Royal Holloway, University of London

The efficiency of conformal predictors for anomaly detection

A thesis submitted in fulfilment of the requirements

for the degree of Doctor of Philosophy

Author:

James Smith

Supervisors:

Alex Gammerman

Chris Watkins

Computer Learning Research Centre

Department of Computer Science,

Royal Holloway, University of London

2016

Declaration

I, James Smith, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed: ..... (James Smith)

Date:

Abstract

This thesis explores the application of conformal prediction to the anomaly detection domain.

Anomaly detection is a large area of research in machine learning and many interesting techniques have been developed to detect 'abnormal' behaviour of objects that do not conform to typical behaviour. Recently conformal predictors (CP) have emerged which allow the detection of the non-conformal behaviour of objects using some measures of non-conformity. Conformal predictors have the advantage of delivering provably valid confidence measures under the exchangeability assumption that is usually weaker than those traditionally used.

The suitability of existing performance criteria for conformal predictors applied to anomaly detection problems is explored. A difficulty in some anomaly detection domains is collecting sufficient examples of anomalies. Two new performance criteria average p-value (APV) and logarithmic average p-value (LAPV) are proposed that do not require labelled anomalies unlike previous criteria. These new criteria allow the discovery of appropriate non-conformity measure for anomaly detection under any setting.

Experiments are conducted with real world data on ship vessel trajectories. A dimensionality reduction package is used and a comparison of a kernel density based non-conformity measure with a k-nearest neighbours non-conformity measure is presented and the results are discussed.

In previous applications of applying conformal prediction to anomaly detection, typically one global class of 'normal' is used to encompass all previous data. However with vessel trajectories there exists an information hierarchy between objects. In this thesis a multi-class hierarchy framework for the anomaly detection of trajectories is proposed. Experiments are conducted comparing the multi-class hierarchy approach to the traditional global class under various conditions and the results are discussed. A study of aggregating p-values from various classes in the hierarchy is also presented. This framework can also

be applied to similar anomaly detection problems where a class hierarchy exists.

Acknowledgements

I'd like to thank my supervisors Alex Gammerman and Chris Watkins for their invaluable suggestions, patience and support throughout this research.

I am extremely grateful to my industrial supervisors Rachel Craddock and Charles Offer from Thales UK, for their thorough support and for the wisdom they have imparted to me. I would also like to thank them and Thales UK for introducing me to AIS data as well as providing me with a dataset to use for my research.

I'd also like to thank Ilia Nouretdinov, Volodya Vovk for providing various insights and discussions regarding conformal prediction.

My sincere thanks and gratitude to my fellow research students and colleagues at the computer science department for the supportive and friendly environment especially Robert Walsh, Tim Scarfe, Jiaxin Kou, Valentina Fedorova, Jamie Al Nasier, Dmitry Adamskiy, Gisela Rossi, Wouter Koolean, Andrej Gregoric and Chenzhe Zhou .

I would also like to thank the Department of Computer Science at Royal Holloway and Thales UK for providing me with the opportunity to carry out this research.

Lastly but most importantly, I would like to thank my deeply supportive and loving parents. Without whom none of this would have been possible.

Table of Contents

Chapter 1

Introduction

This chapter introduces the concepts of *anomaly detection* and *conformal prediction* and discusses relevant previous research. The end of this chapter lays out the contributions and content of this thesis.

1.1 Anomaly detection

In this section we will introduce some of the definitions and techniques previously used for the detection of anomalies. The detection of anomalous events and objects (anomalies) is a crucial and critical task. Anomalies in statistics are commonly known as *outliers* and anomaly detection is sometimes called *outlier detection*. These are objects that deviate considerably from what is typically expected. Anomaly Detection is used by a wide variety of domains for a range of purposes. These domains include but are not limited to:

- Cyber Security [30] and Computer Intrusion Detection [5, 43]
- Fault Detection [3, 38]
- Maritime surveillance [15, 33, 45]

The detection of anomalies is useful in each of these domains as it has the potential to lead to valuable information. For example in fault detection it is important to know if equipment is damaged so that it can be repaired/replaced and so that its task can be resumed as soon as possible. In manufacturing industries, machines need to be running to produce things which in turn are valuable. A problematic/broken machine can cost companies a lot of money. Automated methods for the detection of such anomalous events/objects are beneficial for several reasons: As the cost of computer hardware continues to get cheaper it becomes more financially viable to use algorithms and sensors over human operators. This can potentially lower the amount of work for human operators simply by filtering the data so that it shows the candidates most likely to be anomalies. There is also the potential for automated algorithms to detect faults sooner, in some applications this might be a critical goal. Over time these algorithms may also have direct access to the history of a large quantity of objects which may help give them an edge in doing comparisons.

In anomaly detection there are two classes *normal* and *anomaly*. Most anomaly detectors can be thought of as simple classification algorithms that must categorize a new object as one or the other. An important thing to note is that the exact notation of an anomaly is dependent on the problem. Chandola [4] wrote an extremely detailed review of anomaly detection algorithms and in his own words "The exact notion of an anomaly is different for different application domains".

A simple and common approach to anomaly detection is to produce rules that *normal* observations are expected to follow. New objects will be tested against these rules and if they violate rules they will be treated as anomalies [48]. Rules may be more intuitive to operators, it is clear which rule they violated and these rules could be defined by operators handling the data. Though for more complex problems these require the time of domain experts to setup and potentially maintain. The disadvantage here is that the rules are highly dependent on the exact problem and may not be resilient to changes as the definition of *normal* for that problem may change in the future. It may also be extremely

difficult to create reliable rules for problems with a vast amount of features.

An alternative to rule based systems is to use a data driven approach using examples of *normal* objects. In principle standard classification can be applied. However *anomaly* objects by nature occur less frequently and make up the minority of cases. So it is often inappropriate to model an anomaly as a separate class in standard classification algorithms due to a lack of data. This leads to single-class classification where based on just *normal* observations we test if the object fits in the *normal* objects.

Hawkins [14] provides a statistical definition of an outlier: "An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". A mechanism can be thought of as a probability distribution and a test can be used to determine how likely it is that an object (observation) originated from this distribution. So for statistical approaches to anomaly detection, the *normal* class can be thought of as a probability distribution. The previous observed *normal* examples can be used in a test to determine if new observations belong to the distribution. This leads to a data derived definition of *normal* based on labelled *normal* examples. New observations/objects that have a low likelihood to have originated from the probability distribution of *normal* will be labelled as *anomaly* objects. Hawkins' definition of an outlier is the definition of an anomaly we will use throughout this thesis.

A lot of general techniques, classification techniques, have been applied to anomaly detection problems previously including: Regression, Support Vector Machines, Neural Networks, Bayesian networks, Clustering, Density Estimation, Mixture Models, Parametric and Non-parametric statistical methods.

Data is essential to testing and evaluating anomaly detection algorithms. It is also used to help build the models used in the algorithms. The availability of labelled data is an essential property to consider when developing algorithms. In some domains and problems the data may be plentiful and in others it may be difficult to acquire. Most data is often labelled manually by a domain expert to ensure it is accurately labelled. In light of this some algorithms are designed to operate with lots of prior labelled data and others

are more flexible. This property is known as the setting. There are three common settings for anomaly detection algorithms:

- The *supervised setting* - In the supervised setting the real labels are known and they are revealed to the algorithm immediately after it has classified the object. This can be problematic as getting sufficient anomalies can be difficult as these are typically hand collected by a domain expert. This however leads to the best accuracy as it ensures that if an example is found to be a *normal* object it can safely be added to previous examples.

- The *semi-supervised setting* - In the semi-supervised setting there are only labelled examples available of the *normal* label. These labelled examples of *normal* make up the training set. This setting does not need labelled anomalies. It should be noted that this is a different definition to what is typically used in machine learning [49] but this definition is specific to anomaly detection [4] .

- The *unsupervised setting* - In the unsupervised setting the real labels are unknown. Algorithms designed for this setting typically make the assumption that anomalies will make up a small proportion of the dataset and that the majority of objects we start with belong to the *normal* class.

In this thesis we are concerned with using statistical models utilizing historic data to predict the likelihood of an object being anomalous. A significant amount of research has already been done in this domain and in this thesis we aim to extend the prior work.

1.1.1 Maritime surveillance domain

In this thesis in our experiments we study the domain of maritime surveillance. The maritime surveillance domain is concerned with the behaviour of ships (vessels). The aim is typically to detect anomalous trajectories such as: Sudden stops and starts in unusual locations, deviations from typical journeys, speeding or travelling the wrong direction in

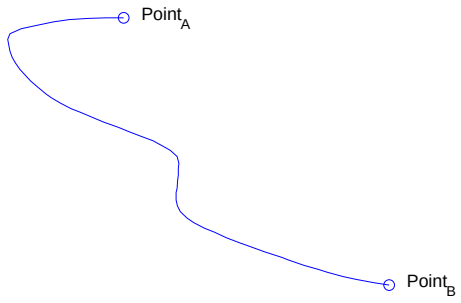

Figure 1.1: Example of a trajectory

a sea lane. In this thesis a trajectory can be thought of as a journey between two points (ports): It has a start and end point and a path it takes.

For maritime surveillance data in real world problems AIS data is commonly used.

1.1.1.1 Automatic Identification System (AIS) data

Automatic Identification System (AIS) is a system for tracking vessels. Vessels are equipped with an AIS broadcast system alongside a GPS (Global Positioning System) device. The location of the vessel is retrieved by an on-board GPS receiver. The AIS system will then make regular broadcasts. The broadcasts contain the following information on the vessel: unique id known as a Maritime Mobile Service Identity MMSI, latitude, longitude, speed, heading, timestamp and other less useful bits of information (status codes). Not all vessels are mandated to use AIS but all passenger and tanker vessels are required to. As well as vessels weighing 300 gross tons or more are required to do so under the International Maritime Organisation's (IMO) regulations [29].

One of the large problems in this domain is the lack of publicly available datasets. The international body that regulates the AIS specification decided that the publication of AIS data undermines the safety of security of ships and ports and as such condemned its public

distribution.[1]. However it is still possible to get access to AIS data as anyone with an AIS receiver can pick up the data and record it. In this thesis the AIS data was provided to me by Thales UK. There also exist several databases on the internet that contain further information about the ship which can be looked up by using a vessel's MMSI number. These details include the ship type, pictures and size of the ship.

The broadcasts produced by AIS systems are broadcast every few seconds while vessels are moving allowing for a suitable amount of data to accurately track vessels as they conduct their daily business.

1.1.1.2 Prior research of anomaly detection in the maritime surveillance domain

Anomaly detection in the maritime surveillance domain has had a substantial amount of previous research. Most previous research uses data acquired through AIS broadcasts or radar data.

The main areas of research seem to be: expert systems incorporating in domain specific knowledge, high level systems and algorithms development which involves constructing an appropriate feature model of the trajectory. Most algorithms are typically symbolical rule reasoning, statistical and/or using machine learning to create a model of normality.

Rivero et al. [36,37] have explored using visualisation of vessel trajectory anomalies to aid an operator.

A reasonable amount of research has investigated rule based systems. Nilsson and Laere et al. [28, 45] have consulted with domain experts to produce rule based systems that use rules to model normality and flag violations as anomalies. Rhodes, Siebert, Bomberger et al. created SeeCoast [33,39] which is a rule-based system extending the US coast guard's security and monitoring system. Firstly by fusing AIS and radar data then applying rules. Brax et al. [2] implemented a multi-agent system utilizing a rule based approach in which anomalies are represented by agents in a supervised setting with an

[1] http://www.imo.org/en/OurWork/Safety/Navigation/Pages/AIS.aspx

operator. Holst et al. [15] proposed a system to merge statistical and symbolic methods that are both interchangeable in the formation and understanding of rules.

Bomberger [1], Gargeic [9] & Rhodes [32, 34] have all implemented and investigated using various variants of neural networks to classify maritime trajectories as normal or anomalous. Laxhammar [18] has explored the use of Gaussian mixture models to detect anomalous trajectories. Ristic [35] investigated using kernel density estimation. Later Laxhammar compared using Gaussian mixture models and kernel density estimators [23]. Kowalska et al. [16] use a Bayesian approach with Gaussian processes combined with active learning outputting a measure of normality for a trajectory. Bayesian networks have also been explored by Mascaro et al. [26] and Lane et al. [17].

The use of Hidden Markov Models has also been explored by Du Toit et al. [6] and Shahir et al. [40]. Support vector machines have also been explored by Handayani et al. [12]. Perera et al. [31] explore the use of neural networks to detect and track vessels.

One shortcoming with these methods is they tend to directly output whether a trajectory/vessel behaviour is either an *anomaly* or *normal*. Most do so without outputting a reliability factor of their predictions, making it difficult to understand whether a single prediction is trustworthy. In order to make predictions that offer a measure of reliability of their predictions we use conformal predictors.

1.2 Conformal Predictors

Conformal predictors [47] are a machine learning technique that provides predictions with a provably valid measure of reliability (confidence) in either classification or regression problems. These techniques output a set of labels for a given object, so that the object is likely to belong to one of the labels in the set subject to a conditional probability.

In conformal prediction we consider the classification case in which we have the set of all possible objects (the object space) $\mathbf{X}$ and the set of all possible labels (the label space) $\mathbf{Y}$. We assume that reality outputs a pair $\mathbf{Z} = \mathbf{X} \times \mathbf{Y}$ subject to a probability distribution

P.

Conformal predictors assume all pairs originate from the same probability distribution P and that the probability of a pair appearing is independent of the previously outputted pairs. This is the independently and identically distributed data (i.i.d). assumption.

For any label y there exists a subset of $\mathbf{X}$ that belongs to that label. In order to calculate the likelihood that a new object x belongs to the label y we must compare it against all objects previously known to belong to y. To achieve this we calculate the p-value $p_{n+1} = (x_{n+1}, y)$. The p-value can be understood as an upper estimate of the probability of x_{n+1} appearing, assuming it belongs to y, this is called a conditional probability.

The performance of this method depends on the selection of a *Non-Conformity measure (NCM)* denoted as A – that is a sort of information distance between an object and a set of the same type objects where the objects are taken together with their labels.

Basically, the method tests whether a *new object* $\mathbf{X} \times \mathbf{Y}$ might be generated by the same distribution as the previous *(training)* objects $z_1, \ldots, z_{n-1}$. If produced *p-value* $p_n = p_n(\mathbf{Z}_n)$ is small, then the hypothesis of the new object's belonging to the class is likely to be rejected.

Another setup of the conformal prediction problem is to use a *significance* level instead of outputting a prediction set. The *significance* denoted by ϵ regulates the pre-determined level of confidence. The confidence of the prediction is $1 - \epsilon$. According to the validity property [47] of conformal prediction, if all the data objects $z_1, \ldots, z_n$ are really generated by the same distribution, then the probability that $p_n \leq \epsilon$ is at most ϵ. This means that if we are 99% confident in a prediction $\epsilon = 0.01$ the p-value for an object must be 0.01 or greater for it to belong to the class. In the context of anomaly detection this means that if z_n is not an anomaly, it will be classified as anomaly with probability at most ϵ.

1.2.1 Validity

The validity property [47] of conformal prediction is an important characteristic of a conformal predictor. The validity principle states that the p-values are valid if the assumptions

are met. A p-value for a given class y is the probability of an object z_n belonging to class y assuming it belongs to class y. The null-hypothesis is that the object belongs to class y. This is useful as if our new example z_n does belong to label y there is only a ϵ probability that it is incorrectly predicted as not belonging to class y. This has been proven in the supervised setting [47].

1.2.2 Efficiency

The validity property ensures the correctness of the predictions if the object belongs to the class y but makes no guarantees if the object does not truly belong to the class y. In essence it is desirable to ensure that objects that do not belong to class y are given small p-values for class y. In the set predictor mode this can be achieved by minimising the number of outputted classes for an object. This is known as *efficiency* [47]. The smaller the prediction set the more efficient the predictor. When creating a non-conformity measure it should be as efficient as possible as this will lead to the best performance. In the case of anomaly detection: if z_n is an *anomaly*, we wish this to be captured by our non-conformity measure assigning a small p-value to it. The following chapter 2 introduces and explores various measures of *efficiency*.

1.3 Conformal Prediction applied to Anomaly Detection

There are a few formalisations of how the classes can be constructed in the anomaly detection setting. In the application domain there are three possible terms associated with an object: *normal*, *abnormal* & *anomaly*. A 'normal' object is one that is typical, an *abnormal* object is an object that deviates from typical behaviour but it is still considered part of *normal* behaviour. An 'Anomaly' is an object that deviates from typical behaviour but is not considered part of *normal* behaviour. One of the major advantages of using

conformal prediction is that it allows the calibration of the false-positive rate as the expected false-positives will be ϵ [19] assuming the assumptions are met and the supervised setting is used. In this thesis we consider anomaly detection under the independently and identical distribution (i.i.d.) assumption.

Therefore we use two labels: 'normal' and 'anomaly', here are a few ways to construct the anomaly detection problem into a classification problem.

1.3.1 Single 'Normal' class

In the single class 'normal' predictor, we require prior data that is considered to be *normal*. A new object is compared against prior *normal* objects and a prediction is made if it belongs with the distribution of *normal* objects. If the object does not belong it is classed as *anomaly*. This is what we'll refer to in this thesis when we refer to single-class anomaly detection.

1.3.2 Binary Class

The simplest case is representing the problem as a two-class problem, this requires having labelled data for both *anomaly* and *normal*. When predicting the label of a new object it is compared with all the examples from the *normal* class and a p-value is calculated. The same is done for the *anomaly* class. This requires a substantial amount of anomalies and it could well be the case that anomalies are not all generated by the same distribution. In most anomaly detection problems this is inappropriate.

1.4 Conformal Anomaly Detection

Conformal Anomaly Detection (CAD) is an extension of Conformal Prediction that focuses on anomaly detection in the unsupervised and semi-supervised settings proposed by Laxhammar [21]. In conformal prediction the goal is to predict the corresponding label of an object, however in conformal anomaly detection the problem is determining if the new

Conformal Anomaly Detection

Input : Non-Conformity Measure A, significance level ϵ, training objects $z_1, z_2, ..., z_{n-1}$ and new object z_n

Output: P-value p_n, boolean variable Anomaly

$D = \{z_1, ..., z_n\}$

for $i \leftarrow 1$ **to** n **do**
| $\alpha_i \leftarrow A(D \setminus z_i, z_i)$

end

$\tau \leftarrow U(0,1)$

$p_n \leftarrow \frac{|\{i:a_i>a_n\}|+\tau|\{i:a_i=a_n\}|}{n}$

if $p_n < \epsilon$ **then**
| $Anomaly_j \leftarrow$ **true**

else
| $Anomaly_j \leftarrow$ **false**

end

Figure 1.2: The single-class conformal anomaly detector

object is *normal* or *anomaly*. CAD focuses on the single *normal* class, but this can be expanded to use multiple *normal* classes. A new object is evaluated and for each label, a p-value is calculated determining how well the object fits in with the observed samples of the distribution associated with the label. The CAD algorithm is shown in fig. 1.2.

1.5 Prior research and research focus

Rikard Laxhammar [19] has conducted extensive research with applying conformal predictors to anomaly detection and the maritime surveillance domain. Firstly the proposal of conformal anomaly detection and secondly proposing two non-conformity measures suited

for the maritime domain. This thesis aims to further the research in both the application of conformal predictors to anomaly detection and its application to the maritime surveillance domain.

In his research Laxhammar proposes the use of a Hausdorff distance k-nearest neighbours non-conformity measure [19, 21]. In chapter 4 this non-conformity measure is used due to its good suitability. In my study of efficiency measures in chapter 2 the Hausdorff distance is utilized as an input to apply dimensionality reduction.

In essence Laxhammar's prior research focuses on the introduction of conformal prediction to the anomaly detection problem as well as its application to the maritime domain in which he also proposes two non-conformity measures.

This thesis aims to explore efficiency measures and their appropriateness to anomaly detection problems. Appropriate efficiency measures for anomaly detection are essential for the creation of suitable non-conformity measures. In Laxhammar's work he primarily focuses on using traditional classification metrics. This is explained and explored in depth in chapter 2.

Another area of interest is understanding the dependency and relationship between the classes in the maritime domain. Laxhammar did conduct some experiments using vessel types as classes but there are no experiments comparing the effectiveness between one representation and another or attempting to exploit this structure.

1.6 Thesis structure

Here is a summary of the upcoming chapters and what each addresses.

1.6.1 Efficiency Review

In chapter 2 we review previous conformal prediction efficiency measures and discuss their suitability for the anomaly detection domain. As well as their applicability to conformal anomaly detectors.

1.6.2 Efficiency Applications

In chapter 3 we empirically study the suitable measures of efficiency for conformal prediction and anomaly detection. The results are then discussed.

1.6.3 Multi-Class Hierarchy Chapter

In chapter 4 we investigate what happens when a hierarchy of overlapping classes is used to represent the normal data.

1.6.4 Conclusion

Chapter 5 wraps up the thesis, highlighting the key findings, a discussion on applications of the research is also presented.

1.7 Main Contributions

The following are the main contributions of this thesis:

- Two new efficiency criteria that are dedicated to anomaly detection. These also allow measuring the efficiency without the need for labelled *anomaly* objects unlike previous criteria. Furthermore a thorough analysis of the suitability of previous efficiency criteria for anomaly detection is presented.
- A Multi-Class Hierarchy approach that provides the potential to achieve better performance by utilizing a hierarchy of classes. Typically previous conformal prediction anomaly detection methods focused on using a single global class or multiple classes but without an overlapping hierarchy.

1.8 List of publications

The following is a list of publications that were published in the pursuit of this thesis:

- Anomaly Detection of Trajectories with Kernel Density Estimation by Conformal Prediction [41]
- Conformal Anomaly Detection of Trajectories with a Multi-class Hierarchy [42]

Chapter 2

The performance of Conformal predictors in the Anomaly Detection context

Well designed performance measures are critical for identifying and understanding appropriate algorithms for a given problem. In this chapter our goal is to identify performance measures suitable for the application of conformal predictors to the anomaly detection problem. We introduce and explore commonly previously used performance measures for conformal predictors and binary classification problems. It turns out that there is room for improvement and later in this chapter we propose our own performance measures to address these shortcomings. The following chapter 3 deals with the experimental evaluation of these measures.

2.1 Introduction

In this section we introduce the commonly used performance measures for anomaly detection and conformal prediction from literature. We introduce the basics as this is necessary to build upon in the later discussions. Most of these measures are also used in later chapters of this thesis in the experimental evaluation of various techniques. We start with a short introduction on what a performance measure is.

A performance measure provides a method for comparing the performance of two or more approaches. They allow us to understand which of these approaches is the better solution. Typically many performance measures can be applied to a problem, however each of these measures encouraged different kinds of behaviour. It is critical to understand the behaviour that each measure encourages so that an appropriate performance measure is used for evaluating different approaches.

There are two main categories of performance measures that can be used with conformal predictors applied to anomaly detection:

- Binary classification performance measures - Anomaly detection can be modelled as a binary classification problem, with the labels *normal* and *anomaly*. This allows the use of all binary classification performance measures. We discuss these measures further in section 2.1.1

- Conformal prediction efficiency measures - Traditionally conformal prediction is primarily used as a multi-label set predictor. In reality objects have only one label associated with them and not a set of labels. These measures typically focus on achieving a prediction set size of 1 for every object. The size of these prediction sets is known as the *efficiency*. An example of an inefficient prediction is one that outputs the set of all labels for any given object. Several methods have previously been proposed to calculate the efficiency of predictions and these can be utilized as performance measures. We explore and discuss their suitability in section 2.1.2.

2.1.1 Binary classification performance measures

In this section we discuss binary classification performance measures and several approaches that have previously been used in literature. We start by formulating the binary classification problem.

In binary classification there are two labels namely *positive* and *negative*. The predicted label is the label outputted by the classification algorithm. The true label of an object is the label that belongs to the object. The goal of classification algorithms is to predict the true label for every object given only the object.

Binary classification measures start by comparing the true label of an object against the predicted label. Here we define the basic notation. If the predicted label and true labels match it will either be a true positive (tp) or true negative (tn). It is true positive if the labels are *positive* or a true negative if the labels are *negative*. If the predicted label and true labels do not match it will be either a false positive (fp) or false negative (fn). So if the labels of the predicted label and true label don't match, a false positive occurs if the true label is *negative* or a false negative occurs if the true label is *positive*. This is summarised in table 2.1.

		True Label	
		Positive (anomaly)	**Negative (normal)**
Predicted Label	**Positive (anomaly)**	true positive (tp)	false positive (fp)
	Negative (normal)	false negative (fn)	true negative (tn)

Table 2.1: Classification outcome interpretation

The false positive rate (fpr) is the percentage of objects belonging to the negative label that are predicted as positives (false positives). The true positive rate (tpr) is the

percentage of *positive* objects correctly being predicted as positive. It is desirable for any binary classification algorithm to maximize the true positive rate and minimize the false positive rate. Conversely it is desirable to minimize the false negative rate and maximize the true negative rate. In the case where the fpr is 0 and the tpr is 1 perfect classification performance has been achieved.

In the anomaly detection context we will use the positive label to be the *anomaly* label and the negative label to be the *normal* label. The false positive rate is the percentage of normal objects misclassified as anomalies, and the true positive rate is the percentage of anomalies that are correctly predicted.

Typically binary classification algorithms calculate a score for objects they are classifying. This score is then checked against a threshold parameter.

This threshold then determines what values of the score will be predicted as either *positive* or *negative*. Altering this parameter affects the performance of the algorithm. Extreme values lead to both the tpr and fpr rates being either relatively high or low. Conformal prediction makes use of such a threshold namely the significance parameter ϵ.

Receiver Operating Characteristic (ROC) curves [10] provide a method to visualize and compare the classification performance using such a parameter. They plot the true positive rate against the false positive rate while varying the parameter. Fig 2.1 shows an example of a ROC curve. It shows two classifiers one marked with the red curve and one marked with the blue curve. The black dashed line shows the expected result if predicting randomly. Both classifiers perform better than predicting randomly. The classifier represented by the red curve also performs better than the classifier represented by the blue curve. The closer the classifier to the top left the better. This is equivalent to maximizing the true positive rate for low false positive rates.

The Area Under the ROC Curve (AUC) [13] is used as a measure to summarize the ROC curve into an objective single scalar across all possible threshold values. The larger the AUC the better the overall performance for all possible values of the threshold parameter. So we can say that if a classifier has a larger AUC than another that its performance in

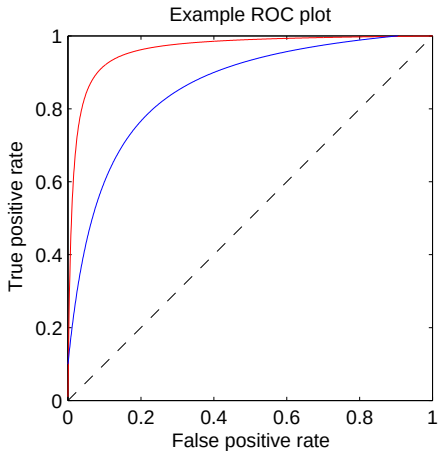

Figure 2.1: Example ROC

general across all threshold parameter values is better. A classifier that predicts randomly will converge to an AUC of 0.5. An AUC of 1 indicates a perfect classifier as the *true positive rate* is always 1, and an AUC of 0 is the worst possible classifier. Typically however a threshold is chosen for an application specific reason that comes from the problem specification.

ROC and AUC are both widely used in the study of binary classifiers.

AUC does have its disadvantages. Hand [11] indicates that it is not always the best measure. Recall that AUC is calculated across all parameter thresholds and is an indicator of overall performance. If an algorithm has a better AUC than another algorithm it does not mean it is better for every possible threshold parameter value; This is true for cases where the ROC curves of two classifiers cross.

Figure 2.2 shows an example where two ROC curves cross. The red curve has an AUC of 0.675 and the blue curve has an AUC of 0.665. By AUC alone the red curve is a better choice due to its higher AUC value, however we can see that the classifier represented by the blue curve performs better for small false positive values particularly 0.1 .

AUC also assumes that the error cost of mis-classifying the positive and negative

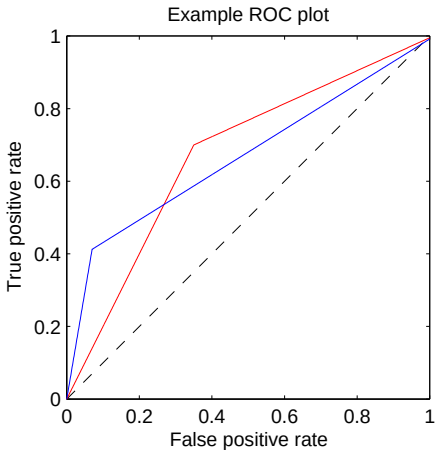

Figure 2.2: Example ROC of two curves that cross

labels are the same. For different applications the cost of either of these errors may well be different. In industry there are typically separate requirements put on the amount of acceptable errors for both types. For anomaly detection the cost is indeed different between the two errors. Missing an anomaly can be far more costly than mis-classifying a normal object as an anomaly. Hand proposes an alternative approach using his proposed measure, *H measure* [11], addressing these concerns. H-measure takes a weight function as an input that specifies the cost between the two error types. It seeks to do better than AUC in all possible cases. In their paper [11] they also offer an extensive discussion contrasting H measure to AUC.

Fawcett et al. [7] suggest a computationally efficient framework for calculating ROC and AUC that we use throughout this thesis whenever calculating AUC or pAUC in the later experimental chapters.

2.1.2 Conformal prediction specific performance measures

Conformal predictors can be treated as binary classification algorithms, but they also have their own properties. These properties offer additional methods to measure their perfor-

mance. Firstly we introduce and discuss these properties in this section. In sections 2.1.2.1 to 2.1.2.6 we introduce the measures that use these properties and in section 2.1.2.7 we discuss their relation to each other and compare them.

Conformal predictors output p-values alongside predictions as introduced in section 1.2. In the case of classification, when classifying a new object a conformal predictor assigns a p-value for each label. The p-value is used as a likelihood of the object belonging to that label. Typically the p-values are then used to output a set of labels. The outputted set of labels depends on the significance ϵ, namely that the prediction is $(1 - \epsilon)$ confident that the correct label will be in the prediction set. This is the property of *validity* and it is not useful to measure as it is proven that in the online setting the error rate converges to the significance ϵ. It has also been empirically shown to hold in the offline setting for a range of problems.

In reality every object has only one true label associated with it, not a set of labels. So it is desirable to have just one label predicted for a given object. The size of the prediction set is known in conformal prediction theory as *efficiency*, and this property can be measured. The more efficient the conformal predictor the better its performance is. A more efficient predictor has a lower chance of objects being predicted as labels that are not their true label.

The potential problem with minimizing the size of the prediction set is that it is possible to predict the empty set (no labels). This occurs for an object where the p-value for every label is smaller than ϵ. However this is undesirable as we know that every object belongs to a label. It turns out this is mitigated as the validity property of conformal predictors ensures that the error rate and thus the upper limit for empty-set predictions converges to ϵ. So even if the goal is to lower the p-values then the error rate is still valid.

Here we briefly discuss what the desirable values of p-values are. The p-values of objects corresponding to the true label are uniformly distributed between 0 and 1. Therefore they have an average of 0.5 provided they originate from the true label. To prevent multi-label predictions p-values should be smaller than the significance ϵ for labels that are not the

true label. Though ϵ is set depending on the requirements of the problem.

In anomaly detection applications, conformal predictors are used as a single-label predictor. An object is predicted as *normal* if its p-value for the *normal* label is greater than or equal to ϵ, otherwise the object is classified as an *anomaly*. We are therefore interested in performance criteria that support a single label.

Recently a paper appeared by Vovk et al. [46] that studied and compared previously used efficiency criteria for conformal predictors. This recent paper is the only review of efficiency measures for conformal predictors. It covers all the known efficiency measures, but does not discuss their suitability for anomaly detection. In the next sections we will introduce and discuss their usefulness for anomaly detection. In sections 2.1.2.1 to 2.1.2.7 we introduce and review these existing efficiency measures. In section 2.2 we discuss their suitability for anomaly detection.

Firstly here we introduce the pre-existing measure of efficiency. Here we provide some notation that will use throughout this section for the definitions of the measures. Firstly to evaluate an algorithm we use a data set $\{x_1, x_2, ..., x_n\}$ that consists of two parts: the training set $T = \{x_1, x_2, ..., x_l\}$ and the testing set $\{x_{l+1}, x_{l+2}, ..., x_n\}$. Γ_i^ϵ denotes the set of labels outputted by the conformal predictor for object i from the data set and the given significance ϵ. By p_i^y, we denote the p-value of object x_i for label y.

- **2.1.2.1 *S-Criterion* Sum of p-values criterion**

  The sum of p-values is a performance criterion that was proposed by Fedorova et al. [8] for classification. It is calculated by generating p-values for all objects in a testing set, and for all labels and summing them together. The smaller the sum the more efficient the predictor.

$$\frac{1}{n-l} \sum_{i=l+1}^{n} \sum_{y \in Y} p_i^y \tag{2.1}$$

  By favouring outputting smaller p-values across all the labels it is likely that fewer labels will be predicted for each object therefore the efficiency is better.

- **2.1.2.2 *N-Criterion* Size of prediction set criterion**

The size of the prediction set criterion is equal to the average number of predicted labels for each object in the testing set.

$$\frac{1}{n-l}\sum_{i=l+1}^{n}|\Gamma_i^\epsilon| \tag{2.2}$$

This approach is dependent on ϵ. This is great for cases when a suitable ϵ is already chosen. However this is problematic to use in the case where an ϵ has not been chosen, as it will need to be evaluated for multiple values of ϵ.

As an example the size could be taken at the minimum p-value $\epsilon = \frac{1}{n+1}$ to give an idea for the worst possible performance. However ϵ is typically set depending on the requirements of the application.

- **2.1.2.3 *U-Criterion* Unconfidence criterion**

Unconfidence is the average second largest p-value over all labels. Smaller unconfidence values indicate better efficiency. The U-Criterion is built on the assumption that the feature space is split up into disjoint labels and therefore only one p-value per object should be high. This measure was proposed in [8].

$$\frac{1}{n-l}\sum_{i=l+1}^{n}\max_{y'\neq y}p_i^y \tag{2.3}$$

- **2.1.2.4 *F-Criterion* Fuzziness criterion**

The fuzziness criterion is the average sum of the p-values for all labels per object except the largest p-value. This uses the same idea, namely the unconfidence. If there are only two-labels this is equivalent to the unconfidence efficiency measure.

$$\frac{1}{n-l}\sum_{i=l+1}^{n}\Big(\sum_{y}p_i^y-\max_{y}p_i^y\Big) \tag{2.4}$$

- **2.1.2.5 *M-Criterion* Multiple criterion**

  This is the percentage of objects that are predicted to belong to multiple labels, i.e. have a prediction set that is larger than one for a given ϵ.

$$\frac{1}{n-l}\sum_{i=l+1}^{n}||\Gamma_i^p|>1| \tag{2.5}$$

- **2.1.2.6 *E-Criterion* Excess criterion**

  For a given ϵ, excess criterion is the average size of the prediction set for sets that are larger than one. It assumes that the prediction set should contain at least one label per object.

$$\frac{1}{n-l}\sum_{i=l+1}^{n}\max\{|\Gamma_i^p|,1\}-1 \tag{2.6}$$

  If this is applied to a two-label problem, this criterion is equivalent to the multiple criterion efficiency measure.

The above are all applicable to classification problems. An alternative formulation for measuring efficiency for regression was explored by Lei and Wasserman [25]. They proposed measuring the size of the prediction region in regression problems using Lebesgue measure. This is the analogue of the N-Criterion for regression.

2.1.2.7 Summary of efficiency measures

The criteria described above are not appropriate for all circumstances. Table 2.2 provides detail about what is required to use each measure. The requirements include how many labels are required and whether they are dependent on ϵ. Those that are dependent on ϵ require testing multiple values of ϵ if ϵ is likely to change in the final application. This helps create a more accurate efficiency score.

In the two label case, the fuzziness criterion outputs the same result as the unconfidence criterion. The multiple criterion outputs the same result as the excess criterion in the two label case. The fuzziness and excess criteria are both redundant if there are not at least 3

Efficiency Measure	**Required minimum number of labels**	ϵ dependent
Sum of p-Values	1	No
Size of prediction set	1	Yes
Unconfidence	2	No
Fuzziness	3	No
Multiple	2	Yes
Excess	3	Yes

Table 2.2: Comparison of efficiency measures

labels. Therefore these two criteria are not applicable for the two label anomaly detection application. We primarily focus on the single-label case for which only the sum of p-values (S-Criterion) and size of the prediction set (N-Criterion) are applicable.

2.2 Performance measures applied to anomaly detection

In this section we discuss suitability of the previously introduced performance measures for the anomaly detection context. We start by briefly commenting on applying the binary performance metrics and shift to a more in depth discussion on the efficiency metrics. Lastly we discuss the challenges that all these methods face when selecting training and testing data.

As anomaly detection is already a binary problem all the binary metrics can be applied to anomaly detection. However the key difference is that anomalies by nature occur at a low frequency. We typically wish to catch all the anomalies but this has a trade-off of causing more false-positives. Partial AUC can be used to provide a more useful AUC value for anomaly detection.

Partial AUC (pAUC) is a modification to AUC that calculates the area between two false positive rates of the ROC curve [27]. The pAUC value is then normalised so that its outputs are in $[0, 1]$. This is useful for applications in which we are only concerned with performance for a subset of the false positive rates. This has been used in the following applications: ranking, biometric screening and breast cancer detection [27].

In Laxhammar's thesis he suggests that anomaly detection is one of these applications [19]. He argues that in anomaly detection problems, performance is most important at small false positive rates. There are many reasons for this, for example in practical applications the threshold is likely to be set to a value that ensures a small false-positive rate. This is because the rate of anomalies appearing in the dataset is expected to be very low (less than 1%).

For example if the false positive rate is 1% and the rate of anomalies in the dataset is 1%, the number of objects predicted as an *anomaly* that have the true label *anomaly* will generally be 50% of the objects predicted as an *anomaly*. As such the suggested false positive rate range of interest for pAUC is $[0, 01]$ as suggested in Laxhammar's thesis [19] for anomaly detection problems.

Conformal prediction is a framework that wraps around a non-conformity measure. The non-conformity measure is what affects the performance of the conformal predictor and as such when discussing the performance of a conformal predictor we are discussing the performance of the non-conformity measure the conformal predictor is using. The major advantage of using conformal prediction for anomaly detection is that the number of false-positives converges to ϵ. This allows the operator to set the approximate number of false-positives they are prepared to accept.

Anomaly detection is a form of classification and so most efficiency measures and the reasoning behind them are applicable to anomaly detection. The anomaly detection problem is different to standard classification in the conformal prediction context. It is setup as follows. We assume that objects from the *normal* label are produced from a independent and identical distribution. When using conformal prediction *anomaly* is a

label but we do not produce a p-value for it, or attempt to model it. There are two reasons for doing this. Firstly for some anomaly detection problems there are difficulties retrieving anomaly objects in sufficient quantities to represent all possible anomalies as a label. Secondly for most applications it is extremely unlikely that anomalies are produced from an independent and identical distribution or exchangeable distribution. This means conformal prediction is not able to offer a valid p-value for the *anomaly* label and this is why we only produce a p-value for the *normal* label.

The *anomaly* label is predicted instead of predicting an empty-set (no-label). Therefore if the object is not predicted as *normal* it is predicted as *anomaly*. Although there are two labels it is only beneficial to measure the efficiency of the *normal* label. This is because only *normal* data from the training set that we predict is used to predict and the efficiency of the *anomaly* label is the direct opposite of the efficiency of the *normal* label. Therefore the only pre-existing efficiency measures that are applicable are those suited for a single label. These are the sum of p-values (S-Criterion 2.1.2.1) and the size of the prediction set (N-Criterion 2.1.2.2).

The only prior work that uses conformal prediction with anomaly detection that uses efficiency measure is a paper by Laxhammar [20]. In Laxhammar's work with anomaly detection and conformal prediction, one of his papers empirically studies efficiency [20]. In his paper Laxhammar conducts an experiment using three different ship types (cargo, passenger and tanker) as labels with an empty set prediction being treated as an anomaly. The mean and median size of the prediction set (N-Criterion) and the number of multiple and empty predictions (M-Criterion) are all used to measure the efficiency to investigate the non-conformity measure proposed in his paper. The M-Criterion is not applicable to the single-label anomaly detection case as it requires at least two labels.

Here we discuss in detail the pre-existing efficiency criteria suitable for anomaly detection.

The N-Criterion is dependent on ϵ. For a given ϵ the N-Criterion prioritizes predicting as little of the testing set as *normal* as possible. Intuitively this seems bad as the testing

set contains *normal* objects, and we want them to be predicted as *normal*. However recall that the validity property of conformal predictors ensures that $(1-\epsilon)\%$ of the *normal* objects in the testing set are predicted as *normal*. This actually favours decreasing the number of *anomaly* objects predicted as *normal* and the ratio of objects predicted as *normal* converges to $(1-\epsilon)\%$.

Now we discuss applying the sum of p-values (S-Criterion) to anomaly detection. Unlike the N-Criterion it does not depend on ϵ. The sum of p-values (S-Criterion) favours minimizing the p-values for all objects in the testing set. However in anomaly detection applications, anomalies will typically make up a small amount of the testing set, so the majority of objects belong to the normal label. We wish for the normal label to have high p-values and anomalies to have low p-values. As the testing set is dominated by normal label objects we expect their average p-value to converge to 0.5 due to the uniform distribution of p-values for objects which belong to the *normal* label. The usefulness of the S-Criterion for measuring anomaly detection performance is therefore dependent on the proportion of *normal* objects to *anomaly* objects. The more *anomaly* objects it contains the better but by their nature, the number of anomalies will be few.

Vovk et al. [46] explored various efficiency measures and argue for using a special class of efficiency measures called *probabilistic*. A probabilistic efficiency measure is one that it has been proven that the idealized conformity measure is optimal for. Both the S and N criteria are *probabilistic* as proven in [46].

For applications where ϵ may change the S-Criterion may be a better choice. Both of these criteria require the use of labelled anomalies in order to gauge performance. These labelled anomalies provide a testing set which is used to evaluate performance.

In summary both the S and N criteria are applicable to anomaly detection. The S-criterion has the added benefit of being ϵ independent. However both of these measures require the use labelled examples of anomalies. Earlier we established that it can be difficult acquiring reliable labelled anomalies and in these circumstances these are not ideal.

One of the challenges of designing any anomaly detection algorithm is selecting the training set and testing sets used to evaluate it. If a testing set is not properly selected it could lead to promoting a sub-optimal classifier in real world applications. A testing set needs to be representative of the real data to be reliable. It would be beneficial to avoid the possibility of poor performance against a specific type of anomaly that wasn't in the datasets. Failure to address these concerns could lead to a classifier being promoted.

Anomalies by their nature rarely occur. For several applications it may be a challenge to collect and label examples of anomalies. It is also difficult to collect anomalies that reflect all possible anomaly objects. It may be costly for a domain expert to accurately label a dataset as they would need to search large volumes of *normal* data to ensure a sufficient quantity of anomalies are being labelled. This is because most of the objects in the data will belong to the *normal* label. An example would be monitoring a road surveillance camera looking for accidents. There will be hours and hours of footage to search.

An alternative to collecting real anomalies is to generate artificial anomalies, but this comes with its own challenges. There is still a danger that artificial anomalies may lead to promoting a sub-optimal non-conformity measure for real world data. The artificial anomalies need to be truly representative of real world anomalies.

Instead of needing to use artificial anomalies, we will later use an approach that uses the size of a set or of a region, as a measure that does not require labelled anomalous examples.

One of main flaws with the previous binary and efficiency performance methods is their dependency on having anomalies in the testing data to evaluate an algorithm. As discussed above there are several challenges that can be avoided by using a method that can measure performance without being dependent on the testing set. Another benefit to have a method that doesn't require a testing-set is that fits in line with the *semi-supervised anomaly detection* [4] and *unsupervised anomaly detection* [4] settings in which we have no knowledge of the anomalies.

2.3 New performance measures

In this section we propose and discuss new performance measures that do not suffer from some of the pitfalls mentioned in the previous section. As discussed in the previous section prior efficiency measures for conformal predictors require the use of labelled data. Binary classification performance measures (AUC and pAUC) require the use of labelled data. We discussed in the previous section that there are several challenges that arise with using labelled data. In this section we propose a method that does not require labelled anomalies to avoid these issues. This also makes the method applicable to semi-supervised anomaly detection [4]. In semi-supervised anomaly detection only the labels of the training data are known.

2.3.1 Average P-value (APV)

We propose an ϵ-independent efficiency measure called *average p-value* (APV). APV is the average p-value of the *normal* label across the feature space. The only inputs to APV are the training set consisting of only *normal* objects and the non-conformity measure it is evaluating. In principle APV can be applied to other conformal prediction problems and in general every label will have its own APV. APV uses the unsmoothed conformal predictor which outputs p-values for a new object in the range of 1 and $\frac{1}{n+1}$ where n is the size of the training set. This ensures that the outputted p-values are as optimistic as possible for any given point.

APV is closely related to the S-Criterion but instead of using the testing set it uses the feature space. The aim of APV in the feature space is to tightly wrap the prediction region around regions/objects belonging to the *normal* label. Tightly wrapping maximises the amount of the feature space that will be predicted as anomalous, thus increasing our chances of detecting all anomalies. Predicting the entire feature space as anomalous is clearly an incorrect approach on its own as then no objects would be predicted as belonging to the *normal* Label. However the validity property of conformal predictors

ensures that the ratio of *normal* objects predicted as *normal* converges to $(1-\epsilon)\%$. So even if the efficiency measure aims to predict as much of the feature space as belonging to the *anomaly* label as possible the validity property prevents the problem of predicting none of the feature space as *normal*.

An object is predicted as *normal* if its p-value is greater than or equal to *epsilon*. For points in the feature space to be predicted as anomalous there p-values need to be small. As such the smaller the APV the more efficient the conformal predictor.

APV can be formulated two ways as the mean of all p-values across the feature space or as integral across the feature space.

$$APV = \frac{1}{|X|} \sum_{x \in X} pCalculate(x) \tag{2.7}$$

$$APV = \frac{1}{|X|} \int_{X} pCalculate(x)dx \tag{2.8}$$

Where $pCalculate(x)$ calculates a p-value for the point x from the feature space using the training set.

2.3.1.1 Approximate average p-value

Sampling every point in the feature space to calculate the APV is too computationally expensive. In this section we present and discuss various aspects of approximating average p-value. An approximation of APV can be calculated by using a finite uniformly spaced grid of points. For every point in the grid a p-value is calculated using the training set and the point. Once the p-value of the object is calculated the object is discarded and is not added to the training set.

A grid of Λ cells is generated where g is the grid saturation, and d is the number of dimensions of the feature space. So $\Lambda = g^d$. A p-value is generated for each cell using the center point of each cell as the object to be evaluated.

Here we briefly define some notation: p_i is the p-value of object x_i from the grid. Only the p-values of objects from the grid are used, not the p-values of objects in the training

set.

$$APV = \frac{1}{\Lambda} \sum_{j=1}^{\Lambda} p_j \tag{2.9}$$

Approximating the average p-value requires defining an appropriate d-dimensional grid to sample p-values from. Previously suggested in our paper [41] is to use the minimum and maximum points $point_{min}$ and $point_{max}$ respectively from the training set as the corners of the grid. The minimum and maximum points are the minimum and maximum values for each dimension from objects in the training set.

Defining the grid:

Figure 2.3: 3x3 grid demonstrating the two different techniques of defining the grid.

Fig 2.3 demonstrates two different methods for selecting the points in the grid. On the right the centres of cells are used while on the left the cell vertices are used points on the grid. When using the vertices $point_{min}$ and $point_{max}$ are the most extreme bottom left and top right points of the grid. In this thesis we use the grid vertices method but in practice using either leads to similar results.

2.3.1.2 Optimal choice of bounds for APV

In this section we discuss various bound choices to use when calculating APV. Our goal here is to find the most practical and optimal choice of bounds. The bounds reflect the area that APV represents. These bounds affect the usefulness of APV and therefore must be chosen appropriately. Firstly we establish the properties of bounds and secondly we introduce various choices of bound.

When comparing non-conformity measures using APV it is critical that the same grid is used and thus the same bounds. This is because if the areas are not the same the APV

values are not comparable, as they will be the average p-value of two different areas in the feature space.

Ideally the bounds should contain all the areas that have p-values larger than the smallest p-value $\frac{1}{n+1}$. This is because these areas have the potential to affect the ranking of two non-conformity measures when using APV. Adding the areas with the minimum p-value $\frac{1}{n+1}$ will only cause the APV to tend towards $\frac{1}{n+1}$ if they are added. Here we prove that adding additional points that have a p-value larger than $\frac{1}{n+1}$ will only decrease in score. We formulate APV in the following way where Λ is the number of points in the grid testing set, APV_Λ is the average p-value for Λ and P_Λ is the p-value for our new testing point. This is for a fixed non-conformity measure and training set and the spacing between points does not change.

$$APV_\Lambda = \frac{((\Lambda - 1) \times APV_{\Lambda-1}) + P_\Lambda}{\Lambda} \tag{2.10}$$

In the case where we only add points with the minimum p-value $\frac{1}{n+1}$, we want to show that the APV monotonically decreases towards $\frac{1}{n+1}$ as more points are evaluated. This leads to a slight reformulation where we substitute in the minimum p-value.

$$APV_{m+1} = \frac{(m \times APV_m) + \frac{1}{n+1}}{m+1} \tag{2.11}$$

Firstly we prove by induction that if $APV_{m+1} > \frac{1}{n+1}$ then $APV_m > \frac{1}{n+1}$

$$
\begin{aligned}
APV_{m+1} &> \frac{1}{n+1} \\
\frac{(m \times APV_m) + \frac{1}{n+1}}{m+1} &> \frac{1}{n+1} \\
(m \times APV_m) + \frac{1}{n+1} &> X\frac{m+1}{n+1} \\
(m \times APV_m) &> \frac{m+1}{n+1} - \frac{1}{n+1} \\
(m \times APV_m) &> \frac{m}{n+1} \\
APV_m &> \frac{1}{n+1}
\end{aligned} \tag{2.12}
$$

Next we show that APV_m only decreases as more points are added ($APV_m > APV_{m+1}$) assuming the additional p-value is the minimum p-value.

$$
\begin{aligned}
APV_m &> APV_{m+1} \\
APV_m &> \frac{(m \times APV_m) + \frac{1}{n+1}}{m+1} \\
(m+1)APV_m &> (m \times APV_m) + \frac{1}{n+1} \\
APV_m + \frac{APV_m}{m} &> APV_m + \frac{\frac{1}{m}}{n+1} \\
\frac{APV_m}{m} &> \frac{\frac{1}{m}}{n+1} \\
APV_m &> \frac{1}{n+1}
\end{aligned} \tag{2.13}
$$

As long as APV_m is larger than $\frac{1}{n+1}$ we know that as we are add more points the value decreases and will be at least $\frac{1}{n+1}$. As p-values are outputted between 1 and $\frac{1}{n+1}$ we know that any P_Λ must be $\frac{1}{n+1}$ and therefore the above holds.

Now as any APV will only decrease once all the points containing a p-value larger than $\frac{1}{n+1}$ we know that the ordering of two non-conformity measures will be the same if the bounds contains all points that have a p-value larger than $\frac{1}{n+1}$.

Note not all non-conformity measures result in the prediction areas being finite with the p-values decreasing towards the minimum p-value as distance from the training set increases. These non-conformity measures are known as unbounded. An example of a non-conformity measure that is unbounded is one that outputs a fixed value for all objects and thus predicts the entire feature space with a p-value of 1 for every object. This is inefficient and not a useful non-conformity measure.

The p-values of bounded non-conformity measures decrease the further a point is away from the training set and eventually drop to the minimum p-value. In principle we are only interested with bounded non-conformity measures because unbounded non-conformity measures are not useful for anomaly detection as we are concerned with predicting as much of the feature space as anomalous as possible. Regardless unbounded non-conformity measures will have considerably higher APV values than bounded non-conformity measures.

As our approximate of APV is dependent on the choice of grid; The order of the approximate values is more significant than the values. The reasoning is that as the grid is expanded beyond the bounded region the order remains the same while the obsolete values tend towards $\frac{1}{n+1}$.

A trivial example of a bounded non-conformity measure is the k-nearest neighbours non conformity measure (k-NN NCM). In the K-NN NCM the non-conformity score is the distance to the kth nearest neighbour. The max value of k is the size of the training set and the largest possible distance is the distance between the two points furthest away from each other. Therefore if an example is more than this largest distance away from the training set its p-value will be the minimum p-value and thus k-NN NCM is bounded.

Intuitively the smallest bound should contain all the points in the training set. As these should all have a p-value larger that the smallest possible p-value. This is because

if the test point from the grid is at the same position as one of the training set, they will both have the same non-conformity being outputted and thus they will both have a p-value of at least $\frac{2}{n+1}$ which is greater than the minimum p-value.

To define the bound we will define it as the minimum bounding box that contains the set of points *region*. We also assume for this section 2.3.1.2 the size of the grid is infinite $g = \infty$ as we are addressing the choice of bounds.

Choices for determining the bounds:

- Broad Bound. Broad bound uses the entire feature space X (infinite). Whilst impractical to calculate it is useful in discussion. It is defined as:

$$region = \{x | x \in X\} \tag{2.14}$$

- Tight Bound. The tight bound is practical to obtain, because it requires nothing more than the training set. It is simply the training set T. This ensures that all known objects in the dataset are contained inside the bounds.

$$region = T \tag{2.15}$$

- Valid Bound. In the case where the probability distribution Q of the *normal* label is known. Valid bound contains all objects that have a probability greater than zero of appearing from the distribution .

$$region = \{x | Q(x) > 0 | x \in X\} \tag{2.16}$$

  In the case that the training set is infinitely long, the training set T contains all objects that have a probability of appearing from the *normal* label distribution Q. Therefore in the case where the training set is infinitely long tight bound is the same as the valid bound. The tight bound will converge towards the valid bound as the training set grows.

- Optimal Bound. The un-smoothed conformal predictor has a minimum p-value of $\frac{1}{n+1}$. The optimal bound contains all the points with a p-value larger than the

minimum p-value. p_x is the p-value for object x.

$$region = \{x | p_x > \frac{1}{n+1} | x \in X\} \tag{2.17}$$

If the grid contains the optimal bound of two non-conformity measures the APV of both will always be in the same order. This is because as shown earlier that APV converges to $\frac{1}{n+1}$ as the grid is made larger that the optimal bound.

In the case of an unbounded non-conformity measure this leads to the same bound as the broad bound.

It may possible to find the optimal bound for a given non-conformity measure but in practice this is computationally expensive. Secondly if we wish to compare non-conformity measures we must then calculate the optimal bound so that it fits all the non-conformity measures we wish to compare. If the bounds used when generating APV are not the same the APV values will not provide a reliable comparison. This leads to an unnecessary problem that can be avoided by having the bounds dependent on the training set rather than the non-conformity measure.

- Excess Bound. The excess bound is designed as a practical alternative to the optimal bound and tight bound. This bound is only dependent on the training set like tight bound unlike optimal bound.

  There will be objects from the training set that are on the edge of the tight bound. Typically points near objects from the training have p-values larger than the minimum p-value. Therefore there will likely be points outside the tight bound that are close to objects from the training set and as such they will have a p-value larger than the minimum value. Tight bound is unable to account for the p-values of these points. As such a larger bound that encompasses these points should be used.

  As such we propose a bound than is larger than tight bound. We use the principle that we are interested in non-conformity measures where the non-conformity measure will not output a p-value larger the minimum p-value beyond a particular distance

from the training set. We suggest this distance should be at least the maximum distance between any two points in the training set.

A computationally efficient method is to take the two furthest points of the bound and increase their distance from the center by the distance between the two furthest points. This ensures that the bound contains all the area that is within our suggested suitable distance from the training set.

In the following T_d is the training data for only dimension d of the feature space and d is the number of dimensions. In this case $dist$ is just the euclidean distance between the two points.

$$
\begin{aligned}
point_{min} &= \{\min T_1, \min T_2, ..., \min T_d\} \\
point_{max} &= \{\max T_1, \max T_2, ..., \max T_d\} \\
point_{distance} &= dist(point_{min}, point_{max}) \\
point_{max} &= point_{max} + point_{distance} \\
point_{min} &= point_{min} - point_{distance} \\
region &= \{point_{min}, point_{max}\}
\end{aligned} \tag{2.18}
$$

Fig 2.4 and 2.5 provide some visual examples of the various bound choices. This is an artificial example that uses a training set compromised of data from 5 randomly placed Gaussian distributions. This is because the *normal* label typically consists of several behaviours or clusters. These distributions are bounded so that they do not output values beyond what is marked by the valid bound (green) which uses the extreme values from this distribution. Fig 2.4 shows a bounded non-conformity measure and fig 2.5 shows a non-bounded non-conformity measure.

In the rest of this section we compare the different bound choices and discuss their relationship.

Under certain circumstances it is possible to prove that optimal, tight and valid excess

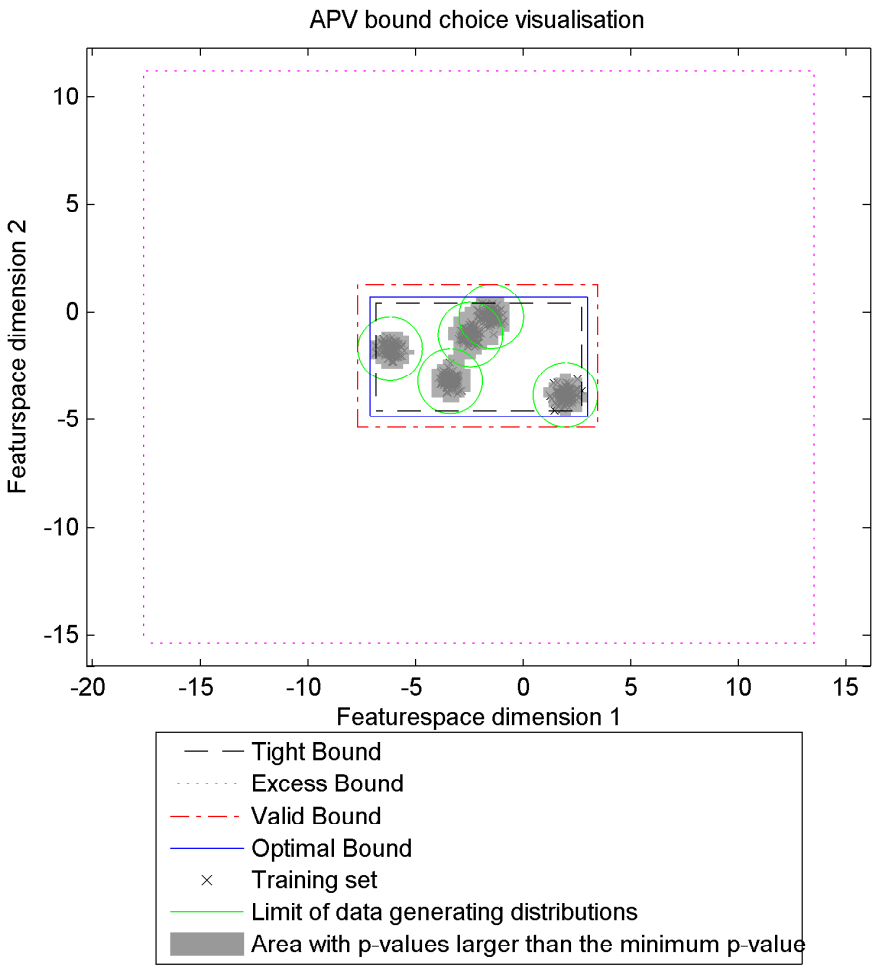

Figure 2.4: Bounds Diagram - Artifical 2D data showing the various bounds for the 1-Nearest Neighbours non-conformity measure.

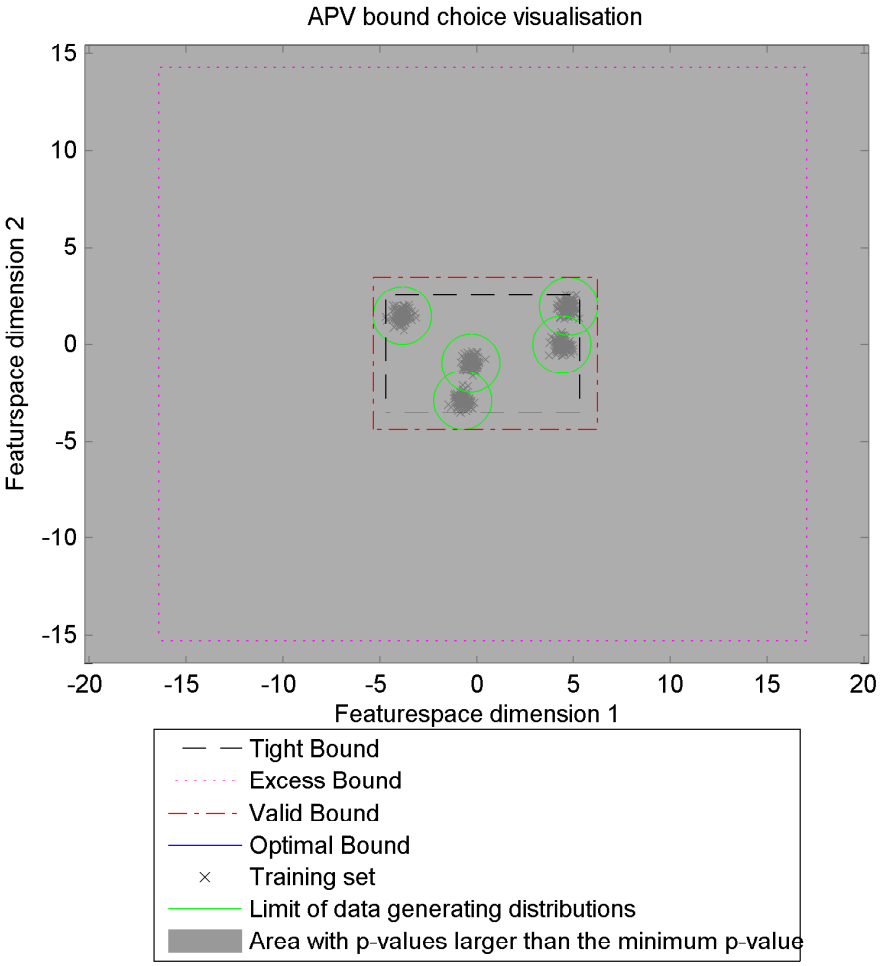

Figure 2.5: Bounds Diagram - Artifical 2D data showing the various bounds for the non-conformity measure that ouputs 1 for any input.

bounds all lead to the same bound. However the ordering of the APV values will stay the same across all of the bound choices under ideal conditions.

The valid and optimal bound cases are identical if the idealized conformity measurer is used. This is because the idealized conformal measure outputs the probability of the object occurring. Any area that doesn't have a probability of appearing will have the minimum p-value and thus valid and optimal bound are the same if the idealized conformal measure is used.

As the training set continues to expand tight bound converges towards the valid bound. Eventually every possible instance from the distribution will be in the training set. Therefore tight bound and valid bound are identical in the case of an infinitely long training set.

Here we summarize the bound options. Valid bound requires knowledge of the distribution from which the data is created. Optimal Bound is dependent on the non-conformity measure as the areas that are predicted with the largest and/or smallest p-values change between non-conformity measures. Broad bound is too big to calculate for infinitely sized feature spaces. Tight bound is usually too small to capture useful areas. Excess bound offers a compromise between tight bound and broad bound. Excess bound is the most practical as it only requires knowledge of the training set and avoids some pitfalls of the tight bound.

In the case of infinite number of points optimal bound, broad bound and valid bound all offer optimal bounds. However broad bound is unnecessarily large and for practical applications the grid points will be the furthest apart so it is unlikely to be useful for any practical calculation of APV unless for a finite space.

2.3.1.3 Optimal choice of grid saturation

Recall that the grid saturation g is the number of points per dimension to be sampled. Ultimately the choice of g has the greatest effect on the computational cost of calculating APV, as there are g^d points to calculate p-values for.

The larger the grid saturation g the more accurate the approximation, as this leads to more of the bounded area being accurately represented.

There is a danger that if g is too low, finer p-value differences in the box will be missed. A potential heuristic to avoid this is to ensure that a cell of the grid is the same length as the smallest distance between two objects in the training set.

2.3.1.4 Theoretical optimal conformity measures of APV

In the recent work of Vovk et al [46] they explore the usefulness of efficiency measures. In particular they prove that several efficiency measures are optimal. They are optimal in the sense that the idealized conformity measure is ranked best using the measures. The conditional probability idealized conformity measure outputs the true probability of an object appearing from the data-generating distribution.This is the best possible conformity measure. It is proven that under certain conditions the idealized conformity measure is an optimal choice of conformity measure under the S and N efficiency criteria.

APV is similar to the S-criterion but differs in that it is computed across a grid where as S-Criterion is computed across a testing set.

In Vovk's work the conditions used to prove this optimality require an infinitely long training and testing set. In this case S-Criterion and APV are equivalent. Under this circumstance APV regardless of choice bounds it is the same as using the broad bound which encompasses all of the feature space in the testing set and all the normal examples are in the training set. As the S-Criterion and APV are equivalent under these conditions then the idealized conformal predictor is also optimal for the APV efficiency measure.

2.3.2 Average logarithmic p-value (ALPV)

APV measures performance across all significance levels evenly. However in practical anomaly detection problems minimizing the number of false positives is key to good performance. Thus performance at lower ϵ levels is more important. This is motivated by the same idea of using partial AUC over full AUC as suggested by Laxhammar [19].

ALPV is a modification of APV. Instead of using the sum of the p-values from all points it uses the mean log p-value for all objects in the grid. By using the logarithmic p-value more weight is given to smaller p-values. As $log(0)$ is undefined the unsmoothed conformal predictor must be used giving a minimum p-value of $\frac{1}{n+1}$ where n is the number of objects in the training set.

$$ALPV = \frac{1}{g^d} \sum_{i=1}^{n} \log p_i \tag{2.19}$$

2.4 Summary

In this chapter we have introduced and discussed the suitability of pre-existing performance measures for both conformal prediction efficiency and binary classification. We discussed the application of binary classification measures and efficiency measures to anomaly detection. We highlighted that various efficiency criteria require a certain number of labels, and that some are dependent on the significance parameter ϵ. The challenge in some anomaly detection applications of a lack of anomaly examples was also discussed.

We proposed APV and ALPV to address the poor availability of labelled anomalies in the anomaly detection domain. APV and ALPV both work effectively in the single-label problem. APV and ALPV are used to measure the performance of non-conformity measures based on the amount of the feature space predicted as *normal*. This is so that we can find non-conformity measures that predict as much of the feature space as anomalous as possible. It desirable to predict as much of the feature space is possible to pick up as many *anomaly* objects as possible. The validity property of conformal predictors ensures that the ratio of correctly classified *normal* objects converges to $(1-\epsilon)$. The key novelty is there now exists an approach suitable for measuring performance of conformal predictors for anomaly detection without the need for labelled examples of anomalies. APV and ALPV only require a testing set to assess a non-conformity measure.

Chapter 3

Application of efficiency measures

In this chapter we empirically study measures of efficiency for anomaly detection as introduced in the previous chapter 2. Understanding the theoretical concepts and properties of performance measures is useful but the assumptions and ideas behind them may not hold up with real world data. In this chapter we seek to demonstrate and evaluate the methods introduced in the previous chapter. These will involve experiments with real world data.

We evaluate APV 2.3.1, ALPV 2.3.2 alongside other performance metrics for various datasets. We later discuss the relation between them and discuss in what circumstances each should be used.

This chapter introduces and extends the work in our publication [41].

3.1 Experiments

3.1.1 Low dimensional AIS Dataset

As discussed in the last chapter the computational complexity of APV grows exponentially with the dimensionality of the dataset. We aim to apply APV to a real world problem such as detecting anomalous trajectories. To accomplish this we apply APV to AIS data (section 1.1.1.1), however the dimensionality of such data is high. Typically a trajectory is made up of a sequence of points. These points typically include position, a velocity vector and the time they are at that position. Every trajectory can also have a different length.

To overcome these problems we apply a dimensionality reduction technique called t-SNE to create a lower dimensional dataset. This also has the benefit of lowering the computational cost of computing the non-conformity scores. The disadvantages of lowering the dimensionality is that extra computation must be done to reduce the dimensionality of the data. There is also potential for the dimensionality reduction technique to aid in separating the anomalies itself.

Throughout the experiments we use leave-one-out cross-validation with *supervised anomaly detection* which has labelled anomalies and normal objects (from a *testing set*) where the correctness of the output can be checked.

3.1.1.1 Dimensionality reduction

The dimensionality reduction is achieved by applying a package called T-SNE. The t-Distributed Stochastic Neighbour Embedding (T-SNE) algorithm [44] is a non-deterministic and effective dimensionality reduction algorithm. It has been primarily used for visualisation but we use it to transform our data to lower-dimensional space to evaluate non-conformity measures.

In this particular application of T-SNE to trajectory data we replaced the Euclidean pairwise distance matrix with the Hausdorff distance matrix [21], but otherwise we use the

standard MATLAB implementation[1]. Hausdorff distance was shown to be a good discriminator against anomalies [21] and was previously successfully used in the k-NN algorithm for trajectory data. The directional Hausdorff distance $\overrightarrow{H}(F,G)$ is the distance from set F to set G. The symmetrical Hausdorff distance is denoted by $H(F,G)$. Hausdorff distance uses a distance metric *dist* between the sets of points:

$$\overrightarrow{H}(F,G) = \max_{a \in F} \left\{ \min_{b \in G} \{\text{dist}(a,b)\} \right\}$$
$$H(F,G) = \max \left\{ \overrightarrow{H}(F,G), \overrightarrow{H}(G,F) \right\}$$

3.1.1.2 Non-conformity measures

In this experiment we consider two Non-Conformity Measures (NCM): the first is based on Kernel Density Estimation (KDE) and another, for comparison, on the k-Nearest Neighbours algorithm (kNN). Lei et al. [24] considered KDE as a conformity measure in the unsupervised setting.

We use these as the theoretically optimal non-conformity measure is density based and both KDE and kNN approximate density. The idealized conformity measure is introduced in [46], in the case of anomaly detection (single class) this can be thought of as $Q(x_i)$ where Q is the probability distribution of the *normal* label. The density of the probability distribution for a particular object x_i is $Q(x_i)$. Therefore the idealized non-conformity measure is $-Q(x_i)$.

We start by introducing the Kernel Density Estimation (KDE) measure. It allows assessing non-conformity based on the density of data points. The normal objects are usually concentrated in relatively small areas (high density areas or clusters) while anomalies will be outside these clusters. This can be exploited by estimating a probability density function from an empirical data set. A standard method to do this is to use kernel density estimation. It is a non-parametric technique that requires no knowledge of the underlying distribution.

[1] http://lvdmaaten.github.io/tsne/

We can interpret a density function as a measure of conformity – many similar type of data points will be located together; hence we can multiply it by minus one to convert it to a non-conformity measure for consistency as we have introduced the theory using non-conformity measures rather than conformity measures.

Input : Object z_i, Set of objects $z_1, z_2, ...z_n$ (note in this setup z_i is included in the set), bandwidth h, Kernel function K, number of dimensions d

Output: Non-conformity score A

$A_i = -\left(\frac{1}{nh^d}\sum_{j=1}^{n} K\left(\frac{z_i - z_j}{h}\right)\right)$

Kernel density estimators use the previous objects with a bandwidth parameter h that specifies the width of each object.

We will treat the bandwidth uniformly in each dimension, and fixed for each object. A kernel K is a symmetric function centred around each data point. In this thesis we use a Gaussian Kernel function for KDE: The Gaussian kernel is defined as follows:

$$K(u) = (2\pi)^{-d/2} e^{-\frac{1}{2}u^T u}$$

Lei et al. [24] have carried out work extending conformal prediction to produce minimal prediction regions with the use of kernel density estimators (KDE) and initially proposed KDE as a conformity measure in the unsupervised setting. Their method is underpinned by utilizing a custom bandwidth estimator that minimises the Lebesgue measure of the prediction set in the space.

We have not applied any bandwidth estimators in this experiment because we wish to compare KDE with another method that also has a parameter and test performance for the parameters against multiple performance criterion.

We also apply k-Nearest Neighbour (kNN) NCM [20]: d_{ij}^{+} is the jth nearest distance to an object z_i from other objects.

Input : Object z_i, Set of objects $z_1, z_2, ...z_n$, number of nearest neighbours k

Output: Non-Conformity score A

$A_i = \sum_{j=1}^{k} d_{ij}^{+}$

The nearest neighbour non-conformity measure was found to be useful in detecting anomalies [20] and we shall use it to compare performance with the KDE NCM.

3.1.1.3 Data

An object in our task is a trajectory that can be represented as a function of position over time. We convert the trajectories into a sequence of discrete $4D$ points (x, y, x_{speed}, y_{speed}) in a similar method to [21].

The original broadcasts are interpolated at a sampling distance of 200m.

If a vessel leaves the observation area for a period of 10 minutes or more, or if the vessel is stationary for a period of 5 minutes or more we consider this as the end of a trajectory. Therefore a *trajectory* is a sequence of $4D$ Points and can have any length. The $4D$ points are normalised so that $x, y \in [0, 1]$ and $x_{\text{speed}}, y_{\text{speed}} \in [-1, 1]$.

The Portsmouth dataset we evaluate was collected from a single AIS receiver on the south coast of England, during July of 2012 for one week. We filtered the data such that it only contains AIS broadcasts that report their location in a specific area between the Isle of Wight and Portsmouth. This was done to ensure better data reliability as the further an AIS broadcast travels the more likely it is to not be received.

In this dataset we consider only passenger, tanker and cargo vessels to reflect a degree of 'regular' behaviour (e.g. going from A to B and back). We assume that this data does not contain anomalous behaviour. We will start with applying traditional performance metrics (see 2.1.1), therefore we have to add some artificial anomalies to the data, there are two sources of them. The first contains 22 search & rescue helicopter trajectories. The other source is 180 'artificial' anomalies: random walks that have been generated starting from a random position of a random observed normal vessel. They follow a random direction and speed and a new point is generated every 200m as it has been suggested in [20]. However,

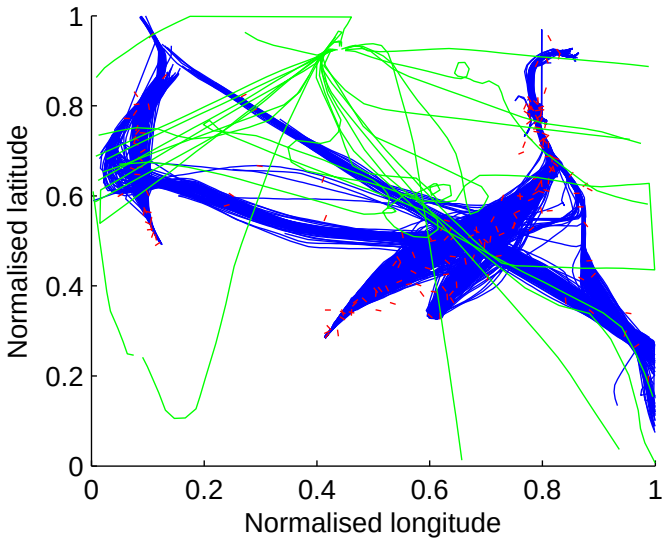

Figure 3.1: Blue shows normal trajectories. Red shows the last 200m of the artificial anomalous trajectories. The green trajectories are the helicopters.

.

unlike in [21] we only consider the entire trajectory and do not calculate detection delay.

Instead of generating anomalous trajectories of 3km in length we are using different length of 'artificial' anomalies. The composition of our 180 'artificial' trajectories is the following: 150 of length 200m, 20 of length 400m, 10 of length 600m, 10 of length 800m and 10 that are 1000m long. The aim is to diversify the difficulty by providing both easy and difficult anomalies to detect.

The dataset consists of 1124 normal trajectories with 202 anomalies added to it. All these trajectories can be seen in Fig 3.1.

Prior to applying conformal prediction we run the T-SNE algorithm to produce $2D$ representations of the trajectories.

3.1.1.4 Results

For measuring the performance of the non-conformity measures we use AUC as introduced in section 2.1.1. The partial AUC (pAUC) is also used to show performance for $fpr \in [0, 0.01]$, note that pAUC is normalised to be in the range $[0, 1]$. The average p-value (APV) introduced in section 2.3.1 is calculated, recall the lower the APV the more efficient the classifier. The average logarithmic p-value (ALPV) as introduced in section 2.3.2 is also calculated.

AUC and pAUC are our criteria for anomaly detection ability in the supervised setting and the average p-value in the unsupervised setting which doubles as a measure of efficiency. We compare both non-conformity measures for the best parameter values of AUC, pAUC, APV and ALPV. The APV, ALPV, AUC and pAUC for various parameter values of both NCMs can be found in the Table 3.1.

Table 3.2 was created to expand upon the k neighbours parameter as it is apparent that the highest AUC k-NN classifier was not in the initial parameter set. A rather important thing to note with testing leave-one-out is that anomalies are part of the training set, in practical applications ideally the training set would not contain anomalies. This is because objects in the training set are treated as belonging to the 'normal' label. Adding anomalies to the training set could worsen performance as anomalies do not originate from the distribution of 'normal' objects. From the tables for all the parameters the highest AUC (supervised setting) are for KDE $h = 3$ 0.7830 and kNN $k = 80$ 0.7616, it is clear that KDE has the higher AUC over k-NN, and is therefore better at detecting anomalies across all ϵ in the leave-one-out setting. For both these parameters k-NN ($k = 80$) also has a larger APV 0.0638 against KDE ($h = 3$) 0.0606 which indicates that KDE is more efficient and offers better performance than k-NN when AUC is the criterion.

When we consider the most efficient APV (unsupervised setting) as a criterion k-NN's best parameter is $k = 7$ with APV of 0.0453, whilst KDE's smallest APV is 0.0441 for $h = 1$. The most efficient parameters using the ALPV criterion are different from that of

k (k-NN) or h (KDE)	1	2	3	4	5	6	7	8	9	10
KDE AUC	0.6116	0.7620	**0.7830**	0.7455	0.6727	0.5932	0.5086	0.4406	0.3811	0.3518
k-NN AUC	0.2977	0.3051	0.3407	0.3611	0.3894	0.4066	0.4193	0.4323	0.4466	0.4618
KDE APV	**0.0441**	0.0519	0.0606	0.0694	0.0801	0.0936	0.1103	0.1307	0.1575	0.1941
k-NN APV	0.0490	0.0481	0.0469	0.0461	0.0458	0.0454	**0.0453**	0.0453	0.0455	0.0456
KDE pAUC	0.0082	**0.0484**	0.0285	0.0235	0.0270	0.0297	0.0415	0.0342	0.0250	0.0000
k-NN pAUC	0.0001	0.0099	0.0099	0.0099	0.0150	0.0276	0.0340	0.0381	0.0427	**0.0484**
KDE ALPV	-1.7795	-2.0408	**-2.2053**	-2.1881	-2.1794	-1.9511	-1.5162	-1.3073	-1.3595	-1.4082
k-NN ALPV	-1.6024	-1.6678	-1.6856	-1.6875	-1.7474	-1.7983	-1.7984	-1.9122	-1.9389	-1.9751

Table 3.1: AUC, APV, pAUC and ALPV for various parameters of k-NN and KDE NCMs

k (k-NN)	20	30	40	50	60	70	80	90	100
k-NN AUC	0.6157	0.7051	0.7257	0.7403	0.7519	0.7547	**0.7616**	0.6832	0.6301
k-NN APV	0.0486	0.0519	0.0549	0.0574	0.0595	0.0615	0.0638	0.0705	0.0827
k-NN pAUC	0.0304	0.0253	0.0327	0.0308	0.0400	0.0381	0.0384	0.0434	0.0375
k-NN ALPV	**-2.502**	-2.4585	-2.4243	-2.3225	-2.2919	-2.315	-2.2672	-2.2055	-1.8502

Table 3.2: Extension of k-NN results

the APV criterion this is due to ALPV being more sensitive to smaller p-values. The best parameters for ALPV are $k = 20$ for the k-NN non-conformity measure and $h = 3$ for the KDE non-conformity measure.

The optimal result for the supervised problem requires more neighbours ($k = 80$) than the unsupervised one ($k = 7$) because most of the anomalies are close to each other (concentrating in a small area on figure 3.2) which makes this problem harder. At the same time their influence on the unsupervised prediction is relatively small.

The pAUC Criterion in our leave-one-out setting may not be appropriate as the number of anomalies is far greater than a 1% composition of the dataset, but it is still a vital criterion for the purpose of minimising the false positive rate. KDE's best parameter by pAUC is $h = 2$ with a pAUC 0.484 and k-NN's best pAUC is with $k = 10$ with 0.484, however with these parameters $k = 10$ has a smaller APV and is thus more efficient. k-NN also achieves higher pAUC for more parameter values than KDE. This is quite apparent with pAUC > 0.03 for $k = 7$ to $k = 20$, and for $k = 40$ to $k = 100$, whereas for KDE only $h = \{2, 7, 8\}$ has pAUC > 0.03.

In addition to the results table, the figs. 3.2 to 3.4 visualize the prediction regions in the feature space of the KDE non-conformity measure for various values of ϵ. The figs. 3.5 to 3.8 visualize the prediction regions in the feature space the k-NN non-conformity measure for various values of ϵ. These figures visualise the 'normal' class prediction sets in the feature space of various ϵ. They are generated using a grid of points (pixels) as the test set. The p-value of each point is calculated using all the objects from our dataset as the training set. Note the training set includes 'anomaly' objects because the leave-one-out setting is used in the experiments and the visualisations also reflect this.

It is evident from these visualisations that the APV criterion favours parameters that lead to smaller prediction sets. The visualizations of ALPV do seem to occupy larger prediction regions. This is due to how the visualizations are created. The visualisations show the prediction sets for $\epsilon \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10\}$. In the case of the ALPV the logarithmic scale heavily favours the smallest prediction set

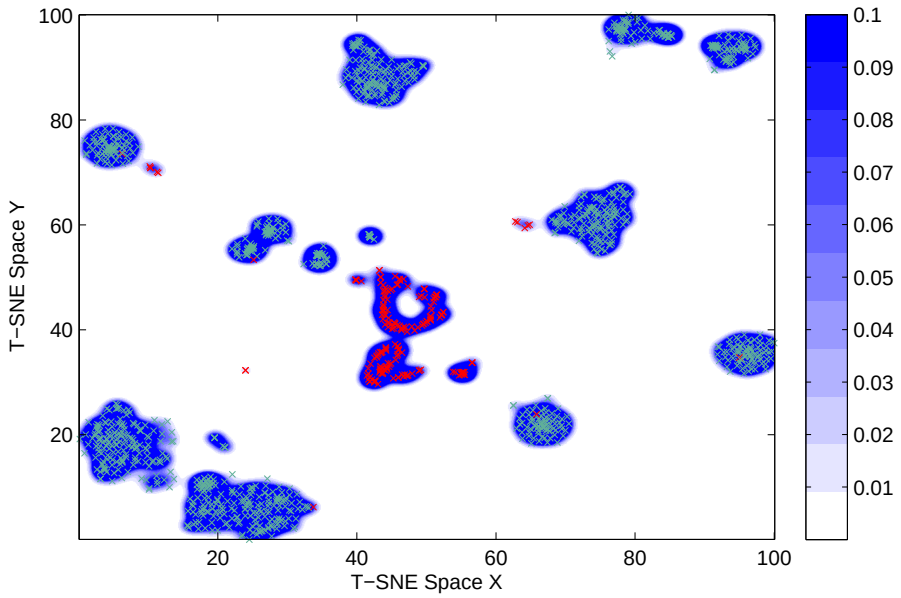

Figure 3.2: Prediction sets for various parameters of ϵ in T-SNE space for KDE NCM (h=1). This is the optimal parameter for under the APV criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

size at the smallest p-values. There are 1326 objects in the training set used for to generate the p-values for these visualisations the grid. The smallest p-value for a point is $\frac{1}{1326+1} \approx 0.0008$ which is considerably smaller than the $\epsilon = 0.01$.

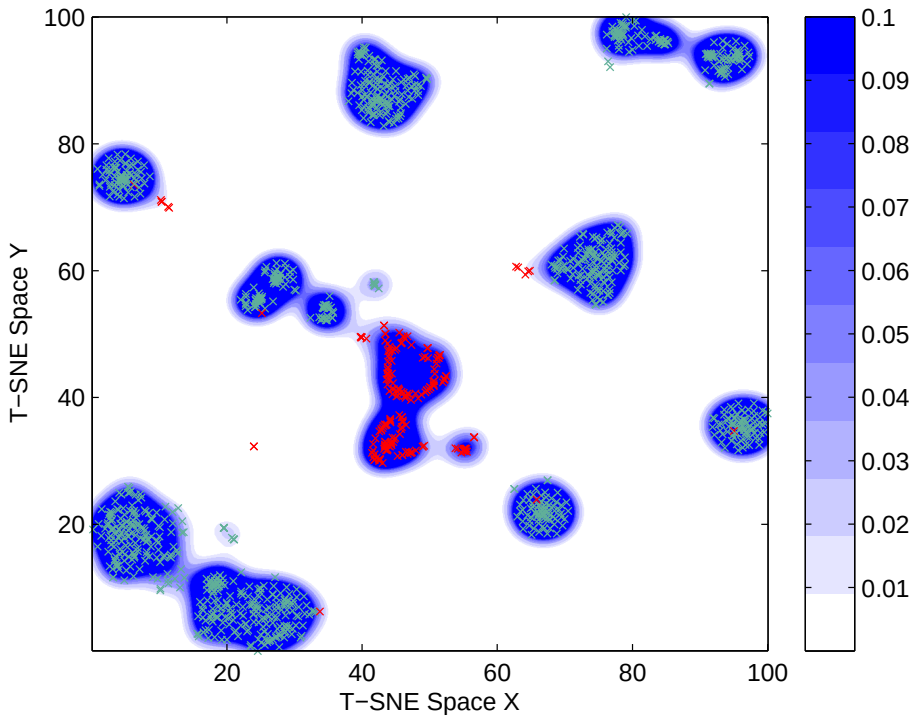

Figure 3.3: Prediction sets for various parameters of ϵ in T-SNE space for KDE NCM (h=2). This is the optimal parameter for KDE under the pAUC criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

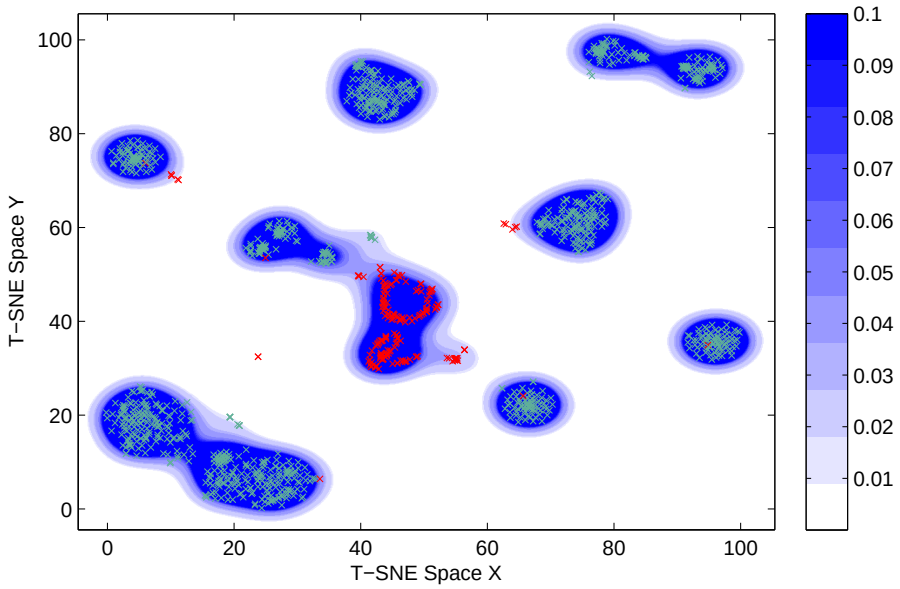

Figure 3.4: Prediction sets for various parameters of ϵ in T-SNE space for KDE NCM (h=3). This is the optimal parameter for KDE under the AUC and ALPV criteria. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

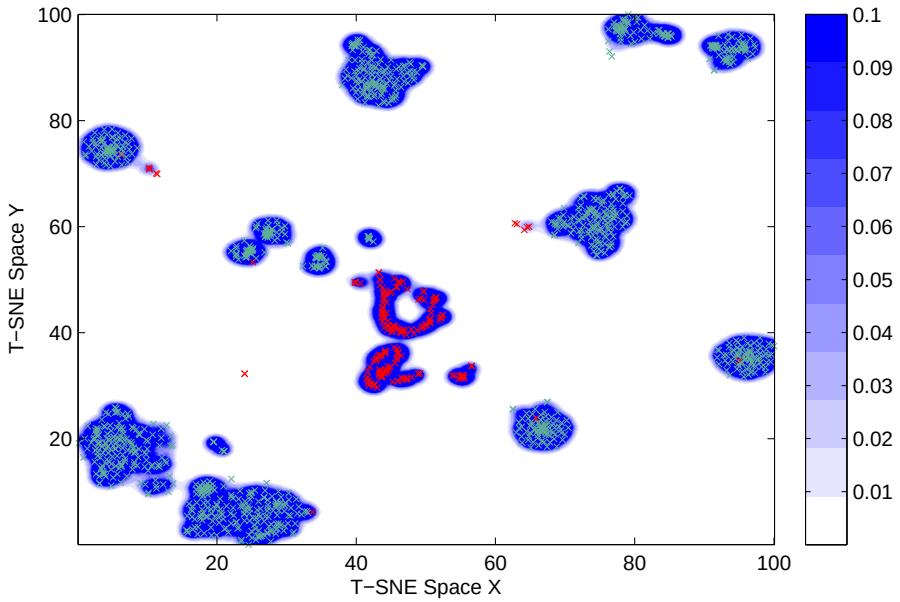

Figure 3.5: Prediction sets for various parameters of ϵ in T-SNE space for K-NN NCM (k=7). This is the optimal parameter for k-NN under the APV criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

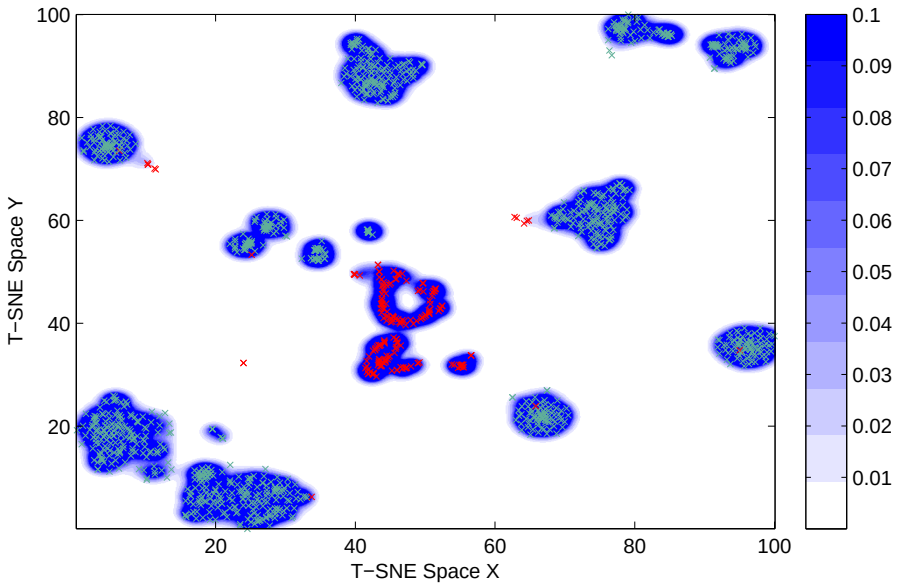

Figure 3.6: Prediction sets for various parameters of ϵ in T-SNE space for K-NN NCM (k=10). This is the optimal parameter for k-NN under the pAUC criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

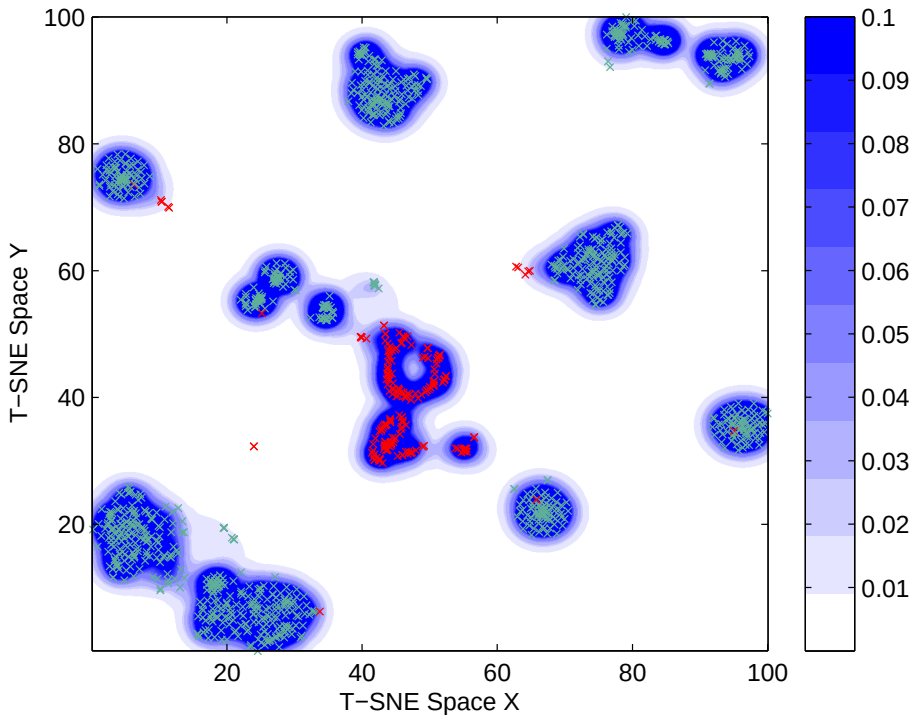

Figure 3.7: Prediction sets for various parameters of ϵ in T-SNE space for K-NN NCM (k=20). This is the optimal criterion for k-NN under the ALPV criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

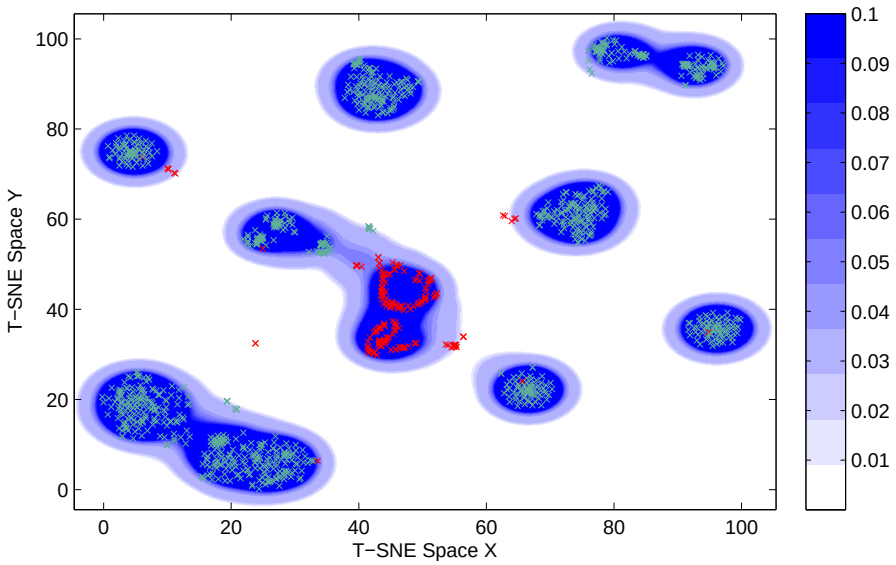

Figure 3.8: Prediction sets for various parameters of ϵ in T-SNE space for K-NN NCM (k=80). This is the optimal criterion for k-NN under the AUC criterion. The colour scale on the right represents the prediction regions for various values of ϵ. The crossed points are trajectory objects. The red points represent *anomaly* trajectories and the teal points represent *normal* trajectories.

3.2 Conclusions

In this chapter we offered a comparison of two non-conformity measures applied to an anomaly detection problem using various performance criteria. These non-conformity measures are based on the nearest neighbours (k-NN) algorithm and kernel density estimation (KDE). Both non-conformity measures consider an entire trajectory from the maritime surveillance domain. In addition, we reduced the dimensionality of our dataset to compare the different non-conformity measures.

The performance of both KDE NCM and k-NN NCM for all criteria is heavily dependent on the choice of parameter h and k respectively. We evaluated the performance for various parameter values.

In the leave-one-out supervised setting KDE NCM for our dataset in the supervised leave-one-out setting has higher AUC than the k-NN NCM. However for most anomaly detection applications performance at small false positive rates is more important. If small false positive rate (in the form of pAUC) is the primary criterion then k-NN NCM performs better than the KDE NCM.

For APV it is apparent that KDE can lead to more efficient predictions with a smaller average p-value than k-NN, this indicates KDE NCM in the unsupervised setting with a good choice of parameter performs better with our dataset than the k-NN NCM. In the experiment the ALPV criterion indicates that with a good choice of parameter the k-NN NCM performs better than the KDE NCM.

Chapter 4

Multi Class Hierarchy

Observations in real world problems are typically the result of many distributions coming together. In this chapter we seek to explore exploiting the structure of this data. For instance in the maritime surveillance domain we seek to detect anomalous behaviour in the movement of ships. The principle idea here is that individual ships are different. What is typical for one ship may not be typical for other ships. This leads to the notion of behaviour relativity. In this chapter we aim to see if we can exploit this by using a multi-class hierarchy to represent the data.

4.1 Introduction

In the past several papers have studied anomaly detection with conformal predictors [19–22, 41]. Most previous papers [19, 21, 22, 41] represent the previous data of all vessels into a single class of 'normal', which we call the global class. These predict the likelihood of a new trajectory having originated from the global class. There is is one exception in which Laxhammar [20] uses classes based on the vessel type (Passenger/Cargo/Tanker) in this case if a vessel is not predicted as belonging to one of the vessel types it will be classed as an anomaly. Laxhammar's paper provides no comparison to the global class.

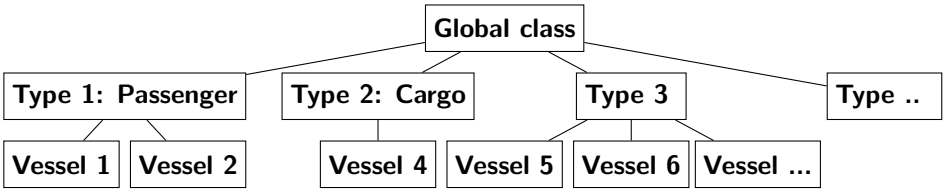

Figure 4.1: Visualisation of the multi-class hierarchy

In this chapter we wish to explore the prospect of not just using different representations of the 'normal' class but comparing them and investigate using a hierarchy of the 'normal' class. As previously stated the maritime surveillance domain is concerned with detecting anomalous trajectories. The global class leads to detecting anomalies that are peculiar compared to all vessel behaviours. There is also the idea that using the vessel's type (Passenger/Cargo/Tanker) may provide a better context for anomaly detection.

We propose putting all the trajectories into a three-level artificial hierarchy of classes as shown in Fig 4.1. As we progress down the hierarchy each level is a subset of the upper level. The **global class** contains all the previous data of the 'normal' class at the top layer, this is split as we progress down the hierarchy. At the next layer, the data is split into **type classes**, one for each vessel type. The final layer of the data is separated into **local classes**, one for each vessel.

Our approach of having multiple 'normal' classes adds more complexity to the problem, and in the rest of this chapter we investigate if in practice any benefits can be gained by using a multi-class hierarchy.

4.2 Method

Multi-Class Hierarchy

In previous applications of applying conformal anomaly detection to trajectories, typically one global class of 'normal' is used to encompass all previous data. However with vessel trajectories there exists an information hierarchy as introduced earlier in Fig 4.1.

In trajectory data some vessels such as passenger vessels will make the same repeated journeys. These repeated journeys are routes that the vessel typically takes. It can be considered 'abnormal' if they deviate from these routes. This leads to the idea of treating every vessel as its own local class as different vessels may well have different routes that they use. The immediate benefit is that it is likely that a vessel only conducts a subset of journeys from the global class and that a more focused set of previous examples could be used to save computational resources.

The vessel type classes may be beneficial as they contain more data than local classes, but will not contain all the trajectories from the global class. In our data the observed vessels come from 16 types of vessel including passenger, cargo, pleasure, tankers, dredgers, pilot vessels and many others. Each of these vessel types generally have their own limitations where they can operate and how fast these vessels can go. Not all vessel types are suitable for shallower waters.

By comparison the global class is the simplest to implement. It allows predictions to be made for trajectories belonging to vessels that have no prior data in the training set. The global class is also better suited if there is no information available on the type of the vessel. One weakness of the global class is that it is unable to distinguish between vessels of different types and will not be able to detect if a passenger vessel starts conducting journeys similar to that of a military vessel. There are also vessels that due to their nature may not follow any previous trajectories, such as search and rescue vessels conducting searches. These 'deviations' are considered as typical behaviour for such vessels and in this case the global class may perform worse.

The main advantage of producing multiple p-values like this is to attempt to better understand trajectories that are classified as 'anomalous'. The trajectory may be anomalous in the global context, but 'normal' for its local context and vice versa. We could determine under which class the trajectories are 'anomalous' or 'normal' and use this to gain insight.

In Figure 4.2 $filter$ is a function for filtering the previous trajectories for either vessels

Conformal Anomaly Detection: Multi-Class Hierarchy

Input : Non-Conformity Measure A, significance level ϵ, training objects $z_1, z_2, ..., z_{n-1}$ and new object z_n

Output: P-values: p_{global}, p_{type} and p_{local}, Boolean variable *Anomaly*

$D = \{z_1, ..., z_n\}$

for $i \leftarrow 1$ **to** n **do**
| $\alpha_{global,i} \leftarrow A(D \setminus z_i, z_i)$

end

$\tau \leftarrow U(0,1)$

$p_{global} \leftarrow \frac{|\{i{:}a_{global,i} > a_{global,n}\}| + \tau |\{i{:}a_{global,i} = a_{global,n}\}|}{n}$

$D = filter(D, type, z_n)$;

$N \leftarrow = |D|$

for $j \leftarrow 1$ **to** N **do**
| $\alpha_{type,j} \leftarrow A(D \setminus z_j, z_j)$

end

$p_{type} \leftarrow \frac{|\{j{:}a_{type,j} > a_{type,N}\}| + \tau |\{j{:}a_{type,j} = a_{type,N}\}|}{N}$

$D = filter(D, local, z_n)$;

$N \leftarrow = |D|$

for $m \leftarrow 1$ **to** N **do**
| $\alpha_{local,k} \leftarrow A(D \setminus z_m, z_m)$

end

$p_{local} \leftarrow \frac{|\{m{:}a_{local,m} > a_{local,N}\}| + \tau |\{m{:}a_{local,m} = a_{local,N}\}|}{N}$

if $min(p_{global}, p_{type}, p_{local}) < \epsilon$ **then**
| *Anomaly* $\leftarrow$ **true**

else
| *Anomaly* $\leftarrow$ **false**

end

Figure 4.2: Multi-Class Hierarchy algorithm using the minimum hybrid rule.

of the same *type* or trajectories from the same *local* vessel. In the case of *type* it will only return objects that match the type of the new object, in the case of *local* it will only return trajectories that belong to the same vessel.

As shown in Figure 4.2 we generate p-values to indicate the likelihood of the trajectory belonging to each of its three associated classes these being: p_{global}, p_{type} and p_{local}. In the p_{global} case all prior trajectories are used regardless of which vessel they come from. p_{type} is calculated from using previous trajectories from the same type of vessel. p_{local} is calculated from using previous trajectories from the same vessel. In practice compared to using a single class this requires extra processing.

As we generate 3 p-values there are several possible different methods of using these to determine whether or not to classify a trajectory as *anomalous*. In Figure 4.2 we suggest using the decision rule
$min(p_{global}, p_{type}, p_{local}) < \epsilon$ as if any of the p-values for a particular trajectory indicate that it is anomalous we predict as anomalous. This allows us to catch any possible anomaly that is detectable for a given ϵ. Aggregating the p-values in this manner does affect the property of a well-calibrated false positive rate. Instead of the decision rule being bounded by ϵ, it is bounded by $min(3\epsilon, 1)$; this is because each p-value may contribute ϵ false-positives and the maximum false-positive rate is 1.

There are also alternatives for aggregating the three p-values. We also include an experiment using the the minimum, mean and maximum value of the p-values. Another alternative is to weight each p-value using aggregating algorithms such as a weighted majority algorithm. However in this chapter we do not explore aggregating algorithms as they are a research topic in their own right.

4.3 Experiments & Data

For the experiments we use AIS data as introduced in section 1.1.1.1. We use AIS data collected from Portsmouth on the south coast of England during 2012. For all our data

we know both the vessel type and ship IDs.

This chapter focuses more on exploiting the structure of classes rather than proposing a non-conformity measure. As such it uses a pre-existing system to evaluate the performance. In this chapter we represent trajectories as a sequence of discrete $4D$ points (x, y, $x_{velocity}$, $y_{velocity}$) in a similar method to [20]. The original broadcasts are linearly interpolated. They are spaced at 200m intervals to reduce the problem of over and under-saturation of data as a single receiver may not be able to capture all broadcasts due to a variety of factors such as range, weather, time-interval, GPS error and obstacles.

The algorithm for splitting the broadcasts into trajectories is similar to [21]. If a vessel leaves the observation area for a period of 10 minutes or more, or if the vessel is stationary for a period of 5 minutes or more we consider this as the end of a trajectory. Therefore a *trajectory* is a sequence of $4D$ points which are moving and can have any length. The $4D$ points are normalised so that $x, y \in [0, 1]$ and $x_{velocity}, y_{velocity} \in [-1, 1]$.

In our experiments we consider trajectories that occurred over a several week time period. All the experiments are conducted using a batch offline mode in which the data is split into training and test sets. The testing and training sets being chosen from a randomly shuffled set of trajectories. Most of the experiments are comparing p-values, in these cases the decision rule is that a trajectory is classed as 'anomalous' if $p_{value} < \epsilon$ otherwise it is classed as 'normal'. Recall that we use 16 different types of vessel in our experiments as introduced in the method section.

We also do not investigate the possibility of 'anomalies' existing in the training set. This would lead to a different distribution being represented in the training set and a potential degradation in performance.

Anomaly generation: Random walks

One of the big challenges is that there is a lack of real-world labelled anomalous datasets for AIS trajectories - at the time of writing the author is unaware of any publicly available AIS dataset containing labelled anomalies. Therefore it is necessary for empirical purposes

to create artificial anomalies.

One previously suggested approach [21] is to select a pre-existing trajectory, pick a random point on the trajectory and simulate a random walk for a specified distance. It can be argued that randomness may not truly represent real world anomalies however it does give an indication of ability to distinguish trajectories generated from a different distribution. In these experiments random walks are generated to a distance of 600m. Once generated the random walk trajectories will report the same vessel ID as the vessel ID from the pre-existing trajectory representing observed behaviour for that vessel.

Wrong type behaviour anomalies

In our study of using type and local classes it is useful to demonstrate the property that using a global class does not distinguish if a trajectory is 'normal' for in the context of a particular vessel. Wrong type anomalies are designed to represent this problem. To generate wrong type anomalies we choose a trajectory and assign it a different random vessel ID matching another in our database. The anomalous trajectory will then be compared against the same type as the one, the new vessel ID came from. This emulates a vessel behaving in a manner possibly unexpected for itself.

4.3.1 Experiment 1: Comparing global, type and local models directly

In this experiment we seek to directly compare p_{global}, p_{type} and p_{local} when they are all given the same data. In this experiment we filter the available trajectories to only those of vessels with at least 600 trajectories leaving us with 16 vessels. We then use 200 trajectories from each vessel in the training set and 200 in the testing set. We add 100 artificial random-walk anomalies to the testing set. This leads to a testing set of 3300 trajectories and a training set of 3200 trajectories. For each trajectory we then compute each of the 3 p-values using its given type and vessel ID.

4.3.2 Experiment 2: Maintaining computational cost

In this experiment we examine what happens in the case where we have limited computational resources. This is particularly interesting to investigate in the case where there exists a large amount of historic data but a result is needed in finite time. The question this experiment seeks to answer is in the case of limited computational resources which of the models are best to use? To emulate these conditions, we limit the number of trajectories available for each of the local, type and global models to maintain the same computational cost across models. We test the performance of each model by comparing the performance of the resulting p_{local}, p_{type} and p_{global} p-values to their true labels.

Recall that a p-value is generated by calculating a non-conformity score for every object in the training set and the object that is being tested. In the multi-class hierarchy framework the training set is filtered by the vessel type and vessel ID in the case of p_{type} and p_{local}. If any filtering is carried out this requires the calculation of fewer non-conformity scores.

In this experiment we create training sets that ensure that every model utilizes the same number of non-conformity scores. This requires the creation of specific training sets for each of p_{local}, p_{type} and p_{global}.

To create the dataset we only consider vessels that have at least 1000 trajectories available in the original data. This ensures we have sufficient examples to test this with a large number of trajectories. Only 11 vessels in our data have at least 1000 trajectories in the source data. In order to balance the types properly we further limit the number of vessels to 9, as the data contains 3 vessel types, these will be each represented by 3 vessels. This prevents one type of vessel dominating the data, and ensures that each vessel type has sufficient trajectories in the data.

For each vessel, we randomly sample 500 trajectories from the original data that belongs to that vessel to form the *training source dataset*. This results in the training source dataset containing 4500 trajectories. The training source dataset is then used to create the training

sets for each model.

Local training set

The p_{local} training set is the training source dataset.

Type training set

The p_{type} training is created by randomly selecting 500 trajectories for each vessel type from the 1500 trajectories available for each vessel type in the 'training source dataset'. This results in the training set being a total size of 1500 (3 vessel types x 500 trajectories).

Global training set

The p_{global} training set is created by randomly selecting 50 trajectories for each vessel from the 'training source dataset', resulting in a global training set total size of 450.

Testing Set

The testing set is created by randomly selecting a further 500 trajectories from each of 9 vessel's original data. These are different to the 500 trajectories used to create the training source dataset. The 500 trajectories from each vessel are combined together resulting in a testing set of size 4500. These are all labelled as *normal*, under the assumption that the source data contains no anomalous trajectories.

To add some anomalies we added trajectories containing random walks as described in section x . To do this we created 10 random walks for each vessel, randomly selecting a trajectory from the testing set to serve as a starting point. These 90 random walks are then added to the testing set with the label*anomalous*. Thus the test set size is 4950.

Using this construction, each p-value calculation will rely on 501 non-conformity scores, providing a fair comparison under computational limits. Note that we calculate the non-conformity score of the object we are testing, this is the extra non-conformity score that is calculated so that the total is 501 not 500.

4.3.3 Experiment 3: Wrong behaviour type Anomalies

In this experiment we aim to test how robust the different p-values are to a vessel acting in a manner that is peculiar for itself or its type, yet similar to other vessels.

To do this we create a training set from the 13 most active vessels in our dataset using 110 trajectories from each vessel. The testing set consists of a further 110 trajectories from each vessel alongside a total of 100 random walk anomalies and a total of 100 wrong type behaviour anomalies. We then generate p-values for all trajectories in the testing set.

4.3.4 Experiment 4: Hybrid Rule

This experiment investigates what happens if we merge all three p-values ($p_{global}, p_{type}, p_{local}$) together with a decision rule that determines if a trajectory is predicted as an *anomaly*. This experiment uses the same training and testing sets from experiment 1. There are three aggregations we experiment with. We predict the the object as an *anomaly* if the following rule is true.

- Minimum - $min(p_{global}, p_{type}, p_{local}) < \epsilon$.
- Maximum - $max(p_{global}, p_{type}, p_{local}) < \epsilon$.
- Mean - $\frac{p_{global}+p_{type}+p_{local}}{3} < \epsilon$.

4.4 Results

Below are the tables of the results gathered from the experiments mentioned in the previous section. The tables show the number of true positives (tp) (i.e. anomalies captured) and the number of false positives (fp) - (i.e. 'normal' trajectories mis-classified as anomalies). A bold font has been used to denote the p-value that captures the most true anomalies for a given significance level ϵ.

In table 4.1 we see that when using all the information together p_{type} generally better captures the anomalies than the other p-values. For significances 0.03, 0.05 and 0.10 the performance offered by all them is rather similar (within 1% difference). p_{local} also outperforms p_{global} at the lower significances 0.01,0.02. This reveals that it is clear that with large amounts of training data p_{type} and p_{local} are capable of out performing p_{global},

ϵ	p_{global} tp	p_{global} fp	p_{type} tp	p_{type} fp	p_{local} tp	p_{local} fp
0.01	71%	0.8 %	**85** %	0.8 %	73 %	0.8%
0.02	86%	1.6 %	**91** %	1.9 %	88 %	1.5%
0.03	93%	3.0 %	93 %	2.7 %	**94** %	2.2%
0.05	**94**%	4.3 %	**94** %	4.2 %	**94** %	3.9%
0.10	96%	8.5 %	**97** %	9.6 %	**97** %	9.5%

Table 4.1: Results of experiment 1: Direct comparison

ϵ	p_{global} tp	p_{global} fp	p_{type} tp	p_{type} fp	p_{local} tp	p_{local} fp
0.01	49 %	0.5 %	71 %	0.9 %	**76** %	0.7%
0.02	54 %	0.8 %	80 %	2.5 %	**86** %	1.8%
0.03	60 %	1 %	89 %	3.7 %	**90** %	2.9%
0.05	84 %	4.1 %	91 %	4.9 %	**96** %	5.0%
0.10	89 %	8.9 %	94 %	10%	**97** %	10.4%

Table 4.2: Results of experiment 2: Comparison with the same computational cost

ϵ	p_{type} tp	p_{type} fp	p_{local} tp	p_{local} fp
0.01	**54** %	1.3 %	52.5 %	0.9%
0.02	**76** %	2.4 %	68.5 %	1.5%
0.03	**78** %	3.5 %	77.5 %	2.4%
0.05	**81** %	5.9 %	80 %	4.5%
0.10	85 %	10.1 %	**89** %	10.8%

Table 4.3: Results of experiment 3: Wrong type behaviour anomalies

ϵ	min tp	min fp	max tp	max fp	mean tp	mean fp
0.01	93%	1.8 %	56 %	0.2%	74%	0.4%
0.02	96%	3.6 %	79 %	0.5 %	90 %	1.1 %
0.03	99%	5.5 %	89 %	1.0 %	91 %	1.4 %
0.05	99%	8.4 %	90 %	1.5 %	93 %	3.2 %
0.10	99%	15.8%	94 %	4.7 %	98 %	6.9 %

Table 4.4: Results of experiment 4: Hybrid rule

$\frac{\epsilon}{3}$	min tp	min fp	max tp	max fp	mean tp	mean tp
$\frac{0.01}{3}$	75 %	0.4 %	22 %	0.0 %	34 %	0.1 %
$\frac{0.02}{3}$	87 %	1.3 %	35 %	0.1 %	62 %	0.3 %
$\frac{0.03}{3}$	93 %	1.8 %	56 %	0.2 %	74 %	0.4 %
$\frac{0.05}{3}$	**95** %	2.8 %	75 %	0.4 %	87 %	0.9 %
$\frac{0.10}{3}$	**99** %	5.8 %	89 %	1.0 %	91 %	1.7 %

Table 4.5: Extended results of experiment 4: Hybrid rule

and if a vessel's ID is unavailable knowing its type is enough in most cases. p_{type} performs better than p_{local} for the lower significances of 0.01 and 0.02 where arguably performance is most important. However p_{local} consistently has a lower number of false positives than all other the p-values indicating the best performance for significances 0.03, 0.05 and 0.10.

Table 4.2 shows the performance of the p-values for experiment 2, the case where we consider equal computational resources. It is clear that p_{local} outperforms p_{type}, and p_{type} outperforms p_{global} at identifying a superior number of anomalies for all ϵ, this indicates that having a more focused history of prior examples improves classification performance.

Experiment 3 shows that the type class performs well at detecting 'anomalies' of vessels demonstrating behaviour from other types .

Experiments 1-3 in most cases show that the significance parameter ϵ does provide a

well-calibrated false-positive rate in most cases, even though there is no guarantee of this in the offline mode that is used.

Experiment 4 shows that the minimum hybrid rule performs far better at detecting the random walk anomalies than any of the single p-values in experiment 1. It is important to note that ϵ doesn't calibrate the number of false-positives close to ϵ as the individuals p-values do. The hybrid rule approach can potentially add false positives from the 3 p-values possibly tripling to 3ϵ false positives under the validity property of conformal predictors in the supervised setting.

In addition, we carried out experiments using $\frac{\epsilon}{3}$ to take into account that the false positive rate of the hybrid rule is expected to be below $min(3\epsilon, 1)$ as shown in table 4.5. This allows a fairer comparison to the false positives rates seen in experiment 1. Comparing table 4.1 and the right side of table 4.4, we see that the minimum hybrid method shows the best true positive results for $\epsilon = 0.03$ and $\epsilon = 0.05$ when preserving the same false-positive rate bound and performs better than using a single p-value. It is also clear that there is an overlap of the false-positives from each p-value otherwise the number of false positives would be more in line with 3ϵ for the hybrid rule results.

Of the three hybrid rules evaluated the minimum rule seems to offer the best results as it out performs the mean and max in several cases. Take for example $\epsilon = 0.02$ from the minimum rule and $\epsilon = 0.10$ from the maximum rule it is clear that the minimum rule successfully predicts more anomalies for less false positives. Also take for example $\epsilon = 0.01$ from the minimum rule and $\epsilon = 0.05$ from the mean rule in this case the minimum rules successfully predicts as many anomalies as the mean rule but also does so with less false positives.

4.5 Conclusion

Past approaches using conformal prediction for anomaly detection typically focus on using a global class, or split the classes with little overlap. In this chapter we have proposed a

new multi-class hierarchy framework for the anomaly detection of trajectories. We have also presented a study of this approach showing that there are several benefits from using alternative classes to the global class. We generate three p-values p_{global}, p_{type} and p_{local} for new trajectories. We have discussed the pros and cons of each of the p-values.

We demonstrated that in practice using these extra p-values can lead to the detection of more anomalies for less false-positives.

Computing all p_{global}, p_{type} and p_{local} leads to a higher computational cost compared to using a single class. In the case where limited computational resources are available it is shown that p_{local} detects more anomalies than p_{type} which detects more anomalies than p_{global}.

We have also shown it is possible to combine all the p-values by taking a hybrid approach. In particular using the minimum p-value of p_{global}, p_{type} and p_{local} is shown to be more effective than using a single p-value. Experiment 4 showed that it is possible to detect more anomalies when using this approach than when using individual p-values. This highlights that each p-value is able to detect different anomalies better than the others.

Local classes perform better at detecting anomalies when provided with the same amount of previous trajectories as both the global and type classes. This indicates that local classes are a better option when computational cost is considered.

The multi-class hierarchy framework could potentially be applied to other anomaly detection problems that involve a class hierarchy.

Chapter 5

Conclusion

This thesis has further explored the application of conformal prediction to the anomaly detection domain. In particular the performance measures used are for conformal predictors applied to anomaly detection. A comprehensive study of measures of efficiency for conformal predictors applied to anomaly detection was presented. Shortcomings with previous criteria were highlighted in the unsupervised setting where there is of a lack of labelled data. In some anomaly detection problems it can be problematic collecting sufficient examples of anomalies. We propose two new criteria, average p-value (APV) and average logarithmic p-value (ALPV), to address this case. These new criteria can be used in the discovery and evaluation of appropriate non-conformity measures for anomaly detection. The key novelty is there now exists an approach suitable for measuring performance of conformal predictors for anomaly detection without the need for labelled examples of anomalies. The APV and ALPV criteria only require a training set consisting of *normal* examples to assess a non-conformity measure. These measures are not limited to this scale and are applicable even if there examples of *anomaly* objects.

Furthermore we presented an example of using these measures to compare two non-conformity measures for an anomaly detection problem. These experiments are conducted with real world data on ship vessel trajectories. A dimensionality reduction package is

used and a comparison of a kernel density based non-conformity measure with a k-nearest neighbours non-conformity measure is presented and the results are discussed.

We presented a multi-class hierarchy to exploit the structure of data. Typically past approaches use a single global class for *normal* objects, or split the classes with little overlap. We demonstrated the use of overlapping classes and did some experimental studies to demonstrate that there are possible benefits to use a multi-class hierarchy. Particularly in the case of aggregating the p-values into a decision rule. We demonstrate that for our data using the minimum p-value aggregate from the hierarchy offers better performance than using any of individual p-values from the hierarchy. This is particularly useful when aiming to minimize false-positives which is a vital goal for any anomaly detector. The multi-class hierarchy framework could potentially be applied to other anomaly detection problems that involve a class hierarchy.

5.1 Future work

The following directions for future research may prove to be interesting:

- One of the current limitations is that the computational complexity of APV grows exponentially with the dimensionality. Investigating applying APV to high dimensional problems would be an interesting challenge. One possible approach is to use Monte Carlo simulations that sample points from the feature space.

- Applying the multi-class hierarchy to different anomaly detection problems. In this thesis we only apply the multi-class hierarchy to the maritime surveillance domain. It would be interesting to see how it holds up for other problems.

- Exploring automatic building of a multi-class hierarchy. It would be interesting to use and combine a clustering algorithm to build a hierarchy structure of classes (perhaps these classes would be representative of various behaviours/patterns from observations). Then investigating how much utility a multi-class hierarchy offers.

Bibliography

[1] Neil Bomberger, Bradley J Rhodes, Michael Seibert, Allen M Waxman, et al. Associative learning of vessel motion patterns for maritime situation awareness. In *Information Fusion, 2006 9th International Conference on*, pages 1–8. IEEE, 2006.

[2] Nicolas Brax, Eric Andonoff, and Marie-Pierre Gleizes. A self-adaptive multi-agent system for abnormal behavior detection in maritime surveillance. In *Agent and Multi-Agent Systems. Technologies and Applications*, pages 174–185. Springer, 2012.

[3] TD Butters, S Güttel, JL Shapiro, and TJ Sharpe. Automatic real-time fault detection for industrial assets using metasensors. 2015.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[5] Ozgur Depren, Murat Topallar, Emin Anarim, and M Kemal Ciliz. An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert systems with Applications*, 29(4):713–722, 2005.

[6] J Du Toit and JH Van Vuuren. Semi-automated maritime vessel activity detection using hidden markov models. In *Proceedings of the 43rd Annual Conference of the Operations Research Society of South Africa, Parys*, pages 71–78, 2014.

[7] Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[8] Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Conformal prediction under hypergraphical models. In *Artificial Intelligence Applications and Innovations*, pages 371–383. Springer, 2013.

[9] Denis Garagic, Bradley J Rhodes, Neil Bomberger, Majid Zandipour, et al. Adaptive mixture-based neural network approach for higher-level fusion and automated behavior monitoring. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[10] DM Green and JA Swets. Signal detection theory and psychophysics. *New York*, 888:889, 1966.

[11] David J Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine learning*, 77(1):103–123, 2009.

[12] Dini Oktarina Dwi Handayani, Wahju Sediono, and Aamer Shah. Anomaly Detection in Vessel Tracking Using Support Vector Machines (SVMs). In *Advanced Computer Science Applications and Technologies (ACSAT), 2013 International Conference on*, pages 213–217. IEEE, 2013.

[13] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.

[14] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[15] Anders Holst, Björn Bjurling, Jan Ekman, Åsa Rudström, Klas Wallenius, M Björkman, Farzad Fooladvandi, Rikard Laxhammar, and J Trönninger. A joint statistical and symbolic anomaly detection system: Increasing performance in maritime surveillance. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1919–1926. IEEE, 2012.

[16] Kira Kowalska and Leto Peel. Maritime anomaly detection using Gaussian process active learning. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1164–1171. IEEE, 2012.

[17] Richard O Lane, David A Nevell, Steven D Hayward, and Thomas W Beaney. Maritime anomaly detection and threat assessment. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8. IEEE, 2010.

[18] Rikard Laxhammar. Anomaly detection for sea surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[19] Rikard Laxhammar. *Conformal anomaly detection: Detecting abnormal trajectories in surveillance applications*. PhD thesis, 2014.

[20] Rikard Laxhammar and Göran 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. ACM, 2010.

[21] Rikard Laxhammar and Göran Falkman. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[22] Rikard Laxhammar and Göran Falkman. Online detection of anomalous sub-trajectories: A sliding window approach based on conformal anomaly detection and local outlier factor. In *Artificial Intelligence Applications and Innovations*, pages 192–202. Springer, 2012.

[23] Rikard Laxhammar, Göran Falkman, and Egils Sviestins. Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 756–763. IEEE, 2009.

[24] Jing Lei, James Robins, and Larry Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

[25] Jing Lei and Larry Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):71–96, 2014.

[26] Steven Mascaro, Ann E Nicholso, and Kevin B Korb. Anomaly detection in vessel tracks using Bayesian networks. *International Journal of Approximate Reasoning*, 55(1):84–98, 2014.

[27] Harikrishna Narasimhan and Shivani Agarwal. A Structural {SVM} Based Approach for Optimizing partial AUC. In *Proceedings of the 30th International Conference on Machine Learning*, pages 516–524, 2013.

[28] Maria Nilsson, Joeri Van Laere, Tom Ziemke, and Johan Edlund. Extracting rules from expert operators to support situation awareness in maritime surveillance. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE, 2008.

[29] International Maritime Organisation. Regulation 19 - Carriage requirements for shipborne navigational systems and equipment. International Convention for the Safety of Life at Sea (SOLAS) Treaty. Chapter V, 2011.

[30] Ying Pan and Xuhua Ding. Anomaly based web phishing page detection. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*, pages 381–392. IEEE, 2006.

[31] Lokukaluge P Perera, Paulo Oliveira, and Carlos Guedes Soares. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1188–1200, 2012.

[32] Bradley J Rhodes, Neil Bomberger, Michael Seibert, Allen M Waxman, et al. Maritime situation monitoring and awareness using learning mechanisms. In *Military Communications Conference, 2005. MILCOM 2005. IEEE*, pages 646–652. IEEE, 2005.

[33] Bradley J Rhodes, Neil Bomberger, Michael Seibert, Allen M Waxman, et al. SeeCoast: Automated port scene understanding facilitated by normalcy learning. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE, 2006.

[34] Bradley J Rhodes, Neil Bomberger, Majid Zandipour, et al. Probabilistic associative learning of vessel motion patterns at multiple spatial scales for maritime situation awareness. In *Information Fusion, 2007 10th International Conference on*, pages 1–8. IEEE, 2007.

[35] Branko Ristic, B La Scala, Mark Morelande, and Neil Gordon. Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In *Information Fusion, 2008 11th International Conference on*, pages 1–7. IEEE, 2008.

[36] Maria Riveiro. Evaluation of Normal Model Visualization for Anomaly Detection in Maritime Traffic. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(1):5, 2014.

[37] María José Riveiro. *Visual analytics for maritime anomaly detection.* PhD thesis, 2011.

[38] Osman Salem, Alexey Guerassimov, Ahmed Mehaoua, Andrian Marcus, and Borko Furht. Sensor fault and patient anomaly detection and classification in medical wireless sensor networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 4373–4378. IEEE, 2013.

[39] Michael Seibert, Bradley J Rhodes, Neil A Bomberger, Patricia O Beane, Jason J Sroka, Wendy Kogel, William Kreamer, Chris Stauffer, Linda Kirschner, Edmond

Chalom, et al. SeeCoast port surveillance. In *Defense and Security Symposium*, pages 62040B–62040B. International Society for Optics and Photonics, 2006.

[40] Hamed Yaghoubi Shahir, Uwe Glasser, Narek Nalbandyan, and Hans Wehn. Maritime Situation Analysis: A Multi-vessel Interaction and Anomaly Detection Framework. In *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, pages 192–199. IEEE, 2014.

[41] James Smith, Ilia Nouretdinov, Rachel Craddock, Charles Offer, and Alexander Gammerman. Anomaly Detection of Trajectories with Kernel Density Estimation by Conformal Prediction. In *Artificial Intelligence Applications and Innovations*, pages 271–280. Springer, 2014.

[42] James Smith, Ilia Nouretdinov, Rachel Craddock, Charles Offer, and Alexander Gammerman. Conformal Anomaly Detection of Trajectories with a Multi-class Hierarchy. In Alexander Gammerman, Vladimir Vovk, and Papadopoulos Harris, editors, *Statistical Learning and Data Sciences*, pages 281–290. Springer, 2015.

[43] Salvatore J Stolfo, Shlomo Hershkop, Linh H Bui, Ryan Ferster, and Ke Wang. Anomaly detection in computer security and an application to file system accesses. In Mohand-Said Hacid, Zbigniew W Ras, and Shusaku Tsumoto, editors, *Foundations of Intelligent Systems*, pages 14–28. Springer, 2005.

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[45] Joeri Van Laere and Maria Nilsson. Evaluation of a workshop to capture knowledge from subject matter experts in maritime surveillance. In *Information Fusion, 2009. FUSION'09. 12th International Conference on*, pages 171–178. IEEE, 2009.

[46] Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *Symposium on Conformal and Probabilistic Prediction with Applications*, pages 23–39. Springer, 2016.

[47] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

[48] Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *AAAI/IAAI*, pages 217–223, 2002.

[49] Xiaojin Zhu. Semi-supervised learning. In *Encyclopedia of machine learning*, pages 892–897. Springer, 2011.