajectories. For example, we may estimate a model for the momentary position and velocity vector, $(x, y, vx, vy) \in \mathbb{R}^4$, of trajectories.

4.3.1 Statistical Approaches

Adopting a statistical approach to anomaly detection (Section 2.1.2), it is assumed that the feature values $\mathbf{x} \in \mathbb{R}^d$ of data points from normal trajectories are independently and randomly generated according to an unknown probability distribution P. Given a model, $p(\mathbf{x})$, for P, we can calculate the data likelihood for an individual trajectory point $\mathbf{x}'$, under the assumption that it is normal. If the likelihood $p(\mathbf{x}')$ is below a predefined threshold, we may suspect that $\mathbf{x}'$ was not generated according to P and, thus, it may be classified as an anomaly.

Similar to other real world applications, it is unknown what is the parametric form of P, if such even exists. Thus, in order to estimate P from our

sample data, we need a flexible model that makes relatively weak assumptions regarding the structure of P. In this thesis, we consider two such techniques based on GMM and KDE, which are standard approaches to parametric and non-parametric density estimation, respectively (Section 2.1.2 and 2.1.2).

GMM

GMM is probably the most commonly used parametric density model for approximating arbitrary continuous multivariate PDFs when there are no particular knowledge or assumptions regarding the structure of the distribution (Section 2.1.2). One of the main advantages of GMM is that it is relatively easy to implement algorithms, e.g., EM, for estimating the parameters. However, the standard version of EM requires that the number of Gaussian components C is specified in advance; this parameter is typically unknown. If C is set too small, the resulting GMM will be too simple and, hence, underfit to the data. Conversely, too large C will result in a complex model which overfits to the data.

A relatively simply approach to solve the problem of determining a suitable number of components is to estimate multiple mixture models with different number of components, i.e., with different values of C, and use a holdout method to determine when we are starting to overfit the model to the data. In this thesis, we propose such an algorithm that is based on iterative EM (Algorithm 4.1) and which takes the following parameters as input: training set D_{Train}, model validation set D_{Val}, maximum number of components (iterations) C_{max} that will be considered and number of candidate models q to consider during each iteration. For each iteration, starting with $C = 1$, the algorithm estimates q C-component models based on D_{Train} using standard EM, where initial parameters for each of the q models are randomly generated. For each of the q models, the average of the validation data likelihood is computed; if the largest of these q likelihood values is lesser than the corresponding value from the previous iteration, it is assumed that the C-component model is overfitting the data. Hence, the algorithm terminates and the best of the previous $(C-1)$-component models is returned as solution. If the algorithm reach $C = C_{max}$, the best C_{max}-component model is returned.

KDE

In traffic surveillance applications, e.g., surveillance of vessels at sea, trajectories typically follow traffic lanes that can be described as sequences of straight line segments. Adopting the multivariate Gaussian for the distribution of individual trajectory data points implies that the location and extension of the lane segments are characterised by a Gaussian mean vector and covariance matrix, respectively. Considering the distribution of data points along the minor axis perpendicular to the lane, the Gaussian may very well capture the vessel po-

Algorithm 4.1 Iterative EM for estimating GMM with unknown number of components

Input: Training data D_{Train}, validation data D_{Val}, maximum number of components C_{max}, number of candidate models per iteration q.

Output: Set of component parameters.

$C \leftarrow 1$
$Overfitting \leftarrow$ **false**
while $\neg Overfitting$ **and** $C \leq C_{max}$ **do**
  for $i \leftarrow 1$ to q **do**
    for $c \leftarrow 1$ to C **do**
      $\theta_c \leftarrow randomCompInitialisation\,()$
    end for
    $\{\hat{\theta}_1, \ldots, \hat{\theta}_C\}_i \leftarrow ExpectationMaximisation\big(\{\theta_1, \ldots, \theta_C\}, D_{Train}\big)$
  end for
  $j \leftarrow \underset{i=1,\ldots,q}{argmax}\left(E\Big[p\big(D_{Val}\big|\{\hat{\theta}_1, \ldots, \hat{\theta}_C\}_i\big)\Big]\right)$
  $\Theta_C \leftarrow \{\hat{\theta}_1, \ldots, \hat{\theta}_C\}_j$
  if $C \geq 2$ **and** $E\big[p(D_{Val}\,|\Theta_C)\big] \leq E\big[p(D_{Val}\,|\Theta_{C-1})\big]$ **then**
    $Overfitting \leftarrow$ **true**
    $\Theta \leftarrow \Theta_{C-1}$
  else
    $\Theta \leftarrow \Theta_C$
    $C \leftarrow C + 1$
  end if
end while
return Θ

sition offset relative the sea lane. However, assuming that the distribution of data points along the major axis of the lane is approximately uniform, one may argue that the Gaussian distribution is a suboptimal density model. This observation motivates why non-parametric techniques are also investigated in this thesis.

KDE (Section 2.1.2) is a technique for estimating an unknown probability density function, which, in contrast to GMM, is purely non-parametric, i.e., it makes no assumption regarding the parametric form of the true density; the form of the estimated PDF is explicitly determined by the training data. This property gives KDE an advantage over GMM regarding the ability to accurately model traffic lanes as discussed above. In fact, Ristic et al. (2008) proposed using adaptive KDE for estimating the joint PDF for the position-velocity vector of vessel trajectories (Section 2.2.3). Similar to other applications of KDE for anomaly detection, they proposed adopting a Gaussian kernel.

Considering Position and Velocity Vector

In many applications, such as sea and video surveillance, a feature space comprising current position, speed and course is considered (e.g., Ristic et al. (2008); Johansson and Falkman (2007); Brax et al. (2008)). Previously proposed algorithms typically model all these features by a joint high-dimensional PDF and calculates a single likelihood score for all features of a new data point (Ristic et al., 2008; Johansson and Falkman, 2007). Indeed, considering the joint likelihood does make sense under the assumption that there is a significant correlation between the features. However, a drawback of this approach is that it is not possible to infer whether a particular feature or a subset of features caused the anomaly, i.e., *which* of the features contributed (the most) to the anomaly. Moreover, a high-dimensional model may suffer from the *curse of dimensionality* (Mitchell, 1997), as more subtle, yet important, anomalies in lower dimensional subspaces may be marginalised.

It may be argued that the velocity vector is dependent on the current position. For example, the course and speed of a vessel is typically dependent on whether the vessel is located in a harbour (in which case speed is low or zero) or in a particular sea lane (in which case it follows the direction of the lane and lane speed regulations). The opposite direction of dependency, i.e., that position is dependent on current velocity vector, seems less intuitive (unless we also include previous position, in which case current position is dependent on the previous position and velocity vector).

Based on the reasoning above, we suggest an alternative approach to statistical point-based trajectory anomaly detection that involves estimating two distributions: $p(x, y)$, which corresponds to the unconditional position PDF, and $p(vx, vy \,|x, y)$, which corresponds to the PDF for the velocity vector conditioned on the current position. Assuming that $p(x, y)$ and $p(vx, vy, x, y)$ (the joint PDF for position and velocity vector) can be estimated using, e.g., GMM

or KDE as discussed above, we can express the conditional PDF for the velocity vector using Bayes' rule as follows:

$$p\left(vx, vy \,|x, y\right) = \frac{p\left(vx, vy, x, y\right)}{p\left(x, y\right)}. \tag{4.1}$$

For each of the two PDFs, we define an anomaly likelihood threshold; note that the likelihood score is not a normalised measure and, hence, two different thresholds need to be defined. Given a new trajectory data point (x', y', vx', vy'), we calculate the likelihood scores $p\left(x', y'\right)$ and $p\left(vx', vy' \,|x', y'\right)$ (based on Equation 4.1) and check if there is a position and/or velocity anomaly, respectively.

Cell-based Modelling

In some applications, the surveillance area may be relatively large. In the maritime domain, for example, the surveillance area may extend over several thousand kilometres of coastal area. This implies large amounts of trajectories and, consequently, a large and complex training set. As size and complexity of the training set grow, normalcy learning and anomaly detection may become computationally intractable. For GMM, the relatively large number of cluster components, which are required to adequately model local features of the data, will be scattered in high-dimensional feature space; this may pose serious constraints. In case of KDE, the computational complexity for calculating the adaptive window widths and anomaly detection is quadratic and linear, respectively, in the number of trajectory data points.

In this thesis, we propose the discretisation of the surveillance area into a grid in order to reduce model complexity, while still retaining local resolution. For each cell in the grid, a local model is estimated from the local training data. New data points from new trajectories are classified using the corresponding local model, i.e., the model is chosen depending on which cell the new data point lies within. This approach is known as *cell-based modelling*.

Assuming that complexity of each local model is bounded by, e.g., the maximum number of components allowed in a GMM, increasing and decreasing cell size correspond to decreasing and increasing model resolution, respectively. When determining a suitable cell size, it is important to find a good balance where resolution is sufficiently high for accurately capturing local features of data while cell size is large enough to include a fair amount of training data. In some applications, the geographical distribution and density of data may differ significantly from one area to another. Depending on how grid discretisation is done, some cells may have no or very small amounts of data. Obviously, if no data is available in a particular cell, no model can be estimated. More generally, if local data size is below a certain threshold, no model should be estimated, since it may be highly inaccurate; new data points located in such cells should rather be classified as anomalous by default.

Algorithm 4.2 Single Point Trajectory Nonconformity Measure (SPT-NCM)

Input: Set of trajectories $\{T_1, \ldots, T_n\}$, new trajectory feature vector $\mathbf{x}'$, nonconformity measure A.

Output: Nonconformity score α.

1: **for** $i \leftarrow 1$ to n **do**
2: $\quad \mathbf{x}''_i \leftarrow NearestNeighbour\,(T_i, \mathbf{x}')$
3: **end for**
4: $\alpha \leftarrow A\big(\{\mathbf{x}''_1, \ldots, \mathbf{x}''_n\}, \mathbf{x}'\big)$

4.3.2 Conformal Anomaly Detection Approach

The most straightforward way to adopt the CAD algorithm for the point-based feature model would be to let each data point $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, m$, from a trajectory $T = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be represented as a single example, i.e., $\mathbf{x}$ is substituted for z in CAD (Section 3.3.1). But this would result in a non-IID training set because of auto-correlation in trajectory data. Obviously, the current location of an object is strongly dependent on its previous location. Statistical methods, such as GMM and KDE methods discussed in Section 4.3.1, may not suffer severely from this if the data likelihood is interpreted as a general density measure. But a non-IID training set is undesirable in CAD, since the false alarm rate is then no longer guaranteed to be well-calibrated, i.e., close to ϵ (Theorem 3.1).

In order to eliminate this intra-trajectory dependency, we propose that each trajectory $T_1, \ldots, T_n$ should be treated as a single example in CAD, i.e., z_i corresponds to $T_i = \{\mathbf{x}_1, \ldots, \mathbf{x}_{m_i}\}$ for $i = 1, \ldots, n$, where n is the number of trajectories in the training set and m_i is the number of data points in trajectory T_i. Assuming that the trajectories are independent of each other, we now have a training set $\{T_1, \ldots, T_n\}$ that can be considered IID. But, in order to calculate the nonconformity score α_i for each trajectory T_i, we have to define a suitable NCM that compares two sets of feature vectors where set size may vary. One approach is to select and compare two feature vectors, $\mathbf{x}' \in T_i$ and $\mathbf{x}'' \in T_j$, from the two trajectories, T_i and T_j. For this we propose a specialised NCM called *Single Point Trajectory Nonconformity Measure* (SPT-NCM) (Algorithm 4.2). This function takes as input a set of trajectories $\{T_1, \ldots, T_n\}$ and a *reference feature vector* $\mathbf{x}'$ from a trajectory T for which a nonconformity score α is to be calculated and returned. Moreover, the function takes as input a regular NCM, A, which compares fixed length feature vectors $\mathbf{x} \in \mathbb{R}^d$, e.g., the NCM in Equation 3.2. The general procedure for SPT-NCM is as follows: For each trajectory T_i, $i = 1, \ldots, n$, determine the feature vector $\mathbf{x}''_i \in T_i$ that has the smallest ED to the reference vector $\mathbf{x}'$ in feature space. Next, calculate the nonconformity score α for $\mathbf{x}'$ relative the set $\{\mathbf{x}''_1, \ldots, \mathbf{x}''_n\}$ using A, where $\{\mathbf{x}''_1, \ldots, \mathbf{x}''_n\}$ is substituted for B and $\mathbf{x}'$ substituted for z (Section 2.3).

Algorithm 4.3 Single Point Trajectory Conformal Anomaly Detector (SPT-CAD)

Input: Nonconformity measure A, anomaly threshold ϵ, old trajectory examples $T_1, \ldots, T_{n-1}$ and new trajectory feature vector $\mathbf{x}$.
Output: Boolean variable $Anomaly$.

1: **for** $i \leftarrow 1$ to $n-1$ **do**
2: $\quad \mathbf{x}'_i \leftarrow NearestNeighbour\,(T_i, \mathbf{x})$ // Nearest neigbhour to $\mathbf{x}$ in set T_i
3: **end for**
4: $\mathbf{x}'_n \leftarrow \mathbf{x}$
5: $T_n \leftarrow \{\mathbf{x}\}$
6: $D \leftarrow \{T_1, \ldots, T_n\}$
7: **for** $i \leftarrow 1$ to n **do**
8: $\quad \alpha_i \leftarrow SPTNCM\big(\{D \setminus T_i\}, \mathbf{x}'_i, A\big)$
9: **end for**
10: $\tau \leftarrow U\,(0,1)$
11: $p_n \leftarrow \frac{|\{i:\alpha_i > \alpha_n\}| + \tau |\{i:\alpha_i = \alpha_n\}|}{n}$
12: **if** $p_n < \epsilon$ **then**
13: $\quad Anomaly \leftarrow$ **true**
14: **else**
15: $\quad Anomaly \leftarrow$ **false**
16: **end if**

Yet, it is not clear how the reference feature vector $\mathbf{x}'_i$ should be chosen for each trajectory T_i when calculating its nonconformity score α_i using SPTNCM. For the new trajectory T_n, the choice of $\mathbf{x}'_n$ is trivial; it is simply the new observed feature vector $\mathbf{x}$ that CAD should classify as normal or anomalous. For each of the remaining trajectories in the training set, we simply select the reference point $\mathbf{x}'_i \in T_i$ that has the shortest ED to $\mathbf{x}$ in the feature space; this corresponds to lines 1–3 of the Single Point Trajectory Conformal Anomaly Detector (SPT-CAD) (Algorithm 4.3).

4.4 Trajectory-based Anomaly Detection

Point-based approaches for trajectory anomaly detection are limited in the sense that they do not capture trajectory behaviour over time. Some point-based approaches integrate anomaly scores for individual data points within a sliding window (e.g., Johansson and Falkman (2007)); however, the feature model is still point-based and, thus, do not model the relationship between successive data points. To illustrate the shortcoming of a point-based feature model, consider Figure 4.1, which shows trajectories belonging to two main clusters. Obviously, one of the trajectories is an anomaly since it takes a different route. However, this route anomaly would hardly be detected by a point-based fea-

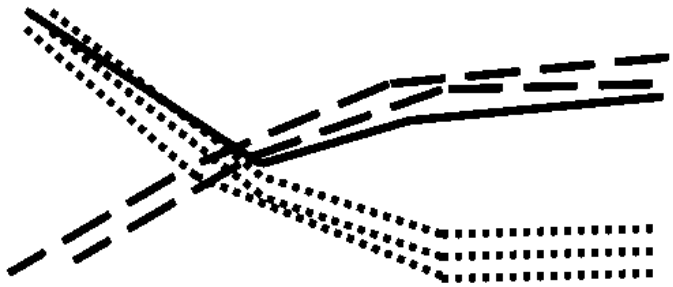

Figure 4.1: Illustration of trajectories from two trajectory clusters (dashed and dotted lines, respectively) and a single trajectory (thick line) corresponding to a deviating route.

ture model since no particular point along the trajectory is far away from the nearest data point of any of the other trajectories.

One approach to deal with this problem is to extend the point-based feature model and consider a high-dimensional feature space, where successive data points are embedded. Indeed, various methods for representing complete trajectories by a single high-dimensional feature vector have been proposed (Section 2.2). Yet, such methods typically require that the complete trajectory has been observed before it can be classified as normal or anomalous (Section 2.2). Hence, they are not appropriate for sequential anomaly detection in incomplete trajectories. A compromise between a point-based feature model and a feature model for complete trajectories is to consider a feature model for partial trajectories, i.e., segments of trajectories. An example of such method was proposed by Hu et al. (2006) where each trajectory is divided into a number of fixed length and non-overlapping subsequences of data points. Each subsequence is represented as a high-dimensional feature vector and could be modelled using, e.g., the point-based techniques described in Section 4.3.1 above. However, such a method typically involves more preprocessing, such as alignment and interpolation, and more parameters, such as size of the subsequence, than a point-based method. It may not be obvious how the size of the subsequence should be chosen. Moreover, there will still be a delay in anomaly detection that is bounded by the length of the subsequence.

Adopting SNN-CAD for trajectory anomaly detection, we would ideally have a dissimilarity measure S that measures the dissimilarity between two trajectories of arbitrary lengths, where one of the trajectories may be incomplete. The requirement of incomplete trajectories reflects the sequential anomaly detection setting in which SNN-CAD has to update the *preliminary* nonconformity score $\alpha_n^\star$ for the *incomplete* trajectory $T_n^\star = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$ and classify it as anomalous if the corresponding p-value drops below ϵ.

4.4.1 A Dissimilarity Measure for Incomplete Trajectories

Based on the observations above, we propose the use of directed HD (Section 2.4), $\overrightarrow{\delta_H}(T_i, T_j)$, for calculating the dissimilarity of an incomplete or complete trajectory T_i relative to a complete trajectory T_j, where trajectories are represented as polygonal curves in $\mathbb{R}^d$. This dissimilarity measure captures the extent to which T_i matches some part of T_j and has the advantage that trajectories are not required to be of equal lengths. Exact calculation of HD can be done efficiently using algorithms from computational geometry (Alt, 2009) (see Section 2.4). Moreover, we observe that the calculation can be done in a recursive manner:

$$\overrightarrow{\delta_H}\big((\mathbf{x}_1, \ldots, \mathbf{x}_m), T_j\big) = max\Big\{\overrightarrow{\delta_H}\big((\mathbf{x}_{m-1}, \mathbf{x}_m), T_j\big), \overrightarrow{\delta_H}\big((\mathbf{x}_1, \ldots, \mathbf{x}_{m-1}), T_j\big)\Big\}. \tag{4.2}$$

This recursive property makes the directed HD well-suited for sequential anomaly detection in trajectory data using SNN-CAD. Given the next data point x_l from the incomplete trajectory $T_n^\star = (\mathbf{x}_1, \ldots, \mathbf{x}_l)$, we can update the *preliminary* nonconformity score $\alpha_n^\star$ based on the updated distance $\overrightarrow{\delta_H}(T_n^\star, T_j)$ to every (complete) trajectory $T_j \in \mathbf{T}$ in the training set $\mathbf{T}$ using Equation 4.2. If the updated p-value drops below ϵ, we classify the incomplete trajectory $T_n^\star$ as anomalous. Since the directed HD monotonically increases with additional data points, i.e., $\overrightarrow{\delta_H}\big((\mathbf{x}_1, \ldots, \mathbf{x}_{l-1}), T\big) \leq \overrightarrow{\delta_H}\big((\mathbf{x}_1, \ldots, \mathbf{x}_{l-1}, \mathbf{x}_l), T\big)$ for $l = 2, 3, \ldots$, we know that $\alpha_n^\star \leq \alpha_n$, i.e., the preliminary nonconformity score $\alpha_n^\star$ is always less or equal to the nonconformity score α_n for the complete trajectory T_n. This property ensures that the probability of false alarm for SNN-CAD will still be equal to ϵ during sequential anomaly detection in trajectories. Note that this would not be the case if we used a distance measure based on, for example, the *average* distance to the closest point of the other trajectory.

4.4.2 A Dissimilarity Measure for Complete Trajectories

In applications other than sequential anomaly detection, it may be more appropriate to consider the undirected HD, δ_H, since it is symmetric and corresponds to a more complete comparison of two trajectories. Apart from anomaly detection in complete trajectories, this measure could be useful for, e.g., clustering trajectories.

4.4.3 Considering Location, Speed and Course

As discussed in Section 4.3.1, it may be appropriate to consider separate models for position and velocity vectors in order to suppress the curse of dimensionality and enhance understanding of which feature constitutes the anomaly. In case

of SNN-CAD based on HD, we suggest an anomaly detection scheme where three different feature subspaces of the trajectories are considered in parallel as follows:

Assume we are dealing with 4D trajectories:

$$T = \big((x_1, y_1, v_1, \theta_1), \ldots, (x_m, y_m, v_m, \theta_m)\big),$$

where (x_i, y_i), v_i and θ_i correspond to the location in the 2D-plane, speed and course, respectively, at some time point t_i. We propose the implementation of a separate SNN-CAD for the:

- 2D trajectories of location vectors, $T' = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$,
- 3D trajectories of location-speed vectors, $T'' = \big((x_1, y_1, v_1), \ldots, (x_m, y_m, v_m)\big)$,
- 3D trajectories of location-course vectors, $T''' = \big((x_1, y_1, \theta_1), \ldots, (x_m, y_m, \theta_m)\big)$.

For each new trajectory T_n, we let SNN-CAD calculate the three p-values p'_n, p''_n, p'''_n for the location, speed and course, respectively. If any of these is below the corresponding anomaly threshold ϵ', ϵ'' and ϵ''', the trajectory is considered anomalous with the respect to that feature. The sensitivity to location, speed and course anomalies can each independently be adjusted online by the corresponding anomaly threshold. Yet, if there is no reason to favour any particular type of anomaly, the thresholds should be set to the same level, i.e., $\epsilon = \epsilon' = \epsilon'' = \epsilon'''$.

When calculating directed or undirected HD between two trajectories T'_i and T'_j of location vectors, we use ED since this is the standard point metric (Alt, 2009). In particular, we consider the *squared* ED, since it requires less computation than ED (no square root) and because CAD (and Conformal predictors) are insensitive to monotonic transformations of the NCM (in this case SNN-NCM) (Shafer and Vovk, 2008). However, considering trajectories of heterogeneous features vectors, such as T'' and T''' above, the features have to be normalised before calculating ED, to avoid feature bias. A general way to normalise a feature value x is to replace it with the corresponding *standard score* x', also known as *z-score* (Han and Kamber, 2006):

$$x' = \frac{x - \mu}{\sigma}, \tag{4.3}$$

where μ and σ correspond to the population mean and standard deviation of the feature (which can be estimated from a sample of normal data). Thus, when calculating HD between two trajectories T''_i and T''_j of location-speed vectors, we use the squared ED for calculating distance between the normalised points $(x_1, y_1, v_1) \in T''_i$ and $(x_2, y_2, v_2) \in T''_j$:

$$dist_{x,y,v} = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (v_1 - v_2)^2 . \quad (4.4)$$

Calculating HD between two trajectories T_i''' and T_j''' of location-course vectors is less straightforward since course is measured in angular units. In this case, we use the point metric defined as the addition of the squared ED in normalised location space and the square of the absolute angle difference $d\theta \in [0, \pi]$ weighted by an application specific parameter w:

$$dist_{x,y,\theta} = (x_1 - x_2)^2 + (y_1 - y_2)^2 + (w * d\theta)^2 . \quad (4.5)$$

4.5 Discussion

The iterative EM algorithm (Algorithm 4.1), proposed for estimating the number components and the parameters of each component in GMM, is relatively straightforward to implement. However, it may be argued that it is relatively inefficient in terms of computational complexity. Indeed, other more efficient and greedy variants of EM, such as the algorithm proposed by Verbeek et al. (2003), can be used for estimating the optimal GMM when the number of components is unknown.

An advantage of KDE and SPT-CAD based on the nearest neighbour NCM is that they are, in contrast to GMM, purely non-parametric methods and, thus, are potentially more accurate for the position along traffic lanes. Yet, the flexibility of KDE and SPT-CAD comes at the cost of increased computational complexity. In order to mitigate the online complexity, offline preprocessing and clustering of trajectories prior to anomaly detection may be appropriate. As discussed in Section 4.3.1, the cell-based modelling reduces complexity. Another approach to mitigate the anomaly detection complexity is to cluster trajectories during offline learning based on the complete trajectory (Hu et al., 2006) or the trajectory origin (Ristic et al., 2008). When classifying a new data point, only data points from trajectories belonging to the same cluster will be used for density estimation (KDE) or nearest neighbour calculations (SPT-CAD).

It may be noted that the cell-based modelling approach is related to the LOF approach (Section 2.1.3), since both methods consider the *local* density of a data point during anomaly detection. However, a principal difference is that in the cell-based approach, the local neighbour (i.e., the boundaries of each cell) is static, whereas in LOF it is dynamically determined based on the k-nearest neighbour of the corresponding data point.

The point-based anomaly detection methods, presented in Section 4.3, address some of the issues discussed in the first section of this chapter: they are relatively parameter-light and easy to implement, and they support sequential anomaly detection in incomplete trajectories. We believe that they serve as a convenient base for initial investigations of sequential anomaly detection in trajectory data. Yet, the range of anomalous trajectory behaviour that can be

detected by these methods is limited to momentary kinematics of the objects. Assuming a feature model based on the current position, speed and course, anomalies that could potentially be detected include objects that:

- travel too fast in speed restricted areas,
- remain stationary in traffic lanes or in areas where stopping is prohibited,
- cross traffic lanes,
- travel in the opposite direction in one-way traffic lanes,
- are located in areas of (very) low traffic density.

Trajectory-based anomaly detectors, such as SNN-CAD based HD, can, in addition to the momentary anomalies presented above, also detect anomalous behaviour that evolves over time. Assuming a point feature model based on position, speed and course (see Section 4.4.3), examples of detectable behaviour includes various route anomalies and anomalous sequences of speed values. For example, highly fluctuating speed may be detected as an anomaly, even though the individual speed values are not anomalous by themselves.

Compared to other trajectory-based anomaly detection algorithms, we argue that SNN-CAD, with $S = \overrightarrow{\delta_H}$, has some principal properties that makes it more flexible and applicable in a wider range of applications, since it:

- does not require that trajectories are of equal length,
- requires no particular preprocessing of trajectories,
- supports sequential anomaly detection in incomplete trajectories,
- supports online learning,
- has relatively few parameters,
- has a principle way of setting the anomaly threshold, which does not make any assumptions on the nature or frequency of anomalies.

The property that trajectories need not be complete has in fact more general implications. It may be argued that the directed HD is not only robust to future data points that have not yet been observed; it also robust to previous data points that are missing due to delayed track initialisation, re-initialisation of a previous track that was lost, etc.

HD is by definition insensitive to the ordering of the data points. Thus, if only position is included in the point feature model, this may result in contra-intuitive matching. An example is two objects that follow the same path but travel in opposite direction. This can of course be addressed by simply extending the point feature model to also include current course, as discussed in Section 4.4.3. Another approach is to include relative time as an explicit feature; this would not only capture the direction but also the speed of the object.

4.6 Summary

In this chapter, we have proposed and discussed algorithms for sequential anomaly detection in trajectory data. Two main approaches were identified that differ in the nature of the feature model adopted: point-based and trajectory-based anomaly detection. For point-based anomaly detection, two types of algorithms were proposed. The first is cell-based statistical modelling of point features, using GMM or KDE. One novelty of this approach lies in the cell-based modelling, which is appropriate for wide-area surveillance applications. Another novelty is the alternative approach where output from a separate position model is combined with output from a position-conditional velocity vector model. The second type of point-based anomaly detector proposed is SPT-CAD, which is a modification of CAD based on SPT-NCM. For trajectory-based anomaly detection, we adopted SNN-CAD and proposed two parameter-free dissimilarity measures based on HD for comparing multi-dimensional trajectories of arbitrary length. One of these measures is appropriate for sequential anomaly detection in incomplete trajectories.

Chapter 5
Empirical Investigations

This chapter presents empirical investigations of the proposed algorithms for sequential anomaly detection in trajectory data. Four algorithms are evaluated: cell-based GMM and KDE (Section 4.3.1), SPT-CAD (Section 4.3.2) and SNN-CAD (Section 3.3.5) based on the HD (Section 4.4). The overall aim of the empirical investigations is to:

- Demonstrate the feasibility and validity of proposed models and algorithms on real world data sets.
- Identify suitable performance measures for algorithms for sequential anomaly detection in trajectory data.
- Evaluate proposed algorithms for sequential anomaly detection in trajectory data, according to identified performance measures.

The chapter begins with an introduction and discussion of the main performance measures identified. This is followed by an overview of all experiments including the data sets and algorithms evaluated in each experiment. The rest of the chapter presents a series of experiments organised according to the main data set used. The chapter is concluded by a summary of all experiments and general conclusions regarding the evaluated algorithms.

5.1 Performance Measures

Obviously, a central performance measure of any anomaly detector is its ability to accurately distinguish between normal and anomalous examples. Assuming that labelled examples from both classes are available, well-established classification performance measures from the field of pattern recognition can be used. In this thesis, we consider the classification performance measures *accur-*

acy, *precision*, *recall* and *false alarm rate (FAR)* according to their standard definitions in pattern recognition (Fawcett, 2006):

$$Accuracy = \frac{(tp + tn)}{(fp + fn + tp + tn)}, \tag{5.1}$$

$$Precision = \frac{tp}{(tp + fp)}, \tag{5.2}$$

$$Recall = \frac{tp}{(tp + fn)}, \tag{5.3}$$

$$FAR = \frac{fp}{(tn + fp)}, \tag{5.4}$$

where tp (true positives) corresponds to number of trajectories correctly classified as anomalous, fp (false positives) trajectories erroneously classified as anomalous, tn (true negatives) trajectories correctly classified as normal and fn (false negatives) trajectories erroneously classified as normal.

Accuracy is a general performance measure for evaluating binary classifiers (Tan et al., 2006); it has been used in the previous experiments that we reproduce in this thesis. Yet, accuracy is a somewhat blunt measure and is therefore often complemented, or replaced, by, e.g., precision and recall, which are more specific performance measures; these measures were also used in previous experiments that we reproduce in this thesis. Recall, also known as *sensitivity* (Fawcett, 2006), reflects sensitivity to labelled anomalies, i.e., ability to detect true anomalies. Precision corresponds to the degree to which detections are true anomalies and not false alarms. Ideally, precision and recall are both 1, which corresponds to perfect accuracy. In practise, this is typically not possible; instead there is a trade-off between precision and recall, which, in case of anomaly detection, is typically regulated by the anomaly threshold.

FAR corresponds to the rate of normal examples that are erroneously classified as anomalous. Analogously to precision, there is typically a trade-off between FAR and recall (sensitivity) that is regulated by the anomaly threshold; a higher threshold typically increases recall but also FAR. It may be argued that FAR is of more practical interest than precision in anomaly detection applications; the frequency of normal examples is typically much higher than the frequency of (true) anomalies and, hence, FAR dominates the overall alarm rate. Assuming that there is an upper bound for the overall alarm rate that can be tolerated by a human operator, FAR is of central importance and should be suppressed. This reasoning is consistent with the discussion by Axelsson (2000) regarding the *base rate fallacy* and its relation the performance of an intrusion (anomaly) detection system. According to Axelsson (2000), the *Bayesian detection rate*, which corresponds to precision, becomes dominated by FAR if the prior probability of an anomaly (in this case an intrusion) is very low. Therefore, "the factor limiting the performance of an [anomaly] detection system is

not the ability to correctly identify behaviour as [anomalous], but rather *its ability to suppress false alarms*". In other words, a very low FAR is needed to achieve a high level of precision. In contrast to precision, estimation of FAR only requires examples labelled normal, which usually exist in abundance. Moreover, in case of CAD, the specified anomaly threshold ϵ corresponds directly to FAR (Theorem 3.1), which therefore can be controlled in practise.

In a trajectory anomaly detection application, we may also be interested in the expected time, or number of data points, required before detection of a true anomaly. In this thesis, we therefore propose *detection delay* as a performance measure, which corresponds to the number of successive data points required from incomplete trajectories labelled as anomalous before they are classified as anomalous. As discussed in Section 2.2, a low detection delay may be an advantage in some applications since it enables earlier response to impending events and situations. Detection delay is a standard performance measure in the fields of sequential statistics and change detection (e.g., Ho and Wechsler (2010)); yet, as far as we know, it has not previously been proposed for evaluating algorithms for trajectory anomaly detection.

5.2 Overview of Experiments

This section presents an overview of the data sets used and the experiments carried out.

5.2.1 Data Sets

We use four main data sets: two non-public data sets with vessel trajectory data and two public data sets of synthetic trajectories and real video trajectories, respectively. The first data set of vessel trajectory data consists of unlabelled position-velocity vectors that were extracted from a database of recorded vessel traffic. The second set of vessel trajectory data consists of unlabelled trajectories extracted from another database of recorded vessel traffic. However, this data set has also been complemented with simulated trajectories labelled as anomalous. The synthetic data set, created by Piciarelli et al. (2008), consists of 2-dimensional trajectories labelled as normal or anomalous. The set of video trajectories, labelled normal or anomalous, were extracted from a real video recording (Pokrajac et al., 2007). More details regarding the data sets and how they are preprocessed are presented in the following sections.

5.2.2 Experiments

A number of experiments are carried out and organised according to the data sets used (see Table 5.1). Starting with Section 5.3, we evaluate cell-based GMM and the point-based position and velocity vector feature model for anomaly detection in unlabelled vessel traffic. The purpose of this experiment is to investig-

Table 5.1: Overview of the data sets and experiments (rows) and the algorithms (columns) evaluated.

Data set	Experiment	Cell-based GMM	Cell-based KDE	SPT-CAD	SNN-CAD
Vessel pos. vel. data	Anomaly detection in unlabelled data (Section 5.3)	X			
AIS vessel trajectories	Normalcy learning (Section 5.4.4)	X	X		
	Sequential detection delay – first exp. (Section 5.4.5)	X	X		
	Sequential detection delay – second exp. (Section 5.4.6)	X	X	X	
	Anomaly detection – precision and recall (Section 5.4.7)				X
	Anomaly detection – false alarm rate (Section 5.4.8)				X
Synthetic trajectories	Accuracy of trajectory outlier measure (Section 5.5.2)				X
	Online learning and sequential anomaly detection (Section 5.5.3)				X
Video trajectories	Accuracy of trajectory outlier measure (Section 5.6)				X

ate whether the proposed algorithm and feature model are feasible for anomaly detection in real vessel trajectories. Results are of qualitative nature, where we investigate the character of some of the most anomalous data points detected in an unlabelled test set.

Section 5.4 constitutes the main body of the empirical investigations. Here, performance of all the anomaly detection algorithms proposed in this thesis are evaluated using real vessel trajectories as normal data and simulated trajectories as anomalous data. In the first experiment, we compare accuracy of GMM and KDE for modelling the position and velocity vector features of vessel trajectory points. The main objective of this experiment is to investigate whether KDE is a more accurate normalcy model than GMM.

Next, sequential anomaly detection delay for cell-based GMM and cell-based KDE, based on different point feature models, are evaluated on a set of anomalous trajectory segments. The main objective of this experiment is to investigate whether:

- KDE is more sensitive to the anomalous trajectory points.
- How different point feature models affect detection delay.

This is followed by a second detection delay experiment, where cell-based GMM, cell-based KDE and SPT-CAD are evaluated. In contrast to the previous experiment, vessel class labels are exploited during learning and anomaly detection. The main objective of this experiment is to investigate whether:

- SPT-CAD is more sensitive to the anomalous trajectory points than cell-based GMM/KDE.
- The additional class information has any effect on anomaly detection performance.
- The false alarm rate of SPT-CAD is well-calibrated.

In the last two experiments of Section 5.4, we evaluate SNN-CAD based on directed HD. The first of these experiments is a reproduced experiment previously published by Brax et al. (2010), where precision and recall for different anomaly detectors are evaluated. In the other experiment, the false alarm rate of SNN-CAD during online learning and anomaly detection is investigated. The purpose of the experiments is to investigate the classification performance SNN-CAD, and to confirm that its false alarm rate is well-calibrated.

In Section 5.5, we further investigate the performance of SNN-CAD on the public set of synthetic labelled trajectories. We first reproduce a previously published experiment, where accuracy for SNN-CAD with $S = \delta_H$ and $S = \overrightarrow{\delta_H}$ on complete trajectories is compared to previously published results for other outlier measures. We then evaluate SNN-CAD for online learning and sequential anomaly detection in the trajectories, with the objective to:

- Demonstrate that labelled anomalies can be detected with high sensitivity and low FAR before the complete trajectory has been observed.
- Show that FAR is well-calibrated, i.e., close to the specified anomaly threshold ϵ, and that sensitivity to true anomalies increases during online learning.

Finally, in Section 5.6 we reproduce a previously published experiment on a public set of labelled video trajectories. In this experiment, classification accuracy of SNN-CAD with $S = \overrightarrow{\delta_H}$ on complete trajectories is compared to previously published results.

5.3 Anomaly Detection in Unlabelled Vessel Position-Velocity Data

A first experiment with cell-based GMM was carried out using a data set of position-velocity vectors extracted from a database of recorded vessel tracks. The aim of this experiment was to investigate whether the proposed algorithm and the point-based position-velocity feature model are feasible for anomaly detection in sea traffic. Since there are no labelled anomalies in this data set, we investigate the character of the most anomalous data points detected by the algorithm, i.e., the data points with the lowest likelihood scores.

5.3.1 Data Description and Preprocessing

Three sets of unlabelled vessel position-velocity vectors were extracted from a vessel track database provided by the Swedish Naval Base[1]. The database contains fused AIS data and radar tracks from vessels, which have been collected during surveillance of the coast of south Sweden and parts of the neighbouring coasts of Denmark, Germany and Poland and the sea in between. The database has two main parts corresponding to approximately one week of summer traffic and one week of autumn traffic, respectively. In addition, there are six shorter recordings spread out over the year, each having a duration of a few hours. In its raw form, each row in the database corresponds to a vessel report that includes a number of attributes. For each row, we extracted a 4-dimensional position-velocity vector from attributes corresponding to location in latitude and longitude, speed and course. The feature vectors where organised into three different sets corresponding to the two main parts and the remaining six scenarios.

5.3.2 Experiment Design

The grid size of the cell-based GMM was arbitrarily set to 30 in latitudinal direction and 40 in longitudinal direction. This was considered to be good enough

[1] http://www.forsvarsmakten.se/en/Organisation/Training-units/Naval-Base-MarinB/

in the sense that the number of data points in each square is sufficiently high (more than 30 data points), while the complexity of each GMM is not too high (less than 20 components). Estimation of the GMM parameters was done using a publicly available implementation of the greedy EM algorithm, with standard parameter values, proposed by Verbeek et al. (2003). During training of each local GMM, the local data from the summer traffic and autumn traffic data sets (Section 5.3.1) were used for parameter estimation and model validation, respectively; recall that the greedy EM algorithm requires a separate validation set to determine a suitable number of mixture components (Verbeek et al., 2003). The maximum number of mixture components was set to 20; this value was never reached, however. For more details on the greedy EM algorithm and its parameters, the reader is referred to Verbeek et al. (2003). The validation set was also used for tuning the anomaly likelihood threshold so that 0.1% of all position-velocity vectors in the validation set were classified as anomalous. The level 0.1% was chosen in consideration of what may constitutes an acceptable (false) alarm rate in a real surveillance applications. The remaining data from the six shorter scenarios was used as test data for anomaly detection.

5.3.3 Results

In total, 0.1% of the position-velocity vectors of the test set were classified as anomalous. Figure 5.1–5.4 present qualitative results for a subset of all cells, where anomalies were detected. Each arrow in the figures represents a position-velocity data point, where the position, size and direction of the arrow correspond to the vessel's location, speed and course, respectively. The colour coding is as follows:

- Blue corresponds to the training data.
- Green corresponds to test data classified as normal.
- Red corresponds to the test data classified as anomalous.

5.3.4 Analysis

The rate of data points in the test set classified as anomalous (0.1%) is equal to the rate of anomalous data points in the validation set, i.e., the data set used for tuning the anomaly threshold. This result is quite expected since the validation set and test set are both assumed to reflect normalcy and, hence, should have approximately the same of amount of anomalies for a given threshold.

Examining Figure 5.1–5.4, we see that the detected anomalies correspond to vessels that are: crossing sea lanes (Figure 5.1, Figure 5.2 and Figure 5.4), travelling in the opposite direction of sea lanes (Figure 5.2) and travelling relatively fast (Figure 5.3 and Figure 5.4). These anomalies appear rather clear in

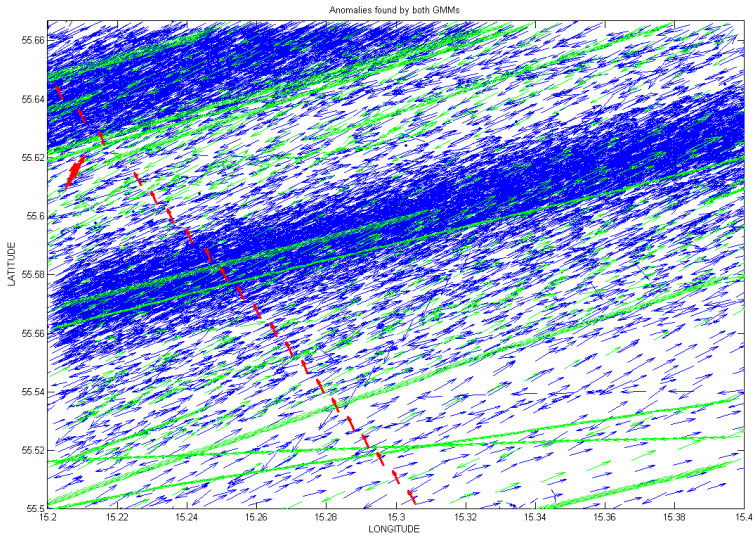

Figure 5.1: Anomaly detection results for cell-based GMM and the joint position-velocity feature model, in a selected cell located approximately 19 nautical miles north-east of the island of Bornholm, Denmark. A two-directed sea lane is clearly illustrated by the high concentration of (blue) traffic in two opposite directions. A minority of the traffic travels in parallel to the main sea lane, but still follows the main direction of motion. The anomaly (red) corresponds to a vessel crossing the sea lane.

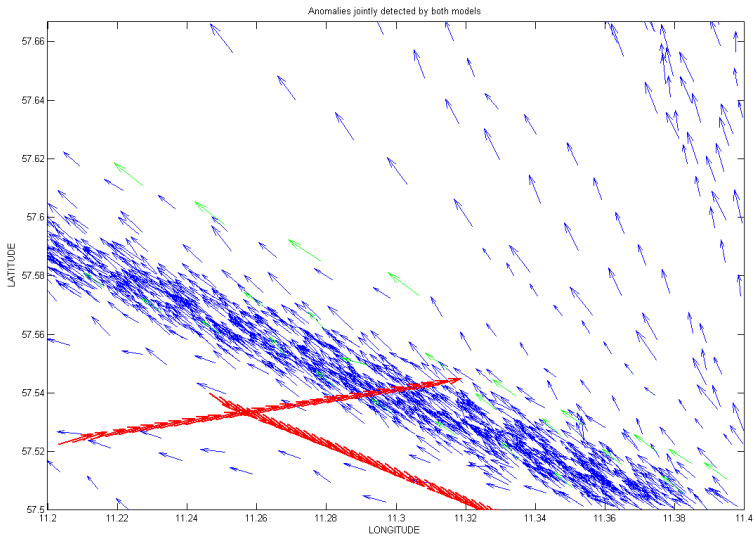

Figure 5.2: Anomaly detection results for cell-based GMM and the joint position-velocity feature model, in a selected cell located approximately 15 nautical miles west of the port of Gothenburg, Sweden. The majority of the tracks of the training data (blue) are concentrated to the sea lane going in north-west direction; a minority of the training data deviates from it. Data points from two vessel tracks are detected as anomalous; the first vessel has travelled close to the sea lane but in the opposite direction, and the other has crossed the sea lane.

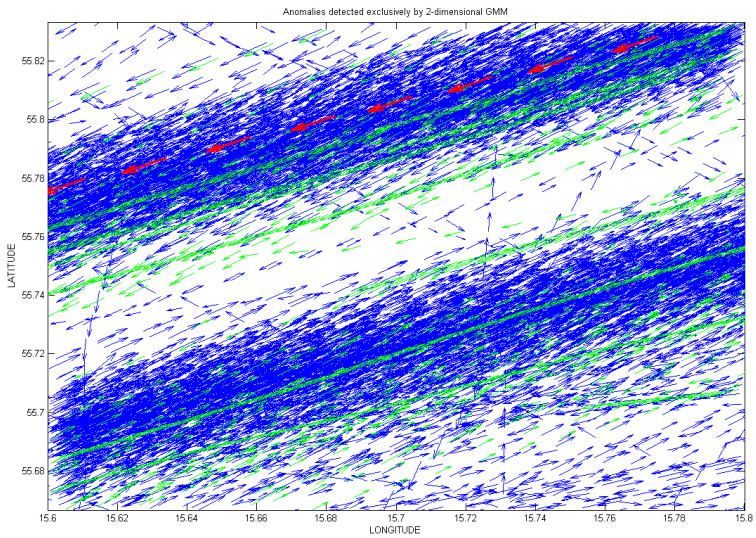

Figure 5.3: Anomaly detection results for cell-based GMM and the joint position-velocity feature model, in a selected cell located approximately 20 nautical miles south of the port of Karlskrona, Sweden. A two-way sea lane is clearly illustrated. The anomalous data points correspond to a vessel travelling in the sea lane but at high speed, which is indicated by the length of the velocity vector.

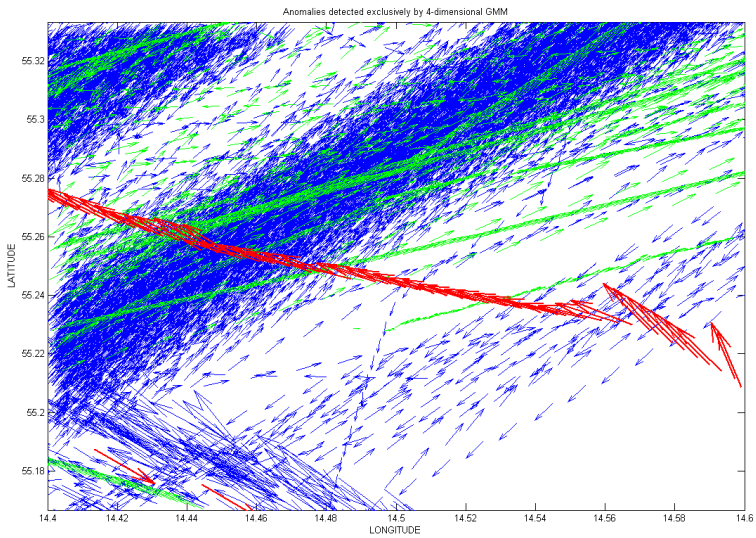

Figure 5.4: Anomaly detection results for cell-based GMM and the joint position-velocity feature model, in a selected cell located in the strait between the island of Bornholm, Denmark, and the south-eastern coast of Sweden. The anomalous data points (red) correspond to a vessel crossing the main sea lane at an anomalous location and at a relatively high speed.

contrast to the normal data. The location and extension of the major sea lanes are often easily identified. However, information regarding the normal speed in the sea lanes is obscured by the high concentration of plotted data vectors. Figure 5.4 confirms that the algorithm is sensitive to anomalous correlations between different feature values. There is a substantial amount of tracks in the training set that crosses the main sea lane; however, the location of the crossing point is different for the test track detected as anomalous, suggesting that the algorithm is sensitive to the relation between current course and position.

The point-based position-velocity feature model is of course limited, since it only considers momentary states of the vessel motion. Furthermore, spatio-temporal relations to surrounding vessels are not considered. This implies that anomalies related to situations that develop over time and which may involve multiple vessels, e.g., smuggling and hijacking, are difficult, if not impossible, to detect. However, the simplicity of the method makes it applicable to any domain involving motion in the two-dimensional plane, requiring no particular domain knowledge.

5.3.5 Summary and Conclusion

In this experiment we have evaluated the cell-based GMM and the position-velocity feature model for anomaly detection in recorded vessel traffic. Results indicate that the proposed algorithm and feature model are feasible for anomaly detection in sea traffic. It has been shown that vessels crossing sea lanes, violating traffic direction and travelling at relatively high speeds can be detected at a low alarm rate (0.1%). Moreover, it has been shown that the proposed algorithm is sensitive to anomalous combinations of the feature values, e.g., vessels crossing sea lanes at novel locations. Yet, anomalies related to behaviour over time are not captured by the method. Hence, the method should be replaced or complemented by a trajectory-based anomaly detector, e.g., SNN-CAD based on HD (Section 4.4).

5.4 Anomaly Detection in Labelled Vessel Trajectory Data

In this section, a series of experiments are carried out where relative performance of cell-based GMM and KDE, SPT-CAD and SNN-CAD based on HD are evaluated on labelled vessel trajectory data. We start by describing the data sets used, including how they were extracted and preprocessed (Section 5.4.1 and 5.4.2). The general setup and the parameters of the evaluated algorithms, used throughout the experiments, are presented in Section 5.4.3. In the first experiment (Section 5.4.4), we compare normalcy learning performance of GMM and KDE. In Section 5.4.5 and 5.4.6, we investigate the detection delay for cell-based GMM and KDE and SPT-CAD. In Section 5.4.7, we reproduce a previ-

ously published experiment where classification performance of SNN-CAD is compared to other algorithms. Finally, in Section 5.4.8 we investigate whether the false alarm rate of SNN-CAD is well-calibrated (according to the definition in Section 3.3.1).

5.4.1 Extraction of Normal Training Data

Vessel trajectories were extracted from an AIS database provided by Saab Transponder Tech. The database includes AIS reports collected from about three weeks of continuous sea traffic along the west coast of Sweden. From each of the raw AIS reports, we extracted the following attributes: MMSI (Maritime Mobile Service Identity), latitudinal and longitudinal position, course, speed, absolute time stamp and vessel class. In order to extract trajectories, we have tracked each individual vessel based on its MMSI number, which is assumed to be an unique identifier. To reduce the size of the data, vessel reports are sampled when the tracked vessel has travelled a distance equal to or larger than the sampling distance 200 m. The sampling distance is chosen quite arbitrary but is considered to sufficiently reduce the size of the data while still retaining a reasonable resolution. During each sampling, latitude, longitude, speed and course of the current vessel report are extracted, resulting in 4-dimensional trajectories. Sampling is continued until either no new reports are received within a particular time interval (e.g., due to the fact that the vessel has left AIS-coverage), or it has remained stationary for more than 5 minutes, in which case it is assumed to be moored. In either case, the current trajectory is terminated and a new trajectory is initiated when the vessel reappears or starts to move again.

In total, approximately 4,500,000 vessel reports were sampled and 36,370 trajectories extracted from the AIS database. These trajectories constitute the *total* AIS vessel trajectory set and are illustrated in Figure 5.5. Moreover, two local sets of 2888 and 1468 trajectories were separately extracted from a confined area outside the port of Gothenburg (Figure 5.6 and Figure 5.7). Trajectories in the second local set are labelled by the corresponding AIS class and belong to one of the three vessel classes *cargo ship*, *passenger ship* or *tanker*; these are the most common classes and constitute approximately 50% of all the trajectories in this area. From the local sets, a subset of 2310 and 1028 trajectories were randomly sampled as training set $\mathbf{T}_{Train1}$ and $\mathbf{T}_{Train2}$, respectively. The remaining trajectories, 578 and 440 trajectories, respectively, were used when creating three test sets of normal and simulated anomalous trajectories, as described in Section 5.4.2 below.

Note that trajectories extracted from the AIS database are unlabelled in the sense that there is no label telling whether a particular report or trajectory is anomalous or normal. However, during the experiments we assume that all the trajectories extracted from the AIS database are normal.

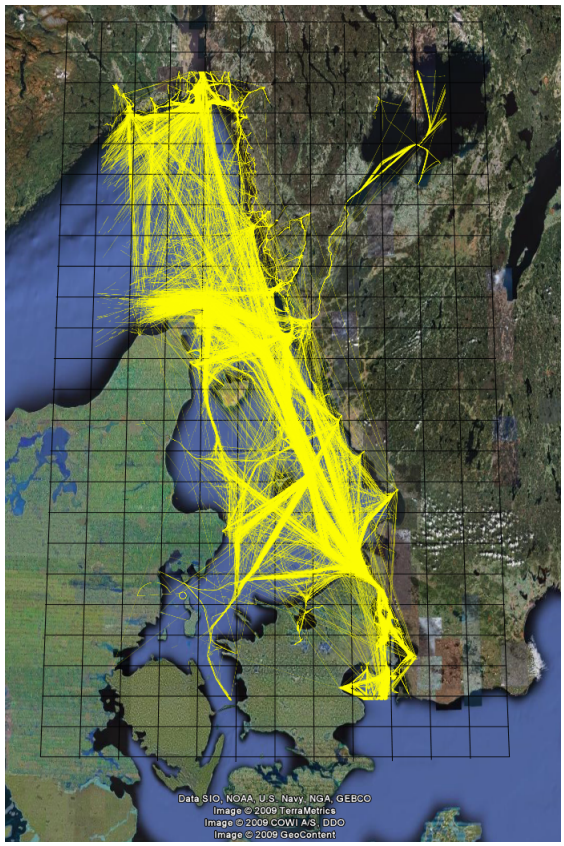

Figure 5.5: Screen capture from Google Earth where all trajectories extracted from the AIS database (Section 5.4.1) have been superimposed on a photo (copyright 2009 TerraMetrics, COWI A/S, DDO, GeoContent) of Sweden (right) and Denmark (left) and the sea in between (Kattegat, Skagerrak and Öresund).

Figure 5.6: Screen capture from Google Earth where the local set of 2888 trajectories from all vessel classes (Section 5.4.1) have been superimposed on a photo (copyright 2009 Lantmäteriet/Metria) of the port area of Gothenburg. The size of the area is approximately 12×8 km. A grid of size 6×4, used for cell-based modelling (Sections 5.4.5 and 5.4.6), is overlaid.

Figure 5.7: Screen capture from Google Earth where the local set of 1468 trajectories from vessel classes *cargo ship*, *passenger ship* and *tanker* (Section 5.4.1) have been superimposed on a photo (copyright 2009 Lantmäteriet/Metria) of the port area of Gothenburg. The size of the area is approximately 12×8 km. A grid of size 6×4, used for cell-based modelling (Sections 5.4.5 and 5.4.6), is overlaid.

5.4.2 Creation of Normal and Anomalous Test Data

Three different test sets, $\mathbf{T}_{Test1}$, $\mathbf{T}_{Test2}$ and $\mathbf{T}_{Test3}$, were created for the experiments. Each trajectory in the first and second test set consists of an initial segment labelled normal, which is followed by a simulated segment labelled as anomalous. The trajectories in the third test set are simply normal or anomalous, where anomalous trajectories are based on normal trajectories that have been distorted. More details on the generation of the test sets follow below.

First Test Set of Vessel Trajectories

The first test set, $\mathbf{T}_{Test1}$, consists of 1000 trajectories that each has two parts: an initial *normal segment* that is followed by an *anomalous segment*. The normal segment was constructed by first randomly sampling a trajectory among the subset of 578 trajectories from all vessel classes extracted from the port area of Gothenburg. A break point along the sampled trajectory was then randomly selected; the part that extended from the first data point to the breakpoint was extracted as the normal segment. The anomalous segment, extending from the selected breakpoint, is a *random walk* generated as follows. Starting from the breakpoint, new values for speed and course are randomly sampled from the uniform distributions on the intervals 0–30 knots and 0–360°, respectively. Next, the subsequent latitudinal and longitudinal coordinates of the trajectory are updated by projecting the next position 200 m ahead of the current position, based on the current course. Thus, the sampling distance of the anomalous segments is equal to that of the normal segments (see Section 5.4.1). Moreover, during each trajectory update (sampling), there is a 10% probability that new values for the speed and course are sampled from the corresponding uniform distributions, independently of each other. If the trajectory is about to leave the local port area (Figure 5.6), a new course is sampled that ensures that the trajectory does not leave the area. The length of each anomalous segment is fixed to 100 points, which was assumed to be long enough for enabling anomaly detection. A sample of trajectories from the first test set are illustrated in Figure 5.8.

Second Test Set of Vessel Trajectories

The second test set, $\mathbf{T}_{Test2}$, was created in the same way as $\mathbf{T}_{Test1}$ above, but with the difference that normal segments were sampled from the subset of 440 trajectories where AIS vessel class is cargo ship, passenger ship or tanker.

Third Test Set of Vessel Trajectories

The third test set, $\mathbf{T}_{Test3}$, was created by Brax et al. (2010). It consists of 161 and 172 trajectories labelled as normal and anomalous, respectively. Both the normal and anomalous trajectories were randomly sampled from the subset of

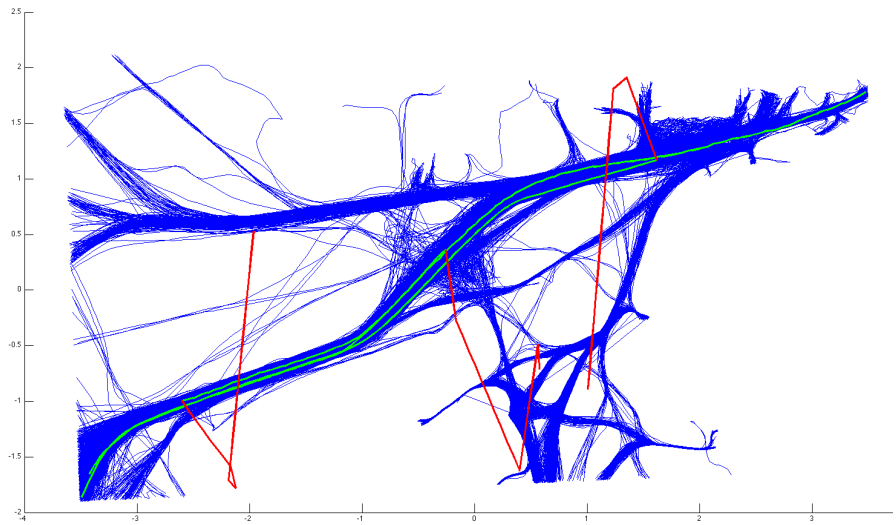

Figure 5.8: Plot of three semi-simulated vessel trajectories from the first test set $\mathbf{T}_{Test1}$, where the (real) normal segments are green and the (simulated) anomalous segments are red (Section 5.4.2). These are overlaid on the set of all (real) normal trajectories in blue from the first training set $\mathbf{T}_{Train1}$ (2310 trajectories) (Section 5.4.1). Note that only the first 20 points from the anomalous segments have been plotted for clarity.

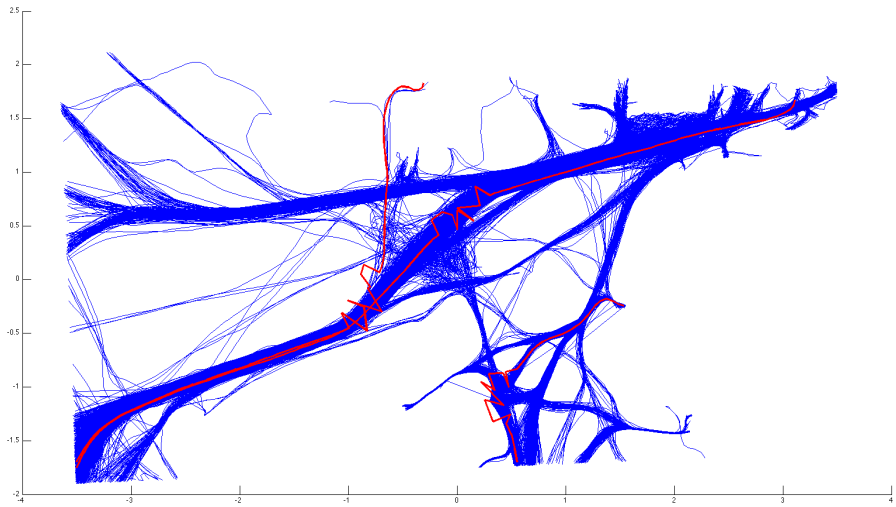

Figure 5.9: Plot of three semi-simulated vessel trajectories labelled anomalous (thick red) from the third test set $\mathbf{T}_{Test3}$ (Section 5.4.2), overlaid on the set of all normal trajectories (blue) from the first training set $\mathbf{T}_{Train1}$ (2310 trajectories) (Section 5.4.1). Note the anomalous segment of each anomalous trajectory, where 10 consecutive points have been shifted.

578 trajectories extracted from the Gothenburg port area. However, each trajectory labelled as anomalous was distorted by randomly selecting a point and shifting it and the 9 consecutive points. The latitude and longitude location of each point was shifted a random distance from its original location according to the uniform distribution on $[-\triangle, \triangle]$, where $\triangle = 500$ m. A sample of the anomalous trajectories are illustrated in Figure 5.9. For more information regarding the generation of this data set, see Brax et al. (2010).

5.4.3 General Setup and Parameters

This section describes the setup, parameters and other implementation details of the anomaly detection algorithms that are evaluated on the AIS data set.

Cell-based GMM

For the cell-based GMM, the iterative EM algorithm (Algorithm 4.1), proposed in Section 4.3.1, has been used for estimating the parameters, including the number of components. The number of candidate models per iteration, q, was set to 5, which was assumed to be enough for adequately suppressing the sensitivity to the random parameter initialisation. The maximum number of components, k_{max}, was unlimited. Moreover, 20% of the training data was randomly

and exclusively selected for the hold-out model validation, i.e., determining the appropriate number of mixture components. This proportion was chosen quite arbitrary, but was assumed to be enough for model validation.

Cell-based KDE

For KDE, we adopted the adaptive window width approach described by Ristic et al. (2008). That is, we selected the multivariate Gaussian kernel with zero mean and fixed covariance matrix, which was set to the sample covariance of the local cell data. The window width h for each kernel was adaptively estimated using the formula originally proposed by Silverman (1986).

SPT-CAD

For SPT-CAD, we adopted the position-velocity vector feature model, where each feature was normalised to $[-1, 1]$. The modified nearest neighbour NCM (Equation 3.2) was used for calculating the nonconformity between pairs of trajectory points (input parameter A to Algorithm 4.2), where the number of nearest neighbours, k, was set to one, which is common for nearest neighbour methods.

SNN-CAD

For SNN-CAD, we used the directed HD as similarity measure, i.e., $S = \overrightarrow{\delta_H}$, considering the single most similar neighbour, i.e., $k = 1$. We adopted the anomaly detection scheme proposed in Section 4.4.3, where three different point feature subspaces were considered. That is, we had three SNN-CAD detectors in parallel, classifying the trajectories of position vectors, position-speed vectors and position-course vectors, respectively, where normalisation of features was done according to Section 4.4.3.

For practical reasons, we have not implemented an exact algorithm for calculating HD between two trajectories, i.e., an algorithm that considers all intermediate points along the line segments, such as the one proposed by Alt et al. (1995). Instead, we use a naive algorithm that approximates HD between two trajectories by only considering the finite set of end points of the line segments.

5.4.4 Normalcy Learning – GMM vs. KDE

GMM and KDE are two statistical models that have been proposed for point-based trajectory anomaly detection (Section 4.3.1). As discussed in Section 4.3.1, there are reasons to believe that GMM is suboptimal for estimating the distribution of the position along traffic lanes; the position distribution along traffic lanes is usually close to uniform rather than Gaussian. In this section, we empirically investigate whether KDE is indeed a more accurate model than

GMM for estimating the distribution of position-velocity values of normal vessel traffic.

Ideally, we would measure the similarity of the estimated distributions with the true distribution using, e.g., the Kullback-Leibler divergence measure (Bishop, 2006). However, this is not feasible since we do not know the true distribution for the normal data. Therefore, we propose a new performance measure known as *normalcy modelling performance*. Assume that we have a normal training set and a normal evaluation set that are IID. Having estimated the underlying distribution using the training set, we calculate the data likelihood for the evaluation set using each of the estimated models. The normalcy modelling performance of a model is proportional to the evaluation data likelihood. If the model is close to the true distribution, we expect that the data likelihood will be relatively high, since the evaluation data was generated from the same distribution as the training set. Hence, the model (GMM or KDE) that assigns the largest likelihood for the evaluation data is assumed to better estimate the true PDF and is therefore regarded as superior.

Experiment Design

We estimated joint position-velocity vector models for the total AIS vessel trajectory set (36 370 trajectories, see Section 5.4.1), using cell-based GMM and cell-based KDE. A grid of size 12×24 in longitude and latitude direction, respectively, was used, where the size of the cells is approximately 22×22 km. The grid is different from that of the previous experiment (Section 5.3); the extension of the trajectories in this data set is different, and cell size was (arbitrary) chosen to be slightly larger. For each cell, we extract a local trajectory segment from each trajectory that passes the cell. The local trajectory segments for each cell are then divided into two sets: 80% are randomly selected as local training set and used for PDF estimation, while the remaining are used as local test data for calculating likelihood scores. Cells with less than 100 trajectory data points in the evaluation set were excluded from normalcy learning. This resulted in 112 out of 288 cells having enough data for estimating normalcy models.

Results

Results for the normalcy modelling experiment using cell-based GMM and KDE are summarised in Table 5.2. The table presents the median and first percentile of the distribution of the natural logarithm of the likelihood for all data points from all the local evaluation sets. For a selected cell (Figure 5.10), the 2-dimensional PDFs in position space for GMM and KDE are illustrated in Figure 5.11 and Figure 5.12, respectively.

Table 5.2: Summary of the distribution of natural logarithm of the likelihood scores for all vessel position-velocity data points of all the local evaluation sets (Section 5.4.4).

	Cell-based GMM	Cell-based KDE
Median of log likelihood distribution	-1.8270	-1.3162
1st percentile of log likelihood distribution	-11.6569	-7.7414

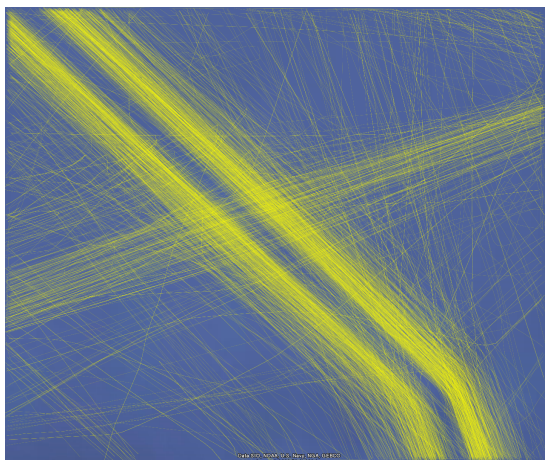

Figure 5.10: Plot of vessel trajectories from the AIS data set (Section 5.4.1) in a selected cell from the grid of size 12×24 (Section 5.4.4). This cell has been selected, since the corresponding GMM and KDE models have interesting features (see Figure 5.11 and 5.12).

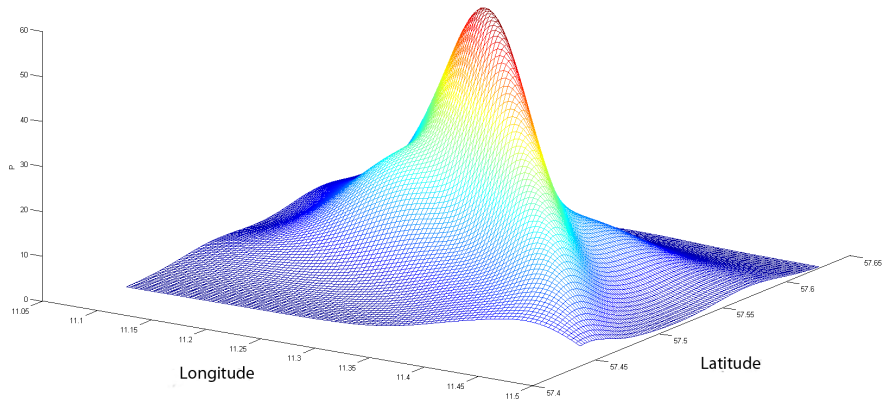

Figure 5.11: Visualisation of PDF in position space for GMM for the selected cell in Figure 5.10. Note the uni-modal peak halfway along the two parallel sea lanes, hiding the separation between them.

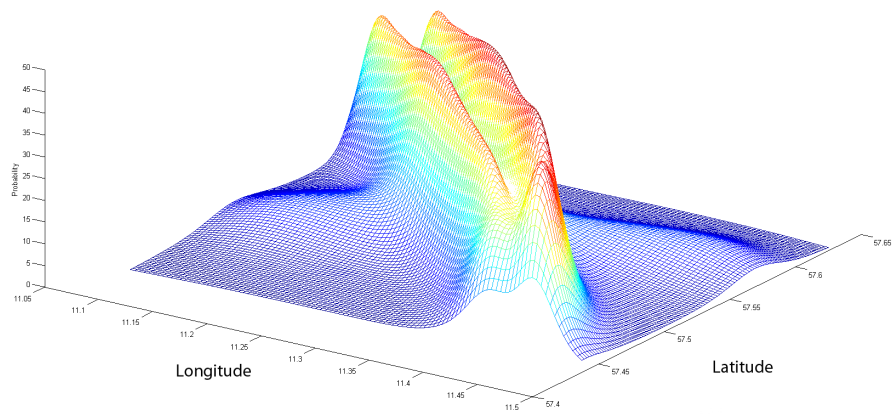

Figure 5.12: Visualisation of PDF in position space for KDE for the selected cell in Figure 5.10. Note how this PDF, in contrast to the one in Figure 5.11, nicely discriminates the two parallel lanes and approximates a uniform distribution along them.

Analysis

Looking at Table 5.2, we may interpret the results as KDE being superior in the sense that the median of the data likelihood of a normal observation is higher than for the GMM model; the median of the likelihood (no logarithm) for the normal evaluation set is approximately 5/3 times the corresponding likelihood for the GMM model. This ratio is even larger, approximately 50:1 in favour of KDE, when considering the first percentile, i.e., the least likely observations encountered in the evaluation set. These results, together with the plotted PDFs in Figure 5.11 and Figure 5.12, support the hypothesis that KDE more accurately captures features in normal data related to, e.g., the distribution of vessel position along sea lanes.

One could argue that the assumption that the recorded AIS data strictly reflects normal traffic is not realistic or feasible. In fact, there may be occurrences in this data that some people, in some context, would consider anomalous and worthy an alert. By taking the median and not the mean value of the (logarithm of the) likelihoods for all data points, our assessment of the normalcy modelling performance is robust with regard to such anomalies. Consider for example the case where we actually have a true anomaly in our evaluation data. A good normalcy model would then assign this data point a (very) low likelihood, while a less good normalcy model might assign it a considerably larger likelihood. Thus, considering the mean instead of median would penalise models that detect these true anomalies in favour for models that do not.

5.4.5 Sequential Anomaly Detection Delay – First Experiment

In this section, we investigate detection delay on the anomalous trajectory segments of the first test set $\mathbf{T}_{Test1}$. We evaluate cell-based GMM and KDE where different point feature models are estimated from the first training set $\mathbf{T}_{Train1}$. The main objective of the experiment is to investigate:

- Whether KDE is more sensitive to the anomalous segments than GMM.
- How different point feature models affect detection delay.

Experiment Design

Grid size was arbitrary set to 6×4 cells in longitude and latitude direction, respectively. The resulting cells were approximately quadratic with size 2 km, which is less than the cell size of the grids used in the previous experiments. The reason for considering another grid, with smaller cell size, is that this experiment involves a subset of the data set from a port area, where traffic density is relatively high. Based on the first training set, $\mathbf{T}_{Train1}$, we estimated, for each cell, three different PDFs based on GMM/KDE: the joint 4D position-velocity vector PDF, the 2D position PDF and the 2D conditional velocity vector PDF

Table 5.3: Sequential anomaly detection delay (Section 5.1) on the anomalous segments of the first test set of vessel trajectories $\mathbf{T}_{Test1}$ (Section 5.4.5). The anomaly threshold of each detector has been tuned so that 1% of the normal trajectory segments in $\mathbf{T}_{Test1}$ have one or more data points classified as anomalous.

Anomaly detector		Mean	Median
Cell-based GMM	Joint position-velocity PDF	18.7	12
	Position PDF	11.3	7.5
	Cond. velocity PDF	22.3	16
	Position PDF *OR* cond. velocity PDF	17.7	11
Cell-based KDE	Joint position-velocity PDF	17.9	12
	Position PDF	10.1	7
	Cond. velocity PDF	26.0	19
	Position PDF *OR* cond. velocity PDF	10.6	7

(conditioned on current position) (see Section 4.3.1). Based on these PDFs, four types of cell-based anomaly detectors were evaluated:

1. Joint position-velocity detector.
2. Position detector.
3. Position-conditional velocity vector detector.
4. Logical *OR* combination of detectors 2 and 3 above.

In order to normalise detection delay results, the anomaly threshold for each detector 1–4 was tuned so that one or more anomalous data points were detected in 1% of the normal segments in the test set $\mathbf{T}_{Test1}$. In case of the combined detector of type 4, the thresholds for detector 2 and 3 were each independently tuned to generate anomalous data points in 0.5% of the normal segments; this ensures that the false alarm rate of the combined detector is bounded by 1%, which we assume is an acceptable rate in a real surveillance application.

Results for detection delay are presented in Table 5.3.

Analysis

Examining Table 5.3, a number of interesting observations can be made. To start with, there seems to be no strong evidence suggesting that KDE is more sensitive to anomalous trajectory points than GMM (or vice verse). This is perhaps a bit surprising, given the prior hypothesis and results from the normalcy modelling experiment (Section 5.4.4). However, there is a possible explanation for the apparent insignificance of GMM and KDE which we will get back to later in this section.

The best results for cell-based GMM/KDE were achieved when considering only the position PDF; the detection delay for GMM/KDE based on only velocity vector is more than twice than that of the corresponding position detector. Hence, it seems that the position is a significantly more accurate feature than the velocity vector for discriminating the anomalous data points, regardless of GMM/KDE. Moreover, performance of the joint position-velocity detector and the combined position and velocity detectors were both worse than the position detector alone. Thus, it seems that for most of the anomalous test segments, the velocity vector features do not contribute to the anomaly detection. It is also interesting to note that the logical OR combination of the position detector and the conditional velocity vector detector performs better than the joint position-velocity detector, in particular for KDE. A possible explanation for this difference in performance is the curse of dimensionality, as discussed in Section 4.3.1; more subtle anomalies in lower-dimensional feature spaces, such as position, are marginalised in the joint high-dimensional feature space.

Generally, we observe that the mean value is larger than the corresponding median value for all detectors. This means that the distribution of the detection delay is skewed to the right, i.e., there is a long tail of relatively large delay values. This could at least partially be explained by those anomalous trajectory segments that, initially, more or less follow the normal sea lanes and lie within the normal speed interval by chance. The detection of these anomalies will typically be delayed until a new course and/or speed is randomly sampled (Section 5.4.2), or the trajectory eventually leaves the current sea lane.

The overall performance of the detectors is not very impressive. Given the rather constrained behaviour of the normal trajectories (see, e.g., Figure 5.8), it seems reasonable to expect that an effective anomaly detector should be able to detect most of the random and sweeping anomalous trajectories at an early stage, e.g., after a couple or three data points. Yet, the best results achieved are 7.5 and 7 data points (median) for GMM and KDE, respectively, which corresponds to approximately 1.5 km. This distance can be compared to the size of the cells, which is approximately 2 km. We argue that it is the very low likelihood threshold that is the key to explaining the suboptimal results. Consider Figure 5.13 which illustrates the data point whose position likelihood value determines the threshold for the KDE position PDF. This data point corresponds to the minimum point likelihood of the normal test segment that has the largest minimum point likelihood among the 1% normal test segments classified as anomalous. Now, looking at the position PDF, it is clear that the data point is assigned a very low likelihood, even though it belongs to a normal segment and is located close to another trajectory in the training set. The reason for this is that the *relative* probability density is very low; a large majority of the trajectories in the training set for this cell follow two other lanes which are clearly indicated by the high density areas of the position PDF. One percentage of the normal trajectory segments have one or more data points that are at least as extreme as this. In order to suppress the rate of false alarms, we are there-

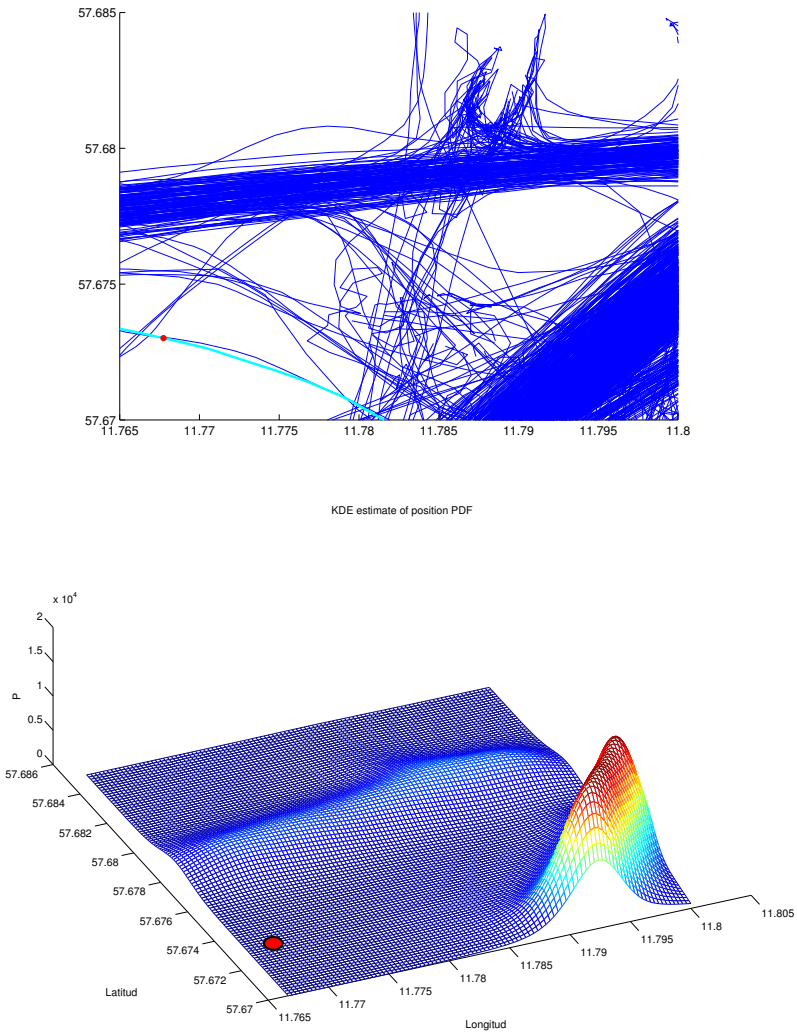

Figure 5.13: Illustration of a position anomaly detected in a normal segment by cell-based KDE, which is tuned to detect position anomalies in 1% of all normal segments. The detected point corresponds to the least anomalous among all data points detected as anomalous in the normal segments. Above: Plot of trajectory segments from local training set (blue) in current cell and anomalous position (red dot) detected in a normal trajectory segment (cyan) from the first test set $\mathbf{T}_{Test1}$. Below: Plot of the position PDF for the cell, estimated using KDE. Note the red point indicating the location and density for the detected position anomaly.

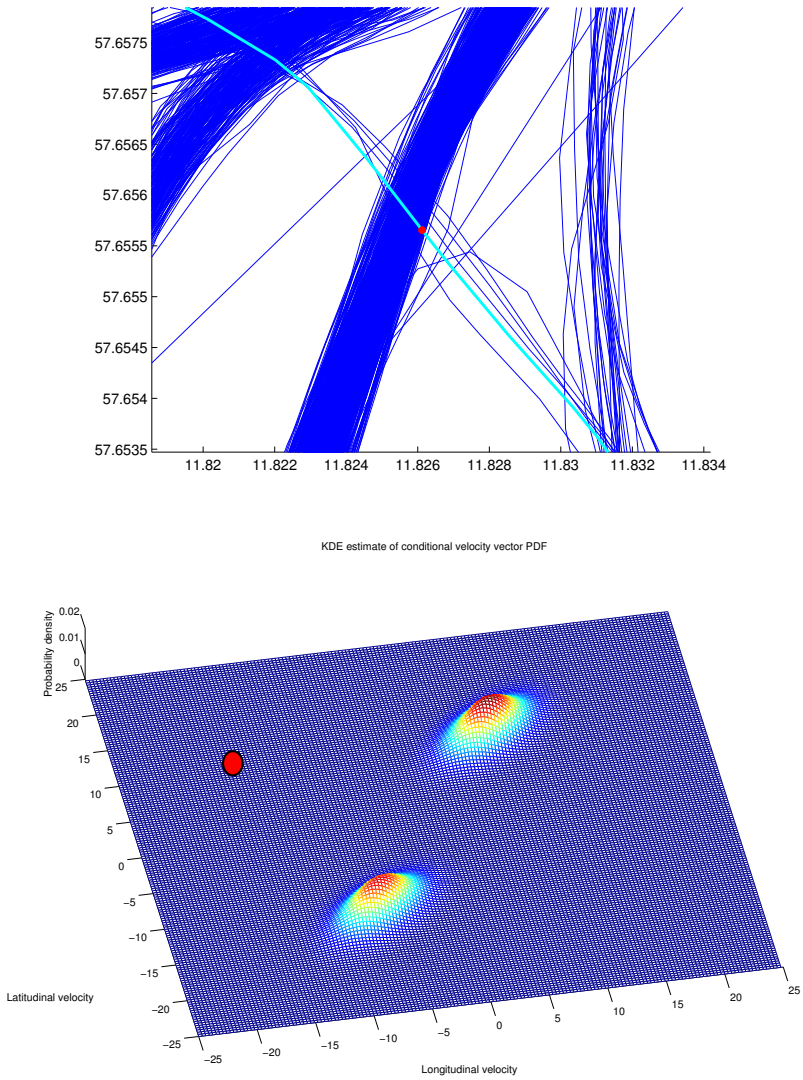

Figure 5.14: Illustration of a position-conditional velocity anomaly detected in a normal segment by cell-based KDE, which is tuned to detect velocity anomalies in 1% of all normal segments. The detected point corresponds to the least anomalous among all data points detected as anomalous in the normal segments. Above: Plot of trajectory segments from local training set (blue) in current cell and location of the velocity anomaly (red dot) detected in a normal trajectory segment (cyan) from the first test set $\mathbf{T}_{Test1}$. Below: Plot of the position-conditional velocity PDF for the position of the velocity anomaly, where the PDF is estimated using KDE. Note the red point indicating the anomalous velocity vector and its density.

fore forced to push the likelihood threshold down to this (very) low level. This means that the true anomalies have to be even more extreme, i.e., they have to have even lower likelihood value, in order to be detected. The situation is similar when considering the data point, illustrated in Figure 5.14, whose velocity vector likelihood determines the threshold for the KDE velocity PDF. From Figure 5.14, it seems that the course has relatively low likelihood, even though there are other trajectories nearby in the training data that have similar course. Yet, since almost all trajectories that are nearby follow another two-directional sea lane, the illustrated data point is assigned a very low likelihood.

Assuming that the segments labelled normal in the test set would indeed be considered normal in a real application, the cell-based approach is clearly suboptimal since it fails to capture relatively infrequent, yet normal patterns, in data. Part of the problem may perhaps be explained by the importance of the cell division and cell size. For example, the data point illustrated in Figure 5.13 would probably not be assigned such as low likelihood if it belonged to the smaller cell defined by the lower left longitude-latitude coordinates $(11.765, 57.670)$ and upper right coordinates $(11.780, 57.676)$. As another example, consider the cell located in the second column and second row, starting from the top, of the grid illustrated in Figure 5.6. In this cell, there is a minor sea lane that extends in north-westwards direction from the major sea lane. Since traffic density of the minor sea lane is relatively low compared to the major sea lane, vessels travelling close to the minor sea lane will be considered rather unlikely, perhaps even anomalous, by cell-based GMM/KDE, even though they follow a sea lane that seems to be part of normalcy. This would not be the case if the minor sea lane was located in a (smaller) separate cell, since its local density would then be larger.

It is quite possible that better performance could be achieved with another cell division where, e.g., cell size is smaller. However, determining the optimal cell division does not seem to be trivial and it may be argued that cell-division is not a natural way to structure and partition vessel trajectory data. An alternative, or complement, to the cell-based modelling is to cluster trajectories following similar routes, with the aim of finding and discriminating the frequent trajectory patterns from the less frequent trajectory patterns (cf. clustering based trajectory anomaly detection methods in Section 2.2.2). For each cluster, the features of trajectories belonging to that cluster may be statistically modelled using GMM or KDE. When classifying a new data point, the most probable cluster would first be determined. Next, its likelihood would be estimated based on the corresponding cluster model. This means that a trajectory following other previous trajectories in a low traffic density area would typically be classified as normal, assuming that a corresponding cluster has been identified for the previous trajectories.

5.4.6 Sequential Anomaly Detection Delay – Second Experiment

In this section, we investigate detection delay on the anomalous trajectory segments of the second test set, $\mathbf{T}_{Test2}$, for the following three detectors: cell-based GMM, cell-based KDE and SPT-CAD. All three detectors are based on the joint position-velocity vector feature model and trained using the second training set $\mathbf{T}_{Train2}$. The objective of this experiment is to investigate:

- Whether SPT-CAD is more sensitive to the anomalous segments than cell-based GMM/KDE.
- Whether the false alarm rate of SPT-CAD is well-calibrated, i.e., if the empirical false alarm rate is close to the specified anomaly threshold ϵ.
- Whether detection delay for cell-based GMM/KDE is affected when incorporating vessel class information.

The setup and design of this experiment is similar to the previous experiment described in Section 5.4.5, but with the following modifications to account for the additional vessel class information of each trajectory: For cell-based GMM and cell-based KDE, a separate model was estimated for each of the three vessel classes. When classifying data points from a test trajectory $T \in \mathbf{T}_{Test2}$, the model corresponding to the class of T was used. In case of SPT-CAD, the subset of trajectories from $\mathbf{T}_{Train2}$ having the same class label as T were provided as training data. For each detector, a single anomaly threshold was used; in case of GMM/KDE, the same likelihood threshold was used for each class specific model. Similar to the first detection delay experiment, the threshold for each detector was tuned to detect one or more anomalous data points in approximately 1.1% or, more exactly, 5 out of 440 of the normal test segments[2]. For SPT-CAD, this rate was achieved by setting ϵ to 0.3%, which resulted in 0.2% of all data points from the normal test segments classified as anomalous.

Results for detection delay are presented in Table 5.4.

Analysis

We observe in Table 5.4 that detection delay for SPT-CAD is less than the detection delay for cell-based GMM/KDE. Hence, it seems that SPT-NCM based on the nearest neighbour distance (Section 4.3.2) is a rather good outlier measure compared to cell-based GMM/KDE. This is perhaps explained by the fact that SPT-NCM is not density-based and therefore, in contrast to GMM and KDE, does not penalise the more infrequent, yet normal, patterns in data (see analysis of results from the previous detection delay experiment in Section 5.4.5). Yet,

[2]It was, for numerical reasons, not possible to achieve exactly 1% anomalous test segments.

Table 5.4: Sequential anomaly detection delay on the anomalous segments of the second test set of vessel trajectories $\mathbf{T}_{Test2}$ (Section 5.4.6). The anomaly threshold of each detector has been tuned so that 1.1% of the normal trajectory segments in $\mathbf{T}_{Test2}$ have one or more data points classified as anomalous.

Detector	Mean	Median
SPT-CAD	3.99	1
Cell-based GMM	6.75	3
Cell-based KDE	4.69	2

results for SPT-CAD and cell-based KDE are quite close and the median detection delay for both methods is very low (1 and 2, respectively), indicating that most anomalous trajectories are very easy to detect. Hence, it is not obvious what conclusions can be drawn regarding relative sensitivity of the detectors.

It is perhaps of more interest to compare the results in Table 5.4 with the results from the first detection delay experiment in Table 5.3. From this comparison, it is clear that performance of cell-based GMM/KDE has improved significantly in the second experiment. This performance improvement is probably related to the additional class information during learning and classification, and the fact that only trajectories of vessel class type cargo ship, tanker and passenger ship are considered. The behaviour of these vessels appears to be more predictable; they follow sea lanes more strictly compared to other vessels, such as tugs and pilots, which move around more freely in the port area. This, we believe, makes the anomalous trajectories appear more anomalous and, hence, performance of cell-based GMM/KDE is improved.

Finally, we observe that the rate of normal data points classified as anomalous (0.2%) by SPT-CAD is just below the anomaly threshold (0.3%). Thus, the false alarm rate, in terms of normal data points classified as anomalous, is well-calibrated.

5.4.7 Anomaly Detection – Precision and Recall

This experiment was first published by Brax et al. (2010), who evaluated precise and imprecise State-based Anomaly Detection (SBAD) using the first training set $\mathbf{T}_{Train1}$ and the third test set $\mathbf{T}_{Test3}$. We repeat this experiment using SNN-CAD with the objective of evaluating its relative classification performance. Given the training set $\mathbf{T}_{Train1}$, the anomaly detector simply has to classify each complete trajectory $T \in \mathbf{T}_{Test3}$ as either normal or anomalous. Results for SNN-CAD are compared to the best results reported by Brax et al. (2010) in Table 5.5.

Table 5.5: Precision and recall (Section 5.1) for different anomaly detectors on the third test set of vessel trajectories $\mathbf{T}_{Test3}$. Note that results for SBAD were previously published by Brax et al. (2010).

Detector	Precision	Recall (sensitivity)
SNN-CAD with $S = \overrightarrow{\delta_H}$	0.97	0.98
Precise SBAD with parameter setting B (Brax et al., 2010)	0.88	0.98
Imprecise SBAD with parameter setting B (Brax et al., 2010)	0.92	0.97

Analysis

In Table 5.5, we see that while SNN-CAD and precise/imprecise SBAD all attain a high level of recall (0.98, 0.98 and 0.97, respectively), SNN-CAD has a higher level of precision (0.97 compared to 0.88 and 0.92).

The normalcy model in SBAD is based on the frequency of different discrete kinematic states and their transitions among data points from normal trajectories. A possible explanation for the superior performance of SNN-CAD is that it models the relationship between all (observed) data points from the corresponding trajectory. In contrast, SBAD only considers the relationship between successive data points, i.e., the relative frequency of one-step state transitions. Moreover, SNN-CAD does not require that features are discretisized, which, in case of SBAD, may result in some loss of sensitivity.

5.4.8 Anomaly Detection – False Alarm Rate

In this section, we investigate the false alarm rate of SNN-CAD during online learning and anomaly detection in the first set of normal vessel trajectories $\mathbf{T}_{Train1}$. The objective of this experiment is to investigate whether the false alarm rate is well-calibrated, i.e., if the empirical false alarm rate is close to the specified anomaly threshold ϵ: According to Theorem 3.1, the expected false alarm rate for SNN-CAD is equal to ϵ, if the training set and new example are IID.

Starting with an initial training set $\mathbf{T}_{Init} = \{T_1, \ldots, T_{100}\} \subseteq \mathbf{T}_{Train}$ of 100 randomly sampled normal trajectories, we let three parallel detectors, as proposed in Section 4.4.3, classify the sequence of remaining trajectories $(T_{101}, \ldots, T_{2310})$ from $\mathbf{T}_{Train}$. After each classification, the corresponding trajectory is added to the accumulated training set, i.e., online learning, regardless of whether it was classified as normal or anomalous. In theory, we may start with an empty training set. However, in a real applications, there will typically be a set of normal trajectories available prior to the deployment of the anomaly

Table 5.6: Empirical false alarm rates, for multiple detectors based on SNN-CAD, during online learning and anomaly detection in a sequence of normal vessel trajectories from $\mathbf{T}_{Train1}$ (Section 5.4.8). The combined detector corresponds to a logical OR combination of the three detectors.

	Empirical false alarm rate
Location detector with $\epsilon' = 0.01$	1.04%
Speed-location detector with $\epsilon'' = 0.01$	0.85%
Course-location detector with $\epsilon''' = 0.01$	0.81%
Logical OR combination of all detectors	1.99%

detector. Hence, we have set the size of the initial training set to 100, which is relatively small compared to the size of total set (2310). Analogously to the previous experiments, the anomaly threshold for each detector was set to 1%, i.e., $\epsilon' = \epsilon'' = \epsilon''' = 0.01$.

Results for the false alarm rates are shown in Table 5.6.

Analysis

In Table 5.6, we see that the empirical false alarm rate for each of the three detectors is close to the expected false alarm rate 1%. It also interesting to note the false alarm rate of the combined detector (1.99%). Clearly, some of the trajectories are classified as anomalous by more than one of the detectors. Moreover, the overall false alarm rate indicates that anomalies detected by different detectors are not independent. Because, if they were indeed independent, we would expect an overall false alarm rate close to $1 - 0.99^3 = 2.97\%$, where 0.99^3 is the probability that a random normal sample will not be classified as anomalous by any of the three detectors. Yet, the overall false alarm rate can still be controlled in practise, since it is bounded by 1 minus the product of the individual anomaly thresholds. In a real application, the anomaly thresholds ϵ', ϵ'' and ϵ''' may be set to different levels, if there is a need for suppressing or favouring particular types of anomalies.

5.4.9 Summary

A number of experiments have been carried out where cell-based GMM and KDE, SPT-CAD based on Euclidean nearest neighbour distance and SNN-CAD based on directed HD have been evaluated on different data sets of real and simulated vessel trajectories.

To start with, qualitative and quantitative results indicate that KDE is a more accurate model than GMM for the position and velocity vector features of normal vessel trajectories. However, no significant difference between GMM and KDE was observed for detection delay in the anomalous trajectories. Moreover, results for cell-based GMM/KDE in the first detection delay

experiment were considered suboptimal. After examining more closely some of the false alarms of cell-based KDE, it was concluded that the model fails to account for relatively infrequent, yet normal, patterns in the training data. Despite the fact that these patterns re-appear in the normal test data, they are assigned a very low likelihood since their relative density in the cell is very low. In order to better account for these patterns, trajectory clustering, as an alternative or complement to cell-division, prior to statistical modelling and anomaly detection was discussed.

An interesting observation from the first detection delay experiment was that the best performance for GMM and KDE was obtained when only considering the position features, i.e., without considering velocity vector features. This indicates that the position is a more discriminating feature than the velocity vector for detecting the anomalous trajectory segments. Furthermore, increased performance was achieved by combining the output of two separate detectors for position and velocity vector, respectively, rather than considering a single detector for the joint position-velocity vector.

Results from the second detection delay experiment show that SPT-CAD performs relatively well compared to cell-based GMM/KDE. A possible explanation for these results is that the former algorithm is not density-based and therefore better captures less frequent patterns in normal data. Yet, we observe that performance for cell-based GMM/KDE is significantly better than in the first detection delay experiment. This difference is probably due to the additional class information during learning and anomaly detection, and the fact that normal trajectories belong to a subset of vessel classes whose behaviour seems to be more predictable.

We have also reproduced an experiment previously published by another author, where precision and recall for different anomaly detectors are measured on a labelled set of vessel trajectories. Results show that SNN-CAD based on HD is an accurate anomaly detector compared to previously proposed anomaly detectors.

5.5 Anomaly Detection in Synthetic Trajectory Data

In this section, we further evaluate SNN-CAD and the proposed trajectory dissimilarity measures using a public data set of simulated trajectories. Two experiments are carried out where one is a repetition of an experiment previously published. Similar to the experiments in the previous section, we have not implemented an exact algorithm for calculating HD between two trajectories. Instead, we use the naive algorithm that approximates HD between two trajectories by only considering the finite set of end points of the corresponding line segments.

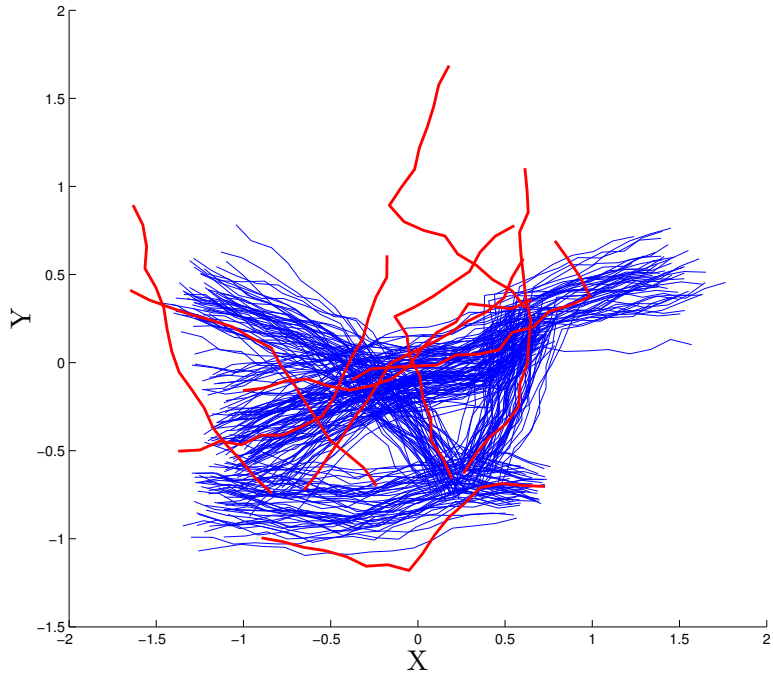

Figure 5.15: Plot of 260 trajectories from one of the 1000 synthetic data sets used in Section 5.5. Blue trajectories are labelled as normal and red trajectories are labelled as anomalous. All anomalous trajectories in this particular data set are detected by SNN-CAD with no false alarms.

5.5.1 Data Description

The public data set[3] of simulated trajectories was previously created by Piciarelli et al. (2008). The data set consists of two main parts; in this paper we use the second part[4], which consists of 1000 randomly generated data sets. Each of these data sets contain 260 2-dimensional trajectories of length 16. Of the 260 trajectories, 250 belong to 5 different clusters and are labelled as normal. The remaining 10 are stray trajectories that do not belong to any cluster; they are labelled as anomalous (see Figure 5.15 for a plot of one of the data sets).

[3] http://avires.dimi.uniud.it/papers/trclust/

[4] The reason we do not use the first part of the data set is that the corresponding training data includes anomalies. While this a very interesting situation, it is considered to be out of the scope of the thesis.

Table 5.7: Average accuracy for different outlier measures on the public set of synthetic trajectories (Section 5.5.2). Note that accuracy results for SVM and discords are 1 minus the corresponding error rate reported by Piciarelli et al. (2008).

	# of most similar neighbours considered				
Outlier measure	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
SNN-NCM with $S = \overrightarrow{\delta_H}$	96.59%	97.12%	97.05%	96.84%	96.72%
SNN-NCM with $S = \delta_H$	97.28%	97.66%	97.63%	97.57%	97.37%
SVM (Piciarelli et al., 2008)	96.30% (Piciarelli et al., 2008)				
Discords (Keogh et al., 2005)	97.04% (Piciarelli et al., 2008)				

5.5.2 Accuracy of Outlier Measure

We reproduce one of the experiments published by Piciarelli et al. (2008) where the authors evaluate two anomaly detectors based on two different *outlier measures*: the first is based on a Support Vector Method (SVM) (Piciarelli et al., 2008) and the second is based on time-series *discords* (Keogh et al., 2005). Given a set of trajectories, the outlier measures produce an outlier score for each trajectory relative the rest. For each of the 1000 synthetic data sets described above, the authors calculated outlier scores for each of the 260 trajectories and checked if the 10 trajectories with highest outlier scores correspond to the 10 trajectories labelled as anomalous. The *error rate*, which is equivalent to 1 minus accuracy, was calculated for each method by averaging over the number of normal trajectories among the 10 trajectories with highest outlier scores. Since the nonconformity scores output by SNN-NCM correspond to the outlier scores output by the other two outlier measures, comparison is straightforward.

Accuracy results for SNN-NCM and the corresponding accuracy results for SVM and discords reported by Piciarelli et al. (2008) are summarised in Table 5.7.

Analysis

It is clear from Table 5.7 that SNN-NCM based on HD is an accurate outlier measure for the simulated trajectories. In particular, the undirected HD outperforms the other two trajectory outlier measures, regardless of the parameter value k. It is not surprising that the undirected HD performs better than the directed HD for complete trajectories, since the former utilises all points from *both* trajectories during comparison (the directed HD only considers distance to a subset of points from the other trajectory). However, the undirected

measure is not appropriate for comparing incomplete trajectories and thus not applicable for sequential anomaly detection in trajectories.

5.5.3 Online Learning and Sequential Anomaly Detection

In this section, we evaluate SNN-CAD with $S = \overrightarrow{\delta_H}$ for online learning and sequential anomaly detection in the synthetic trajectories. The first objective is to demonstrate that anomalies can be detected with high sensitivity and low FAR before the complete trajectory has been observed, i.e., with a detection delay less than 16 data points. The second objective is to show that FAR is well-calibrated and that the sensitivity to true anomalies increases during semi-supervised online learning.

Design

For each of the 1000 sets of 260 synthetic trajectories (Section 5.5.1), we do the following: First we create an initial training set by randomly sampling 100 normal trajectories among the 250 labelled as normal. The remaining 160 trajectories (150 normal and 10 anomalous) are then randomly permuted and sequentially presented to the algorithm. For each trajectory $T_i = \left(z_1^i, \ldots, z_{16}^i\right), i = 1, \ldots, 160$, we let SNN-CAD with $k = 2$ (cf. Table 5.7) and $\epsilon = 0.01$ (similar to previous experiments) sequentially classify each incomplete trajectory $T_i^\star = \left(z_1^i, \ldots, z_m^i\right), m = 1, \ldots, 16$, as normal or anomalous. If T_i is labelled anomalous and successfully detected as anomalous, it is simply discarded. In all other cases, the complete trajectory $T_i = \left(z_1^i, \ldots, z_{16}^i\right)$ is added to the training set before classifying the next trajectory, T_{i+1}, regardless of whether T_i is actually labelled as normal or anomalous. Thus, the algorithm operates in an online semi-supervised learning mode, where the true label for new examples are only given for those detected as anomalies; this corresponds to a setting where, e.g., a human is alerted of trajectories detected as anomalous and either confirms or rejects each alarm.

Results

Results for the 1000 trajectory sets are as follows: average sensitivity and FAR are 98% and 0.96%, respectively. The median detection delay is 6 out of 16 data points. Moreover, the number of false negatives based on the size of the accumulating training set is illustrated Figure 5.16.

Analysis

Results show that SNN-CAD often detects the labelled anomalies before half the trajectory has been observed. This is done at a high level of sensitivity (98%) and a low level of FAR (0.96%), which is well-calibrated, i.e., close to the

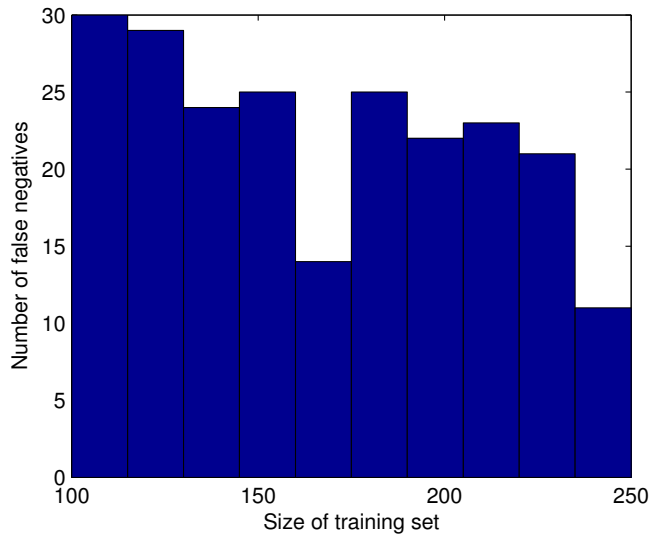

Figure 5.16: Histogram showing frequency of false negatives, i.e., missed anomalies, depending on the size of the accumulated training set for SNN-CAD during online learning and sequential anomaly detection. Results are based on 1000 experiments on 1000 different data sets of simulated trajectories (Section 5.5.3), i.e., one experiment per data set.

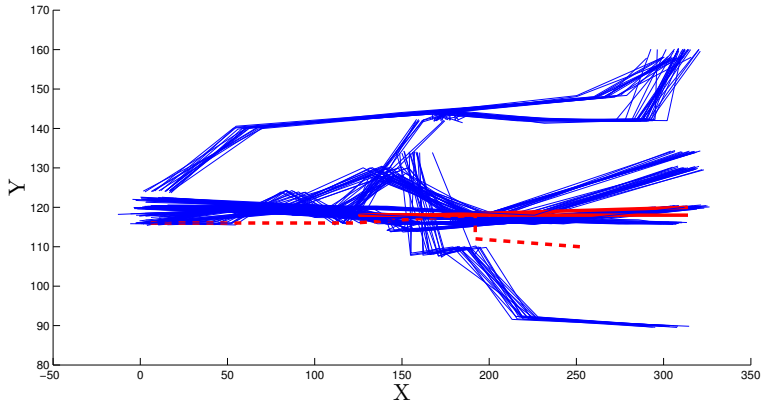

Figure 5.17: Plot of the 239 video surveillance trajectories (Pokrajac et al., 2007). Blue corresponds to normal trajectories. Two anomalous trajectories are indicated by the solid red and dashed red trajectories. Red solid corresponds to a person walking left and then back right and dashed red corresponds to person walking very slowly (Pokrajac et al., 2007). Both anomalies are detected by SNN-CAD with perfect accuracy, i.e., no false alarms (Section 5.6).

specified anomaly threshold $\epsilon = 0.01$. Moreover, Figure 5.16 indicates that sensitivity increases as more training data is accumulated, even though 2% of the labelled anomalies are on average erroneously classified as normal and added to the training set. These outliers in the training set does not seem to have any impact on FAR.

5.6 Anomaly Detection in Real Video Trajectory Data

In this section, we reproduce an experiment on a public data set of 2D video trajectories[5]. The data set consists of 239 labelled video motion trajectories where only two trajectories were visually identified as unusual behaviour (Pokrajac et al., 2007). The trajectories were extracted from IR surveillance videos using a motion detection and tracking algorithm (Pokrajac et al., 2007). Each trajectory is represented by five points in $(x, y, time)$ space. A plot of all 239 trajectories is shown in Figure 5.17.

Analogously to Pokrajac et al. (2007), we calculate the outlier score for each of the 239 trajectories relative the rest using SNN-NCM with $S = \overrightarrow{\delta_H}$ and $k = 1$ (which is a standard value). Sorting the resulting nonconformity scores, we observe that the two trajectories labelled as anomalous have the top-two largest outlier scores. Hence, similar to Pokrajac et al. (2007) and others

[5] www.cs.umn.edu/~aleks/inclof

(Isaksson and Dunham, 2009; Yankov et al., 2008), we achieve perfect accuracy on this data set.

5.7 Discussion

Intuitively, the approximation error of the implemented algorithm for calculating HD between two polygonal curves is bounded by the sampling distance of the points; the more sample points, the more accurate the approximation. It is possible that even better classification accuracy can be achieved by either considering a smaller sampling distance, or by calculating the exact HD using, e.g., the algorithm proposed by Alt et al. (1995). An advantage of the latter algorithm is that computational complexity can be reduced by compressing trajectories using, e.g., the Douglas-Peucker algorithm for line simplification (Douglas and Peucker, 1973).

There exists a lot of published work on algorithms for anomaly detection in trajectory data. But as far as we know, there are no experimental results published on a *public* data set of *real* trajectories with a *fairly large* amount of labelled anomalies. Indeed, acquiring labelled anomalies is usually difficult (Section 2.1.1). Many authors publish good or excellent results on simulated labelled data (e.g., Urban et al. (2010)) or limited amounts of real data with relatively few, if any, labelled anomalies (e.g., Pokrajac et al. (2007); Naftel and Khalid (2006)). Results are often of qualitative nature where authors show a few detected anomalies and argue that these can be considered true positives (e.g., Hu et al. (2006); Pokrajac et al. (2007); Ristic et al. (2008); Johansson and Falkman (2007)). With a few exceptions (e.g., Pokrajac et al. (2007); Piciarelli et al. (2008)), data sets are typically not publicly available. Moreover, it is difficult to asses relative performance of different anomaly detectors when the amount of labelled anomalies is small, since the statistical significance of the results may be questioned.

5.7.1 Limitations

An important issue that has not been addressed in the empirical investigations is computational complexity. In particular, computational complexity during sequential anomaly detection is critical in a real world surveillance application. Another important issue that has not been addressed is how different amounts of anomalies in training data affect learning and anomaly detection performance of the proposed algorithms. For example, subtle anomalies may incorrectly be classified as normal and added to the training set during semi-supervised learning (see Section 5.5.3). Normal trajectories, extracted from, e.g., tracking systems, may include noise due to incorrect measurements etc. Hence, further investigations regarding sensitivity to noise and anomalies, and methods for handling these, should be pursued in future work.

As can be seen from the experimental overview in Table 5.1, only a subset of all the algorithms proposed in this thesis have been evaluated in each of the experiments. Ideally, all algorithms would have been evaluated in each of the experiments. However, this has not been done due to time limitations.

5.8 Summary

In this chapter, we first introduced and discussed a number of performance measures appropriate for evaluating algorithms for sequential anomaly detection in trajectory data. We then carried out a number of experiments where the algorithms proposed in this thesis where evaluated on a number of different data sets. In the first experiment, we demonstrated that the combination of cell-based GMM and the position-velocity feature model is a feasible approach to sequential anomaly detection in vessel traffic; it was shown that vessels crossing sea lanes, violating traffic direction and travelling at relatively high speeds can be detected at a low alarm rate (0.1%). This was followed by a number of experiments where cell-based GMM/KDE, SPT-CAD based on Euclidean nearest neighbour distance and SNN-CAD based on directed HD were evaluated on different data sets of real and simulated vessel trajectories. The main results and conclusions from these experiments are as follows:

- KDE is a more accurate model than GMM for the position and velocity vector features of normal vessel trajectory points.
- Cell-based GMM/KDE are suboptimal for anomaly detection in vessel traffic, since they fail to account for relatively infrequent, yet normal, patterns in the training data.
- SPT-CAD performs relatively good compared to cell-based GMM/KDE, probably due to the fact that the former is not density-based and therefore better captures the less frequent patterns in normal data.
- Position is a more accurate feature than the velocity vector for detecting the anomalous trajectory segments.
- Increased sensitivity to anomalies can be achieved by combining the output of two separate detectors for position and velocity vector, respectively, rather than considering a single detector for the joint position-velocity vector.
- SNN-CAD based on HD is an accurate anomaly detector compared to previously proposed anomaly detectors.

In the last experiments of this chapter, learning and anomaly detection performance of SNN-CAD was further investigated on two public and labelled data sets of synthetic and real video trajectories, respectively. The results from

these experiments showed that SNN-NCM based on HD is an accurate outlier measure for trajectories. It was also demonstrated that the proposed algorithm is capable of detecting labelled anomalies in incomplete trajectories with high sensitivity (98%) and low FAR (0.96%), which is well-calibrated. Moreover, results indicated that sensitivity to labelled anomalies increases as more training data is accumulated during semi-supervised online learning.

Chapter 6
Conclusions

The aim of this thesis was to *investigate properties and performance of algorithms for anomaly detection in trajectory data for surveillance applications, and propose new or updated algorithms that are better suited for this task*. (Section 1.1). In this final chapter, we describe to what degree this aim has been fulfilled. The chapter presents the main conclusions and contributions of the thesis (Section 6.1), an outline of future work (Section 6.2) and how the results from this work can be generalised to other research and application domains (Section 6.3). Final remarks are given in Section 6.4.

6.1 Contributions

We centre the discussion of the contributions around Objective 1–6, presented in the introduction of the thesis (Section 1.1).

Objective 1: Identify important and desirable theoretical properties of algorithms for anomaly detection in surveillance applications

The underlying assumption and starting-point of the thesis is that there is a need for automated anomaly detection in surveillance applications, which motivates the main research aim. However, from the outset, it was not obvious what are the critical and desirable requirements and properties of such algorithms. Hence, the first step towards the research aim of the thesis was to investigate key properties of algorithms for anomaly detection in surveillance applications.

Most of the key properties identified in this thesis were already presented in the introduction and can be summarised as follows:

- Support for online learning, requiring no or limited human feedback.
- Sequential anomaly detection in incomplete trajectories.
- A minimum of parameters.

- Anomaly thresholds that are intuitive for intended users.
- Well-calibrated false alarm rate.

These properties were further discussed in Section 3.1 and 4.1. It should be noted that most of the properties have separately been identified and discussed by previous authors in related research fields. Yet, we argue that the compilation and overall discussion of all these properties is a contribution by itself in the field of anomaly detection in surveillance applications.

Objective 2: Review and analyse previously proposed algorithms for anomaly detection in trajectory data

In this thesis, we have undertaken a literature study in which previously proposed models and algorithms for anomaly detection in general and trajectory anomaly detection in particular have been reviewed. The result of this study was presented in Section 2.1 and 2.2, i.e., the background of the thesis. Moreover, some of the key observations from the analysis of anomaly detection algorithms are presented and further discussed in Section 3.1 and 4.1.

A substantial portion of the review is based on two recently published surveys by Chandola et al. (2009), regarding anomaly detection in general, and Morris and Trivedi (2008a), regarding anomaly detection in video trajectory data. This work has been complemented by studying other literature related to anomaly detection in general and anomaly detection in trajectory data. Apart from constituting a complemented overview of the research area, the literature study has served as a basis for the analysis of previous work and identification of the key properties in Objective 1.

Objective 3: Propose algorithms that are well-suited for anomaly detection in trajectory data

In Chapter 4, a number of models and algorithms for sequential anomaly detection in trajectory data have been proposed: *cell-based GMM*, *cell-based KDE*, *SPT-CAD* and *SNN-CAD based on directed HD*. Each of the algorithms addresses some or all of the key properties identified in Objective 1 and can be categorised according to the underlying learning and anomaly detection algorithm adopted, i.e., GMM, KDE or CAD, and the type of feature model adopted, i.e., point-based or trajectory-based.

Starting with the cell-based GMM and KDE algorithms, they are point-based anomaly detectors that are relatively easy to implement, require minimal preprocessing of trajectory data and do not have many parameters. The fundamental learning and anomaly detection algorithm for cell-based KDE is similar to other previously proposed algorithms (Ristic et al., 2008). For learning in cell-based GMM, we proposed a simple extension of the standard version

of EM for determining an appropriate number of mixture components (Section 4.3.1). As discussed in Section 4.5, more efficient and sophisticated algorithms for determining the optimal number of components have previously been proposed (Verbeek et al., 2003); however, our algorithm has the advantage that it is relatively simple to implement. The main novelties of the cell-based GMM and KDE algorithms in the context of trajectory anomaly detection are two-fold: Firstly, a grid-based approach to suppress model complexity has been introduced, where a separate model is estimated for each cell based on the local training data (Section 4.3.1). This approach is appropriate for, e.g., wide-area surveillance applications. Secondly, a novel approach to point-based statistical anomaly detection has been proposed that involves the combination of the output from two separate detectors based on the unconditional position PDF and position-conditional velocity vector PDF, respectively (Section 4.3.1). An advantage of this approach compared to previously proposed detectors based on the joint PDF (e.g., Ristic et al., 2008) is that more information is provided regarding which of the features contributed to an anomaly. Moreover, the risk that subtle anomalies in lower dimensional feature spaces are marginalised due to the curse of dimensionality is reduced.

Considering SPT-CAD and SNN-CAD, these are both based on CAD, which is a novel anomaly detection algorithm proposed in Chapter 3. CAD is based on the theory of Conformal prediction (Vovk et al., 2005) and a central property that follows from this is that the false alarm rate is well-calibrated, i.e., the expected rate of false alarms equals the specified anomaly threshold $\epsilon \in (0, 1)$, under the assumption that the training data and new normal data are IID. Thus, no application specific anomaly threshold is required. The main design parameter in CAD is the nonconformity measure (NCM). We have proposed SNN-NCM, which is a new NCM appropriate for applications where data is represented as sets or sequences of different size or length, such as trajectories (Section 3.3.5). Apart from the number of most similar neighbours k, the only design parameter in SNN-NCM is the dissimilarity measure S, which can be chosen freely. For trajectory-based anomaly detection, we proposed two parameter-free dissimilarity measures based on HD for comparing multi-dimensional trajectories of arbitrary length (Section 4.4). One of these measures is appropriate for sequential anomaly detection in incomplete trajectories. We have also proposed SPT-NCM, which is another new NCM specifically designed for point-based anomaly detection in SPT-CAD (Section 4.3.2).

The CAD-based algorithms proposed in this thesis are unique in the sense that they address all of the key properties identified in Objective 1, i.e., support for online learning, sequential anomaly detection in incomplete trajectories, very few parameters, application independent anomaly threshold and well-calibrated false alarm rate. The main difference between SPT-CAD and SNN-CAD based on HD is that the latter is trajectory-based and, hence, is more sensitive to anomalous behaviour that develop over time.

Objective 4: Demonstrate feasibility and validity of proposed algorithms on real world surveillance data sets

All the models and algorithms proposed in this thesis have been implemented and demonstrated on real world data sets in Chapter 5. In Section 5.3, the cell-based GMM based on the position-velocity feature model was demonstrated using a relatively large set of unlabelled data extracted from a vessel track database. Results showed that vessels crossing sea lanes, violating traffic direction and travelling at relatively high speeds can be detected at a low overall alarm rate (0.1% of all data points). Moreover, results indicate that the algorithm is sensitive to anomalous combinations of the feature values, e.g., vessels crossing sea lanes at novel locations. The point-based feature model has theoretical limitations regarding the type of anomalous behaviour that can be detected. For example, it is insensitive to anomalous routes since it does not capture behaviour over time. However, it is still capable of detecting a subset of anomalous behaviour, as described above. Moreover, it is relatively simple to implement and requires relatively little preprocessing of data, as discussed in Objective 3 above. Hence, we conclude that the proposed algorithm is a feasible, yet limited, approach to sequential anomaly detection in vessel trajectory data.

The cell-based GMM and KDE, SPT-CAD and SNN-CAD based on HD were all implemented and evaluated using data extracted from an AIS database of recorded sea traffic (Section 5.4). The algorithms were trained using randomly sampled trajectories from the AIS data, assumed to be normal. Test data consisted of other trajectories extracted from the AIS database, also assumed to be normal, mixed with simulated trajectories assumed to be anomalous. Results showed that the anomalous trajectories, generated according to a random walk function, can be distinguished from normal vessel trajectories at a low false alarm rate (1%). Moreover, SNN-CAD based on HD was demonstrated on a relatively small set of real video trajectories (Section 5.6). These results showed that the proposed algorithm can achieve perfect accuracy on a labelled data set, i.e., detection of all labelled anomalies without any false alarms.

Objective 5: Identify suitable performance measures for evaluating algorithms for anomaly detection in trajectory data

Obviously, a central performance measure of any anomaly detector is its ability to accurately distinguish between normal and anomalous examples. In this thesis, we have considered *accuracy*, *precision, recall* and *false alarm rate* (FAR), which are standard classification performance measures in pattern recognition (Fawcett, 2006). In particular we have argued, supported by previous work (Axelsson, 2000; Riveiro, 2011), that FAR is a critical performance measure in anomaly detection applications and that it should be kept at a low or very low level.

In case of sequential anomaly detection, we are also interested in minimising the time, i.e., the number of data points, required for accurately classifying incomplete trajectories. Detection delay, which is a well-known performance measure in the domain of change-detection (Ho and Wechsler, 2010), has therefore been proposed as a complement to the traditional classification performance measures when evaluating sequential anomaly detectors (Section 5.1). As far as we know, detection delay has never before been used as a performance measure in the domain of trajectory anomaly detection.

Evaluating the performance measures discussed above requires a test set of trajectories labelled normal and anomalous. Typically, there are large amounts of historical data available that more or less reflects normalcy. Yet, acquiring a representative set of labelled anomalies is problematic, since anomalies typically occur (very) rarely and may appear very different from each other. Good accuracy on a few prototypical anomalies is no guarantee that the detector will successfully detect future anomalies. In order to obtain statistically significant results, a sufficiently large set of trajectories labelled normal and anomalous is needed. One approach to circumvent this problem is to simulate anomalous trajectories. In this thesis, we have therefore proposed a method for simulating anomalous vessel trajectories based on a random walk (Section 5.4.2).

It may be argued that obtaining an accurate normalcy model is a prerequisite for good classification accuracy of any anomaly detector. Hence, to complement the performance measures discussed above, we have introduced a performance measure known as *normalcy modelling performance* (Section 5.4.4). This measure aims to quantify the relative accuracy of different methods for estimating an *unknown* PDF for normal data. In contrast to classification accuracy, evaluating normalcy modelling performance only requires data labelled as normal.

Objective 6: Evaluate proposed algorithms according to identified performance measures

Based on the performance measures from Objective 5 above, we have evaluated the proposed algorithms using a number of relatively large data sets. In Section 5.4.4, we compared normalcy modelling performance for cell-based GMM and KDE using a large data set of vessel traffic. Results from this experiment confirmed our hypothesis that KDE is a more accurate model than GMM for the position and velocity vector features of vessel trajectory points. In Section 5.4.5 and 5.4.6, we evaluated detection delay for all the point-based anomaly detectors on simulated anomalous vessel trajectories, using real vessel trajectories assumed to be normal as training data. Results from these experiments showed that cell-based GMM and KDE are suboptimal for anomaly detection in vessel traffic, since they fail to account for relatively infrequent, yet normal, patterns in the training data. SPT-CAD performed good, compared to cell-based GMM and KDE, which may be explained by the fact that the former

is not density-based and therefore better captures the less frequent patterns in normal data. Moreover, for cell-based GMM and KDE, it was discovered that the position is a more discriminating feature than the velocity vector for detecting the anomalous trajectories. Results also indicated that a combined detector, based on two separate low-dimensional detectors for the unconditional position PDF and the position-conditional velocity PDF, respectively, outperforms a single detector based on the joint position-velocity vector PDF.

Previous results from evaluations of algorithms for anomaly detection in trajectory data are rather limited (Section 5.7). Generally, there are no results on standardised public data sets of labelled trajectories, since few such data sets seem to exist. Hence, it is difficult to discuss the empirical results in this thesis in relation to previous empirical results. Yet, in the last experiments of Chapter 5, the learning and anomaly detection performance of SNN-CAD was further investigated on two public and labelled data sets of synthetic and real video trajectories, respectively. Results from these experiments showed that SNN-NCM based on HD is an accurate outlier measure for trajectories, compared to previously published algorithms (Section 5.5.2 and 5.6). Thus, good classification results have been achieved by SNN-CAD on public data sets without any parameter tuning. It was also demonstrated that SNN-CAD based on directed HD is capable of detecting labelled anomalies in incomplete trajectories with high sensitivity (98%) and low FAR (0.96%), which is well-calibrated. Moreover, results indicated that sensitivity to labelled anomalies increases during semi-supervised online learning, i.e., classification performance increases as more training data is accumulated.

6.1.1 Summary of Contributions

In this thesis, we have investigated algorithms appropriate for sequential anomaly detection in trajectory data for surveillance applications. We have identified and discussed some key theoretical properties of such algorithms, based on a literature study, which has been carried out in this thesis. The key properties include: sequential anomaly detection in incomplete trajectories, online learning based on new data requiring no or limited human feedback, a minimum amount of parameters and a well-calibrated false alarm rate. A number of algorithms founded on statistical and nearest neighbour methods have been proposed for sequential anomaly detection in trajectory data. Two of these algorithms, SPT-CAD and SNN-CAD, are unique in the sense that they address all of the key properties. They are both based on CAD, which is a novel algorithm for anomaly detection proposed in this thesis. CAD is founded on the theory of CP and a key property that follows from this is that the false alarm rate is well-calibrated. The main difference between SPT-CAD and SNN-CAD is that the latter is trajectory-based and, hence, more sensitive to anomalous behaviour that develops over time. The only design parameter in SNN-CAD is the dissimilarity measure; we have proposed the use of directed and undir-

ected HD, which are both parameter-free dissimilarity measures, for anomaly detection in incomplete and complete trajectories, respectively.

The proposed algorithms have been evaluated on real world data sets, including vessel traffic data, which have been complemented with simulated anomalous data. A number of relevant performance measures have been identified and discussed; two of these are novel in the context of trajectory anomaly detection. The experiments have demonstrated the type of anomalous behaviour that can be detected at a low overall alarm rate. Quantitative results for learning and classification performance of the algorithms have been compared. These results indicate that the statistical methods fail to account for relatively infrequent, yet normal, patterns in some data sets. Moreover, results from reproduced experiments on public data sets indicate that the classification performance of SNN-CAD is relatively good compared to previously published algorithms. Hence, it is concluded that SNN-CAD based on HD is a promising algorithm for anomaly detection in trajectory data.

6.2 Future work

In this section, we list a number of directions for future work based on the results and conclusions of this thesis:

- Investigations of sensitivity to noise and anomalies in training data.

  An important issue that has not been addressed in this thesis is how different amounts of anomalies in training data affect learning and anomaly detection performance of the proposed algorithms. For example, subtle anomalies may incorrectly be classified as normal and added to the training set during online semi-supervised learning (see Section 5.5.3). Normal trajectories, extracted from, e.g., tracking systems, may include noise due to incorrect measurements etc. Hence further investigations regarding sensitivity to noise and anomalies, and methods for handling these, should be pursued in future work.

- Investigations of long-term online learning of CAD, addressing the issues of *concept drift* (Hand, 2006) and *population drift* (Hand, 2006) and complexity issues related to size of training set.

  In this thesis, we have essentially assumed that new data not classified as anomalous by a human is simply added to the training set during online learning. This means that the size of training set monotonically increases as new data is observed. Yet, it has been argued that "it is impractical to store and use all the historical data for training, since it would require infinite storage and running time" (Masud et al., 2009). Moreover, "there may be concept-drift in the data, meaning, the underlying concept of the data may change over time" (Masud et al., 2009). The issue of concept-drift is closely related to that of population drift, which means

that the underlying distribution, from which new data is sampled, has changed (Hand, 2006). Trajectory data in the intelligence and surveillance domain is no exception to this. For example, the extension of sea lanes, and the speed of vessels following them, may change as an effect of new or updated traffic regulations. Hence, future work should investigate appropriate techniques for 1) detecting and handling concept and population drift, and 2) pruning of old and irrelevant data, thereby suppressing computational complexity.

- Investigations of appropriate algorithms for trajectory clustering prior to statistical modelling and anomaly detection.

  During the experimental evaluation, it was found that the cell-based GMM and KDE models did not accurately capture the less frequent patterns in the training data for some data sets. For example, new trajectories following relatively infrequent, yet normal, sea lanes would typically be classified as anomalous. One approach to this problem, which is briefly discussed in this thesis, is to cluster trajectories in the training data prior to the statistical modelling. The clustering should be based on distance or similarity, rather than density, using, e.g., *k-means clustering* (Tan et al., 2006) or *spectral clustering* (von Luxburg, 2007), since the aim is to detect and discriminate different patterns in the training data independently of their relative frequency. For each trajectory cluster, the features of trajectories belonging to that cluster could then be statistically modelled using, e.g., GMM or KDE. During anomaly detection, new trajectories would only be evaluated relative the closest cluster. SNN-CAD and SPT-CAD are inherently more sensitive to the less frequent patterns in training data since they are distance based. Yet, it is quite possible that these algorithms would also benefit from clustering prior to anomaly detection. In particular, it would be interesting to investigate algorithms for online clustering, since clusters may change and new clusters emerge as more training data is accumulated during online learning in CAD.

- Investigations of alternative trajectory dissimilarity measures.

  The directed HD has theoretical properties that are attractive for sequential anomaly detection in incomplete trajectories. Moreover, it has shown promising results during the empirical investigations of this thesis. Yet, it has been argued by others that HD has some general drawbacks, such as sensitivity to noise (Rucklidge, 1996; Atev et al., 2010) and insensitivity to the ordering of points (Alt, 2009; Atev et al., 2010). Various modifications of HD have been proposed to account for its deficiencies (e.g., Atev et al., 2010). Other authors have proposed alternative distance measures to HD, such as the Fréchet distance (Alt, 2009). In contrast to HD, the Fréchet distance does not allow for discontinuities during matching, i.e., two curves should be compared by traversing them both and determine

how close the courses of the two curves stay together. However, it is not clear if these alternative measures are appropriate for sequential anomaly detection in incomplete trajectories. Hence, it would be interesting to further investigate whether the alternative dissimilarity measures, or some extension or adaption of them, share the theoretical properties of directed HD that makes them appropriate for sequential anomaly detection in incomplete trajectories; if they do, empirical investigations should be carried out with the aim of comparing classification performance of SNN-CAD based on the alternative measures and the standard directed HD.

- Investigations of adaptive cell-division algorithms.

  A central parameter of the cell-based GMM and KDE approaches is the size of the cells. In this thesis, we have assumed that cell size is uniform. Moreover, the size of the grid has been set quite arbitrary in the empirical investigations. Yet, it seems reasonable that cell division should depend on the distribution of the data. Indeed, it is quite possible that better performance for cell-based GMM and KDE could be achieved in the experiments using another cell division with, e.g., smaller cell size. Hence, future work related to cell-based modelling should investigate data adaptive algorithms for cell division.

- Investigations of whether HD, or some modification of it, constitutes a valid *Mercer kernel* (Schölkopf and Smola, 2002).

  From a machine learning perspective, it would be interesting to investigate whether HD, or some modification of it, constitutes a valid Mercer kernel. Because if it does, it would enable the use of powerful *kernel methods* (Schölkopf and Smola, 2002) for clustering and classifying trajectories.

6.3 Generalisation to Other Domains

In this thesis, we have proposed SNN-CAD for detecting anomalous trajectories in intelligence and surveillance applications. Yet, SNN-CAD is a general algorithm for anomaly detection that only requires that a dissimilarity measure is specified. Anomaly detection is indeed a ubiquitous problem and we see no principal limitations in applying SNN-CAD, or some other variant of CAD, in other applications, such as fraud, intrusion or fault detection, as long as (normal) data can be assumed to be approximately IID. Moreover, previously proposed algorithms, which have shown good anomaly detection performance for specific applications, may potentially be adopted as nonconformity measures or dissimilarity measures in CAD or SNN-CAD, respectively. That is, CAD and SNN-CAD may serve as wrapper for any anomaly detection algorithm that outputs anomaly scores for individual data points. An advantage of such wrap-

per, compared to the base algorithm, would be the well-calibrated false alarm rate.

6.4 Final Remarks

Detecting abnormal trajectories is important in many surveillance domains, such as maritime surveillance, since these trajectories may correspond to early indications of dangerous, or otherwise interesting, situations. Various algorithms for automated learning and anomaly detection have previously been proposed for assisting analysts in detecting anomalous trajectories. However, as discussed in this thesis, these algorithms typically suffer from one or more issues: Firstly, they are often parameter-laden, which means that they require careful setting of multiple parameters in order to achieve (near) optimal performance; this is an undesirable property for more than one reason (Keogh et al., 2007). Secondly, it may be difficult in practice to balance the sensitivity to anomalies and the false alarm rate, since the anomaly threshold is typically application dependent and not normalised. A possible effect of a badly calibrated threshold is that the false alarm rate becomes too high, which increases the risk that the operator simply ignores all alarms, including true and interesting anomalies. Conversely, if the threshold is unbalanced in the opposite direction, sensitivity becomes unnecessarily low, which increases the risk of missing true and interesting anomalies. Thirdly, many of the algorithms are essentially designed for offline anomaly detection in a trajectory database, where it is assumed that the complete trajectory has been observed before it is classified. This is a limitation in a surveillance application, since it delays the online detection of anomalous trajectories and, thus, the ability to act proactively to impending situations.

In this thesis, we have proposed SNN-CAD, which is a novel algorithm for anomaly detection that has a unique set of properties. SNN-CAD is based on Conformal prediction, which is a relatively new theory in the field of machine learning, and a key property that follows from this is that the false alarm rate is well-calibrated under relatively weak assumptions. That is, the expected rate of false alarms is equal to the specified anomaly threshold $\epsilon \in (0, 1)$ under the assumption that normal data is IID. This is very convenient, since it enables tuning the threshold to an optimal balance between the false alarm rate, for which there may be an upper bound corresponding to an acceptable rate, and the sensitivity, which should be maximised in order to detect as many true anomalies as possible. Apart from the anomaly threshold, the only design parameters in SNN-CAD are the dissimilarity measure, S, and the number of nearest neighbours, k. Assuming that $k = 1$ and that dissimilarity is measured as Euclidean distance in a (normalised) feature space, which are standard in many data mining applications, there is only one free parameter, namely the anomaly threshold ϵ. Hence, SNN-CAD addresses the first and second issues discussed above.

Depending on the application, other dissimilarity measures than Euclidean distance in feature space may be appropriate. In case of anomaly detection in multi-dimensional trajectory data, we have proposed adopting the directed HD as dissimilarity measure in SNN-CAD. This trajectory dissimilarity measure, which is parameter-free, reflects how similar a particular trajectory is to some part of another trajectory. It does not require that trajectories are preprocessed or of equal length, and it can be calculated recursively for each successive data point of the first trajectory. Thus, it is well-suited for online anomaly detection in incomplete trajectories, i.e., it addresses the third issue above.

Results from experimental evaluations on different data sets show that the combination of SNN-CAD and HD is an accurate anomaly detector for trajectory data; compared to other algorithms, good accuracy was achieved without any parameter tuning. Moreover, it has been demonstrated that the false alarm rate is indeed well-calibrated in practise.

References

H. Alt. The Computational Geometry of Comparing Shapes. In H. Albers, S.and Alt and S. Näher, editors, *Efficient Algorithms*, volume 5760 of *Lecture Notes in Computer Science*, pages 235–248. Springer Berlin / Heidelberg, 2009.

H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 13(3-4):251–265, 1995.

F. Angiulli and C. Pizzuti. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, 2002.

S. Atev, G. Miller, and N. Papanikolopoulos. Clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 11(3), September 2010.

S. Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, 3(3), August 2000.

V. Barnett and T. Lewis. *Outliers in Statistical Data*. John Wiley and Sons, Inc., third edition edition, 1994.

M. Berndtsson, J. Hansson, B. Olsson, and B. Lundell. *Planning and Implementing your Final Year Project - with Success!: A Guide for Students in Computer Science and Information Systems*. Springer, 2002.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006.

N. Bomberger, B. J. Rhodes, M. Seibert, and A. Waxman. Associative learning of vessel motion patterns for maritime situation awareness. In *Proceedings of the 9th International Conference on Information Fusion*, 2006.

C. Brax, L. Niklasson, and M. Smedberg. Finding behavioral anomalies in public areas using video surveillance data. In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

C. Brax, L. Niklasson, and R. Laxhammar. An ensemble approach for increased anomaly detection performance in video surveillance data. In *Proceedings of the 12th International Conference on Information Fusion*, pages 694–701, Seattle, July 2009.

C. Brax, A. Karlsson, S. Andler, R. Johansson, and L. Niklasson. Evaluating precise and imprecise state-based anomaly detectors for maritime surveillance. In *Proceedings of the 13th International Conference on Information Fusion*, 2010.

Y. Bu, L. Chen, and D. Wai-Chee Fu, A. Liu. Efficient anomaly monitoring over moving object trajectory streams. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, 2009.

S. Chang. Development and analysis of AIS applications as an efficient tool for vessel traffic service. In *Proceedings of Ocean'04. MTS/IEEE Techno-Ocean: Bridges across the Oceans*, volume 4, pages 2249–2253, 2004.

A. Dahlbom and L. Niklasson. Trajectory clustering for coastal surveillance. In *Proceedings of the 10th International Conference on Information Fusion*, Quebec city, Canada, July 2007.

A. Dahlbom, L. Niklasson, and G. Falkman. Situation recognition and hypothesis management using petri nets. In *Proceedings of the 6th International Conference on Modeling Decisions for Artificial Intelligence*, 2009.

Danish Maritime Authority. Casualty report - collision between Chinese bulk carrier FU SHAN HAI and Cypriot container vessel GDYNIA, 2003.

H. Dee and S. Velastin. How close are we to solving the problem of automated visual surveillance?: A review of real-world surveillance, scientific progress and evaluative mechanisms. *Machine Vision and Applications*, 19(5-6):329–343, September 2008.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, series B*, 39:1–38, 1977.

D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, Inc., 1973.

J. Edlund, M. Grönkvist, A. Lingvall, and E. Sviestins. Rule-based situation assessment for sea surveillance. In B. V. Dasarathy, editor, *Proceedings of the SPIE Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, volume 6242, April 2006.

J. Ekman and A. Holst. Incremental stream clustering and anomaly detection. Technical Report T2008:1, Swedish Institute of Computer Science (SICS), 2008.

E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A geometric framework for unsupervised anomaly detection. In *Proceedings of the Conference on Applications of Data Mining in Computer Security*, 2002.

T. Fawcett. An introduction to roc analysis. *Pattern Recognition Letters - Special issue: ROC analysis in pattern recognition*, 27(8):861–874, June 2006.

F. Fooladvandi, C. Brax, P. Gustavsson, and M. Fredin. Signature-based activity detection based on bayesian networks acquired from expert knowledge. In *Proceedings of the 12th International Conference on Information Fusion*, 2009.

Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Proceedings of IEEE International Conference on Image Processing*, 2005.

A. Gammerman and V. Vovk. Hedging predictions in machine learning. *Computer Journal*, 50(2):151–163, 2007. ISSN 0010-4620.

D. Garagic, B. J. Rhodes, N. Bomberger, and M. Zandipour. Adaptive mixture-based neural network approach for higher-level fusion and automated behavior monitoring. In *Proceedings of the IEEE International conference on Communications*, 2009.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd ed. edition, 2006.

D. Hand. Classifier technology and the illusion of progress. *Statistical Science*, 21(1):1–14, 2006.

D. Hawkins. *Identification of Outliers*. Chapman and Hall, London, 1980.

S. Ho and H. Wechsler. A martingale framework for detecting changes in data streams by testing exchangeability. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), December 2010.

A. Holst, J. Ekman, and S. Larsen. Abnormality detection in event data and condition counters on regina trains. In *Proceedings of the IET International Conference on Railway Condition Monitoring 2006 (RCM 2006), 29-30 November 2006, Birmingham, UK*, 2006.

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

D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, September 1993.

C. Isaksson and M. Dunham. A comparative study of outlier detection algorithms. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 5632 of *Lecture Notes in Computer Science*, pages 440–453. Springer Berlin / Heidelberg, 2009.

F. Johansson and G. Falkman. Detection of vessel anomalies - a bayesian network approach. In *Proceedings of 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2007.

E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005.

E. Keogh, S. Lonardi, C. Ratanamahatana, L. Wei, S.-H. Lee, and J. Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14:99–129, 2007.

J. Kraiman, S. Arouh, and M. Webb. Automated anomaly detection processor. In A. Sisti and D. Trevisani, editors, *Proceedings of SPIE: Enabling Technologies for Simulation Science VI*, volume 4716, pages 128–137, July 2002.

L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition*, 2007.

R. Laxhammar. Anomaly detection for sea surveillance. In *Proceedings of the 11th International Conference on Information Fusion*, pages 1–8, Cologne, Germany, July 2008.

R. Laxhammar and G. Falkman. Conformal prediction for distribution-independent anomaly detection in streaming vessel data. In *Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques*, pages 47–55. Association for Computing Machinery (ACM), 2010.

R. Laxhammar, G. Falkman, and E. Sviestins. Anomaly detection in sea traffic - a comparison of the gaussian mixture model and the kernel density estimator. In *Proceedings of the 12th International Conference on Information Fusion*, pages 756–763, Seattle, USA, July 2009.

J. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the 24th IEEE International Conference on Data Engineering (ICDE)*, pages 140–149, 2008.

M. Markou and S. Singh. Novelty detection: a review - part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, December 2003a.

M. Markou and S. Singh. Novelty detection: a review - part 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521, December 2003b.

M. Masud, J. Gao, L. Khan, J. Han, and B. Thuraisingham. Integrating novel class detection with classification for concept-drifting data streams. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2009.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

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

B. Morris and M. Trivedi. Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. In *Proceedings of IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance*, 2008b.

A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12(3):227–238, 2006.

J. Owens and A. Hunter. Application of the self-organising map to trajectory classification. In *Proceedings of the 3rd IEEE International Workshop on Visual Surveillance*, 2000.

A. Patcha and J. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2007.

C. Piciarelli and G. Foresti. On-line trajectory clustering for anomalous events detection. *Pattern Recognition Letters - Special issue on vision for crime detection and prevention*, 27, 2006.

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

D. Pokrajac, A. Lazarevic, and L. Latecki. Incremental local outlier detection for data streams. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2007.

F. Porikli. Trajectory distance metric using hidden markov model based representation. In *Proceedings of 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2004.

L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, 2001.

B. Rhodes. Anomaly detection and behavior prediction: Higher-level fusion based on computational neuroscientific principles. In *Sensor and Data Fusion*. InTech, 2009.

B. Rhodes, N. Bomberger, M. Seibert, and A. Waxman. Maritime situation monitoring and situation awareness using learning mechanisms. In *Proceedings of the Military Communications Conference 2005*, Atlantic City, NJ, USA, October 2005.

B. Rhodes, N. Bomberger, and M. Zandipour. Probabilistic associative learning of vessel motion patterns at multiple scales for maritime situation awareness. In *Proceedings of the 10th International Conference on Information Fusion*, Quebec, Canada, July 2007.

B. Ristic, B. La Scala, M. Morelande, and N. Gordon. Statistical analysis of motion patterns in ais data: Anomaly detection and motion prediction. In *Proceedings of the 11th International Conference on Information Fusion*, Cologne, Germany, July 2008.

M. Riveiro. *Visual Analytics for Maritime Anomaly Detection*. PhD thesis, Örebo University, 2011.

J. Roy. Anomaly detection in the maritime domain. In C. S. Halvorson, D. Lehrfeld, and T. T. Saito, editors, *Proceedings of the SPIE Optics and Photonics in Global Homeland Security IV*, volume 6945, Orlanda, USA, March 2008.

W. Rucklidge. *Efficient Visual Recognition Using the Hausdorff Distance*. Springer-Verlag New York, Inc., 1996.

S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, second edition edition, 2003.

B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12, 2000.

G. Shafer and V. Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, 2008.

R. R. Sillito and R. B. Fisher. Semi-supervised learning for anomalous trajectory detection. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1034–1044, 2008.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and hall, 1986.

Swedish Maritime Safety Inspectorate. Dry Cargo Vessel OOSTERBRUG - Grounding in Malmö, 2004.

N. Taleb. *Fooled by Randomness: The Hidden Role of Chance in the Markets and in Life*. Penguin Books, 2004.

P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addision-Wesley, 2006.

M. H. Tun, G. S. Chambers, T. Tan, and T. Ly. Maritime port intelligence using ais data. In *Recent advances in security technology*, 2007.

S. Urban, M. Jakob, and M. Pechoucek. Probabilistic modeling of mobile agents' trajectories. In *Proceedings of the International Workshop on Agents and Data Mining Interaction (ADMI)*, 2010.

J. J. Verbeek. *Mixture Models for Clustering and Dimension Reduction*. PhD thesis, University of Amsterdam, Department of Computer Science, 2003.

J. J. Verbeek, N. Vlassis, and B. Kröse. Efficient greedy learning of gaussian mixture models. *Neural Comput.*, 15(2):469–485, 2003. ISSN 0899-7667.

M. Vlachos, G. Kollios, and D. Gunopoulos. Discovering similar multidimensional trajectories. In *Proceedings of the 18th IEEE International Conference on Data Engineering*, 2002.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), December 2007.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387001522.

D. Yankov, E. Keogh, and U. Rebbapragada. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. *Knowl. Inf. Syst.*, 17 (2):241–262, November 2008.

Z. Zhang, K. Huang, and T. T. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proceedings of 18th International Conference on Pattern Recognition (ICPR)*, 2006.

M. Zhao and V. Saligrama. Anomaly detection with score functions based on nearest neighbor graphs. In *Advances in Neural Information Processing Systems 22*, pages 2250–2258. 2009.