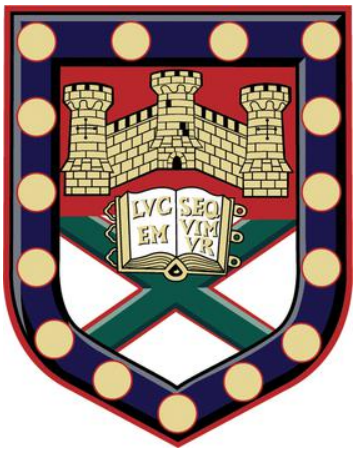
Featured Anomaly Detection Methods and Applications



Chengqiang Huang

College of Engineering, Mathematics and Physical Sciences
University of Exeter

Submitted by Chengqiang Huang to the University of Exeter for the degree of
Doctor of Philosophy in Computer Science

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

Signature:.....

Computer Science Department June 2018

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Chengqiang Huang
June 2018

Acknowledgements

I would like to thank my supervisors, Prof. Geyong Min, Prof. Richard Everson, Dr. Yulei Wu and Prof. Yiming Ying for their kind helps and supports in supervising my works. In addition, I would like to thank Dr. Lejun Chen, Prof. Zhiwei Zhao, Dr. Ke Pei, Dr. Haozhe Wang, Dr. Wang Miao, Dr. Alma Rahat, Prof. Chunlin Wang, Prof. Yang Liu, Dr. Jia Hu, Mr. Jin Wang and Mr. Yuan Zuo for their helpful discussions and comments for my works.

In particular, I would like to thank my mom, Mrs. Shuijiao Zhang, my sister, Mrs. Yongjing Huang, and my lover, Dr. Jingya Liu, who always love me and support me. And I would like to have special thanks to Dr. Lejun Chen and Dr. Yimei Chen for being my greatest friends and mentors during my whole Ph.D. period. Also many thanks to Ms. Zhengxin Yu and Ms. Yang Mi for their supports and helps.

List of Publications

1. **C. Huang**, Y. Wu, G. Min, Y. Ying, Kernelized Convex Hull Approximation and its Applications in Data Description Tasks, *The 2018 International Joint Conference on Neural Networks (IJCNN)*, accepted to appear, 2018.

2. **C. Huang**, Y. Wu, Y. Zuo, K. Pei, G. Min, Towards Experienced Anomaly Detector through Reinforcement Learning, *The 32nd AAAI Conference on Artificial Intelligence (AAAI Student Abstract)*, accepted to appear, 2018.

3. **C. Huang**, G. Min, Y. Wu, Y. Ying, K. Pei, Z. Xiang, Time Series Anomaly Detection for Trustworthy Services in Cloud Computing Systems, *IEEE Transactions on Big Data*, accepted to appear, 2017.

4. **C. Huang**, Y. Wu, Y. Zuo, G. Min, Towards Practical Anomaly Detection in Network Big Data, *Big Data and Computational Intelligence in Networking*, Y. Wu, F. Hu, G. Min, A. Zomaya (editors.), Taylor & Francis/CRC, ISBN: 978-1498784863, Chapter 17, 2017.

5. Y. Zuo, Y. Wu, G. Min, **C. Huang**, X. Zhang, Distributed Machine Learning in Big Data Era for Smart City, *From Internet of Things to Smart Cities: Enabling Technologies*, H. Sun, C. Wang, B. Ahmad (editors.), Taylor & Francis/CRC, ISBN: 978-1498773782, Chapter 6, 2017.

6. **C. Huang**, Z. Yu, G. Min, Y. Zuo, K. Pei, Z. Xiang, J. Hu, Y. Wu, Towards Better Anomaly Interpretation of Intrusion Detection in Cloud Computing Systems, *IEEE COMSOC MMTC Communications - Frontiers*, vol. 12, no. 2, pp. 28-32, 2017.

Abstract

Anomaly detection is a fundamental research topic that has been widely investigated. From critical industrial systems, e.g., network intrusion detection systems, to people's daily activities, e.g., mobile fraud detection, anomaly detection has become the very first vital resort to protect and secure public and personal properties. Although anomaly detection methods have been under consistent development over the years, the explosive growth of data volume and the continued dramatic variation of data patterns pose great challenges on the anomaly detection systems and are fuelling the great demand of introducing more intelligent anomaly detection methods with distinct characteristics to cope with various needs. To this end, this thesis starts with presenting a thorough review of existing anomaly detection strategies and methods. The advantageous and disadvantageous of the strategies and methods are elaborated. Afterward, four distinctive anomaly detection methods, especially for time series, are proposed in this work aiming at resolving specific needs of anomaly detection under different scenarios, e.g., enhanced accuracy, interpretable results and self-evolving models. Experiments are presented and analysed to offer a better understanding of the performance of the methods and their distinct features. To be more specific, the abstracts of the key contents in this thesis are listed as follows:

- Support Vector Data Description (SVDD) is investigated as a primary method to fulfill accurate anomaly detection. The applicability of SVDD over noisy time series datasets is carefully examined and it is demonstrated that relaxing the decision boundary of SVDD always results in better accuracy in network time series anomaly detection. Theoretical analysis of the parameter utilised in the model is also presented to ensure the validity of the relaxation of the decision boundary.

- To support a clear explanation of the detected time series anomalies, i.e., anomaly interpretation, the periodic pattern of time series data is considered as the contextual information to be integrated into SVDD for anomaly detection. The formulation of SVDD with contextual information maintains multiple discriminants which help in distinguishing the root causes of the anomalies.

- In an attempt to further analyse a dataset for anomaly detection and interpretation, Convex Hull Data Description (CHDD) is developed for realising one-class classification together with data clustering. CHDD approximates the convex hull of a given dataset with the extreme points which constitute a dictionary of data representatives. According to the dictionary, CHDD is capable of representing and clustering all the normal data instances so that anomaly detection is realised with certain interpretation.
- Besides better anomaly detection accuracy and interpretability, better solutions for anomaly detection over streaming data with evolving patterns are also researched. Under the framework of Reinforcement Learning (RL), a time series anomaly detector that is consistently trained to cope with the evolving patterns is designed. Due to the fact that the anomaly detector is trained with labeled time series, it avoids the cumbersome work of threshold setting and the uncertain definitions of anomalies in time series anomaly detection tasks.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AMI	Adjusted Mutual Information
ANN	Artificial Neural Network
AR	Auto-regressive model
ARIMA	Auto-regressive Integrated Moving Average model
ART	Adaptive Resonance Theory
AUC	Area Under the Curve
BDD	Binary Decision Diagram
CHDD	Convex Hull Data Description
COF	Connectivity-based Outlier Factor
CRAN	Comprehensive R Archive Network
DDoS	Distributed Denial of Service
DRMF	Direct Robust Matrix Factorisation
DSVDD	Density-induced Support Vector Data Description
DT	Decision Tree
ECG	Electrocardiogram
EEG	Electroencephalograms
EGADS	Extensible Generic Anomaly Detection System

EM	Expectation-Maximisation
ESD	Extreme Studentised Deviate
ES	Exponential Smoothing
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
FSM	Finite State Machine
GMM	Gaussian Mixture Model
GP	Gaussian Process
GPU	Graphics Processing Units
GRNN	Generalised Regression Neural Network
HMM	Hidden Markov Model
HTM	Hierarchical Temporal Memory
iForest	Isolation Forest
KDE	Kernel Density Estimation
KNN	K-Nearest Neighbor
KPCA	Kernel Principal Component Analysis
LDOF	Local Distance-based Outlier Factor
LOF	Local Outlier Factor
LPSVDD+	Linear Programming Support Vector Data Description Plus
LPSVDD	Linear Programming Support Vector Data Description
LR	Linear Regression
LSH	Local Sensitive Hashing
LSTM	Long-Short Term Memory

LUPI	Learning using Privileged Information
MA	Moving Average model
MDP	Markov Decision Process
MLE	Maximum Likelihood Estimation
NMF	Nonnegative Matrix Factorisation
NNLS	Non-Negative Least Square problems
OCSVM	One-class Support Vector Machine
PCA	Principal Component Analysis
PDE	Parzen Density Estimation
QP	Quadratic Programming
RLPSVDD	Relaxed Linear Programming Support Vector Data Description
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RNN	Replicator Neural Network
RPCA	Robust Principal Component Analysis
RRI	Rank-one Residual Iteration
Semi-NMF	Semi-Nonnegative Matrix Factorisation
SOM	Self-organising Map
SVDD	Support Vector Data Description
SVM	Support Vector Machine
SVR	Support Vector Regression
SV	Support Vectors
TN	True Negative
TP	True Positive
YML	Yahoo Membership Login

Chapter 1

Introduction

Anomaly detection is a fundamental research topic that has gained much research attention in various application domains. From critical industrial systems, e.g., network intrusion detection systems, to people's daily activities, e.g., mobile fraud detection, anomaly detection has become the most critical and very first resort to protect and secure the public and personal properties. With the consistent development over the years, the gradual perfection of data collecting, cleaning and integrating have backed anomaly detection in diverse areas. Nevertheless, the explosive growth of data volume and the continued dramatic variation in data patterns pose great challenges on the anomaly detection systems and are fuelling the great demand of introducing more intelligent anomaly detection methods with distinct characteristics to cope with various needs of anomaly detection. Therefore, this thesis is dedicated to offering innovative anomaly detection methods with distinct features so as to suffice the specific requirements under varying anomaly detection scenarios. Before introducing the methods, this chapter firstly clarifies the related concepts of anomaly detection, e.g., outlier detection and novelty detection. Then, diverse aspects of anomaly detection, for example, the types of anomalies and the general taxonomy of methods, are further discussed. With the related concepts and taxonomy being addressed, the research problems, challenges and aims of this thesis are delivered. Finally, the outline of this thesis is presented.

1.1 What is an Anomaly? Outlier or Novelty?

Anomaly detection is a concept encompassing a broad spectrum of techniques concerning the detection of abnormality. In related literature, anomaly detection has different names, such as outlier detection, novelty detection, noise detection and deviation detection. These names are often used interchangeably. In this thesis, "anomaly detection" is utilised as a general term for all related names, while "outlier detection" and "novelty detection" are emphasised

as two primary distinct concepts in anomaly detection. To put it formally, assume a general dataset $X = \{x_1, x_2, \cdots, x_n\}$ with $x_i \in \mathbb{R}^d$, $i \in \{1, 2, \cdots, n\}$, where the notations n and d are the number and dimension of the data instances in the dataset, respectively. The following definitions are summarised to differentiate outlier detection from novelty detection.

> ***Outlier Detection*** is the process of the identification of an observation $x \in X$ (or a **subset** of the observations $X_{sub} \subset X$) which appears to be inconsistent with the remainder of the given observations X.

> ***Novelty Detection*** is the process of the determination of the novelty of a **new** observation y (or observations) according to the known observations X, where $y \in \mathbb{R}^d$ and $y \notin X$.

It is now clear that the concept of anomaly detection involves **1)** the identification of the abnormal data, e.g., noise or outlier, from the original dataset and **2)** the discovery of novel data instances based on the knowledge learned according to the original dataset. From the perspective of machine learning, in outlier detection, the training dataset contains the anomalies that should be pinpointed, while, in novelty detection, the training dataset has no anomaly. In the latter scenario, it is the testing dataset that should be examined for anomalies. Therefore, outlier detection and novelty detection share prominent distinction. Nevertheless, due to the reasons that the term "anomaly detection" is generally used synonymously with "outlier detection" and "novelty detection", and the solutions for anomaly detection, novelty detection and outlier detection often share similar principles, the related work of this thesis aims to consider all such detection schemes and variants.

1.1.1 Types of Anomalies

To the end of better understanding anomaly detection, the taxonomy of the types of anomalies is of fundamental importance. In [29] and [101], anomalies have been categorised into point anomaly, collective anomaly and contextual anomaly. A detailed analysis of this taxonomy reveals intense overlapping between the contextual anomaly and the collective anomaly, while point anomaly may also stem from its anomalous context. Hence, this thesis categorises anomalies into four detailed classes: **1) point anomaly**, i.e., a data instance that is much different from others; and **2) group anomaly**, i.e., a group of data whose patterns or properties deviate significantly from similar groups of other data. When the contextual information is considered as the source data for anomaly detection, a detected point anomaly is called a **3) contextual point anomaly** and a group anomaly is a **4) contextual group anomaly**. A more formal explanation of the concepts are given as follows:

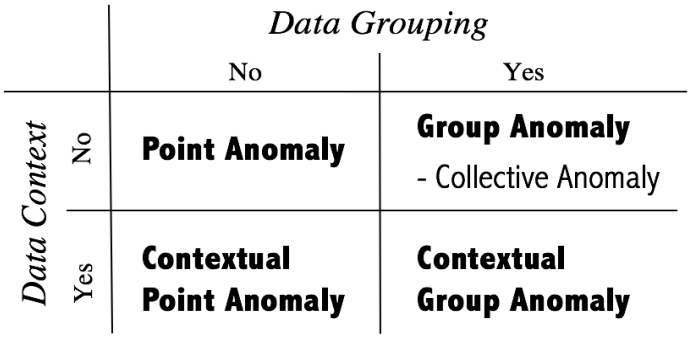


Fig. 1.1 The taxonomy of anomalies

- **Point anomaly** is an observation x or y that deviates remarkably from X according to some predefined criteria, where $x \in X$ and $y \notin X$. For instance, the absence of a student in the Math lesson on Monday morning during the term time is a point anomaly, because, different from other students, the student is absent.
- **Group anomaly** is a set of observations X_o or Y_n, which is grouped based on a predefined criterion, that does not follow the regular patterns of other sets of observations according to certain definitions of the regular patterns, where $X_o \subset X$ and $Y_n \nsubseteq X$. For example, the absences of the students who sat in the first row of the classroom in the Math lesson on Monday morning during the term time is a group anomaly. This is due to the reason that other groups, e.g., the students sat in the second row, are present.
- **Contextual point anomaly** is an observation x or y that deviates remarkably from X according to some predefined criteria under certain context, where $x \in X$ and $y \notin X$. An example of this is the presence of a student in the classroom during the summer vacation. This is because although the presence of the student is usual, the time (context) when the student appears is unexpected.
- **Contextual group anomaly** is a set of observations X_o or Y_n, which is grouped based on a predefined criterion, that does not follow the regular patterns of other sets of observations according to certain definitions of the regular patterns under certain context, where $X_o \subset X$ and $Y_n \nsubseteq X$. Similarly, the presence of a student in the classroom during a period of the summer vacation (a group of contexts) is unanticipated and regarded as a contextual group anomaly.

Note that the utilisations of x, X_o and y, Y_n in the explanations are to differentiate the outlier detection and novelty detection. In reality, the types of anomalies, as well as other factors discussed later, are shared between outlier detection and novelty detection. Therefore, the uses of y and Y_n are omitted later for brevity.

As shown in Fig.1.1, this categorisation emphasises the way in which anomalies are detected, i.e., from individual observations or groups of observations, and particularly stresses contextual information as a critical factor for the classification. It maintains point anomaly, collective anomaly (as a type of group anomaly) and contextual anomaly, but details the contextual anomaly according to data grouping. As a result, specific contexts can be examined for contextual anomaly detection upon diverse types of datasets, e.g., time series, graphs, videos and profiles. In the next part, the properties of various datasets are elaborated with the focus on the different contexts in anomaly detection.

1.1.2 Types of Datasets and Contexts

A key and fundamental aspect of any anomaly detection technique is the nature of the target dataset. Essentially, a dataset is a collection of data instances or observations. According to specific application scenarios, a data instance can be a number, record, video, song, graph, image, event, profile, etc. All these disparate forms of data should be transformed into general data types for the purpose of anomaly detection. Temporally, general data types comprise **1) scalar**, **2) vector**, **3) matrix** as well as **4) tensor**, and their elements are known as data attributes, features, or fields, which can be **1) numerical** or **2) categorical**. In literature, the scalar is called univariate, while the vector, matrix and tensor are all multivariate data types. To sum up, the taxonomy of datasets from the two perspectives is shown in Fig.1.2.

Besides the original dataset, the contexts under which the data instances are observed are another crucial source of information that is helpful in detecting abnormal events/behaviors/etc. of a target system/object/etc. Usually, the contextual information is distinctive and not measured or recorded explicitly in different applications. The context that is helpful for anomaly detection is always obscure. Therefore, in this section, only ubiquitous contexts are considered, i.e., **1) spatial context**; **2) temporal context**; **3) spatial-temporal context**. Specifically, spatial context concentrates on the location where a data instance is observed, while temporal context reveals the sequential information among observations. The integration of the two contexts motivates the spatial-temporal context which has attracted much attention in recent years. To analyse contextual information in anomaly detection, two basic strategies exist. On one hand, feature engineering is adopted to consolidate the contextual information and the original information. A well-developed example is the time embedding technique for time series analysis in which temporal information is critical. On the other hand, contexts are analysed separately for contextual anomalies. For instance, the year-on-year growth of the financial income in a company instantiates the contextual information of previous incomes and can be investigated for contextual anomalies.

Data type	• Scalar } univariate • Vector • Matrix } multivariate • Tensor
Element type	• Categorical • Numerical
Context type	• Spatial • Temporal • Spatial-temporal • others

Fig. 1.2 The taxonomy of datasets

1.1.3 Types of Solutions

To provide a valid solution for anomaly detection, the concept of normality is strictly required. Only with a clear concept or definition of normality, the quantified measurement is achievable. In general, depending on the availability of manual definition of normality, i.e., data labels, the solutions for anomaly detection cover three fundamental categories: supervised, semi-supervised and unsupervised methods:

- **Supervised anomaly detection** models both normality and abnormality. It requires the availability of labels for the definitions of normality and abnormality. Consequently, supervised anomaly detection is essentially a *classification* problem that differentiates normal data from the abnormal ones. Theoretically, supervised anomaly detection is superior in its overall accuracy due to the clear understanding of normality and abnormality. Nevertheless, some practical problems immensely hinder its utilisation. Firstly, the data labels are usually not available or extremely costly to obtain under many scenarios, e.g., the label of a configuration of a network is not entirely clear unless the network is practically run and examined. Secondly, the data labels may not be balanced. Typically, in practical anomaly detection problems, the normal samples greatly outnumber the abnormal ones, which results in a prominent bias in the classification model that may degrade the performance of anomaly detection. Lastly, the involvement of both normal and abnormal data may introduce more noise into the model, hence the disgraced performance.

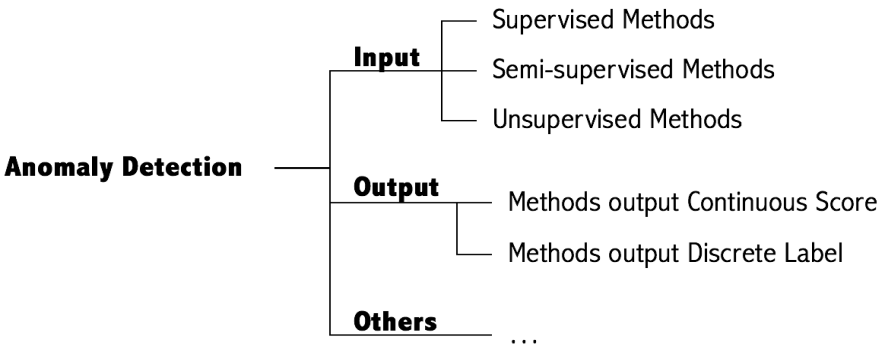


Fig. 1.3 The taxonomy of solutions

- **Semi-supervised anomaly detection** has only normal data samples or only abnormal ones as the inputs. It endeavors to model a single concept and achieves anomaly detection according to the fitness of the data in the concept. Therefore, semi-supervised anomaly detection constitutes a *one-class classification* problem. In comparison to the classification problem, one-class classification requires only normal or abnormal samples, which is more feasible in reality. In addition, due to the sole type of samples, one-class classification negates the problem of imbalance dataset. The very problem concerning the dataset is the inaccurate or noisy data instances that ask for high robustness of the one-class classification methods.

- **Unsupervised anomaly detection** is typically employed in the situation where no prior knowledge of the dataset is known. In other words, no label information is presented. An anomaly detection method has to analyse the dataset to infer the real concept of abnormality or make an assumption of the concept. A concrete example of this type is the set of *clustering-based anomaly detection methods* which presume the data that rest inside small clusters are prone to be anomalous. Unsupervised anomaly detection enjoys similar merits of semi-supervised anomaly detection, while it is always criticised because of the validity of the assumptions made in related tasks.

Distinct scenarios have shown different preferences of the solutions. The three general types of anomaly detection solutions are offered according to the viability of the data labels, i.e., the input of an anomaly detection method. From another perspective, i.e., the output, current anomaly detection supports two typical types of solutions: methods that output **1) continuous scores** and **2) discrete labels**. The continuous score would be preferred by systems that demand detailed analysis of the data instances or favor the adaptive concept of abnormality, while the discrete label is more convenient for users and it greatly simplifies the anomaly detection system design in term of threshold setting for anomalies.

Here, categorisations of the anomaly detection solutions according to their input and output characteristics are presented (Fig.1.3). In chapter 2, a more detailed taxonomy of anomaly detection methods will be given according to the underlying techniques.

1.1.4 Types of Applications

Anomaly detection is a pivotal data analysis tool that finds extensive use in a wide variety of applications. From people's daily life, e.g., health monitoring, to the normal operations of the government, i.e., intrusion detection, anomaly detection plays critical roles. A rough categorisation of the applications is presented in the following list which sorts the applications according to the number of required data sources and detection targets:

- **Single data source, single detection target**: Applications with a single data source for anomaly detection is comparatively easy to deal with. Such applications demand the preprocessing of a sole data type and have a clear idea of the target anomalies. Fraud detection, image novelty detection, meter monitoring and etc. are all instances of this type. Supplied with useful information, such as transaction records, normal images, meter readings and etc., an anomaly detection method should pinpoint anomalous transactions, novel images, erroneous meter readings and etc.

- **Multiple data sources, single detection target**: Anomaly detection applications with multiple data sources, e.g., health monitoring, behavior detection and city traffic monitoring, are relatively hard to analyse. They accept information from a number of data sources. For example, to support health monitoring, the electrocardiogram (ECG), electroencephalograms (EEG), blood pressure and etc. are acquired. The anomaly detection over all the information is to conclude with a single result of whether the monitored patient is healthy. Therefore, while the data sources are somewhat redundant, the detection target is clear.

- **Multiple data sources, multiple detection targets**: Applications, such as network intrusion detection or network fault diagnosis, require anomaly detection as the very fundamental tool to process data from diverse sources. The detections of anomalies in distinct data sources typically reflect different intrusions or faults within the target system. As a result, in systems where detailed analysis of anomalies are mandatory, the task of general anomaly detection covers multiple facets and is much more complex. An anomaly detection system or an anomaly diagnosis engine is demanded for higher level analysis, e.g., anomaly correlation analysis, in order to generate valuable guidance in network intrusion prevention or fault prevention.

Generally, the taxonomy above reflects the complexity of different applications. For simpler applications, the difficulties lie in the data preprocessing process and the anomaly detection process. While in more complex applications, the anomaly detection provides the basic information for further analysis. In both cases, the anomaly detection method plays the core role in the applications. This thesis will, therefore, investigate the most up-to-date anomaly detection methods/systems and existing research problems. Endeavors will be made to achieve practical anomaly detection in some applications.

1.2 Research Problems, Challenges and Objectives

1.2.1 Problems and Challenges

While many anomaly detection methods/systems are attractive and solid theoretically, a host of technological problems need to be overcome before they are practically adopted in various areas. These problems generally concern the accuracy, efficiency, and other capabilities, e.g., interpretability, scalability, etc. Despite the fact that many research efforts have been conducted in dealing with miscellaneous cases, this thesis particularly targets at three critical problems witnessed in time series anomaly detection applications and addressing corresponding difficulties:

- **The high false alarm rate**: In systems where time series analysis is required, the false alarm rate of an anomaly detection method is a vital criterion for deciding its applicability. In critical systems, such as the automatic driving system in an airplane, false alarms are strictly unacceptable. However, many existing anomaly detection methods have high false alarm rates especially when the target environment is noisy. Therefore, better methods are urgently demanded. To achieve better accuracy in anomaly detection, supervised learning is always preferred. However, the availability of labeled data is usually a major issue that hinders the utilisation of supervised learning methods. In addition, training datasets always contain noise that introduces degraded accuracy in sensitive anomaly detection methods. Last but not least, the normal or abnormal patterns are often inexhaustive. The differentiation of normality from abnormality is a challenging problem in the presence of limited patterns. All these difficulties contribute to the doubtful guarantee of the accuracy in anomaly detection.

- **The simplified analysis of contextual information**: The contextual information, e.g., periodicity and trending, in time series analysis is essential for the determination of time series anomalies. Traditional methods typically consider specific contexts and do

not generalise in dealing with novel contexts. For instance, time series differencing is a specific technique for trending analysis in applications such as sales prediction, but it does not consider the periodicity within the time series. Therefore, investigating a general method that is capable of adaptively analysing different contexts is another attractive research problem. For distinct application domains, the exact notions of the contexts are always different. Applying a contextual anomaly detection technique developed in one domain to another is generally not straightforward. Moreover, in a specific domain, various contexts exist such that pinpointing the most useful context for anomaly detection is not an easy task. Furthermore, the criteria for detecting different contextual anomalies usually vary significantly. As a result, although attractive the in-depth analysis of contextual information is very challenging.

- **The incapability of handling dynamic data patterns**: In online time series analysis, the pattern of the time series always evolves along with the change of the underlying system. For example, the pattern of the metric which measures the speed of the engine in a car changes according to the behavior of the car. However, few existing methods are capable of handling dynamically changing data patterns of a sequential dataset, especially when the new patterns are unknown. This has raised a hot research problem in applications such as intrusion detection systems where intruders exhibit diverse/changing patterns of intrusion behaviors. General speaking, the dynamicity is a dominant issue concerning the analysis of sequential data. In many domains, normal behavior keeps evolving and the concept of normality is temporary. Therefore, it requires the incremental updating of the anomaly detection model to keep the pace with the evolvement of the notion of normality. Nonetheless, not all the anomaly detection methods support incremental data analysis. And the design of an incremental anomaly detection method is not a painless task.

1.2.2 Objectives

This thesis aims at proposing practical methods for anomaly detection and particularly targets at easing the aforementioned challenges in time series anomaly detection scenarios. More specifically, methods will be introduced to **1)** enhance the accuracy of time series anomaly detection in network systems, **2)** analyse contextual information and anomalies for better anomaly interpretation and **3)** cope with the challenge of dynamicity in sequential anomaly detection. Further details will be outlined in Section 1.3.

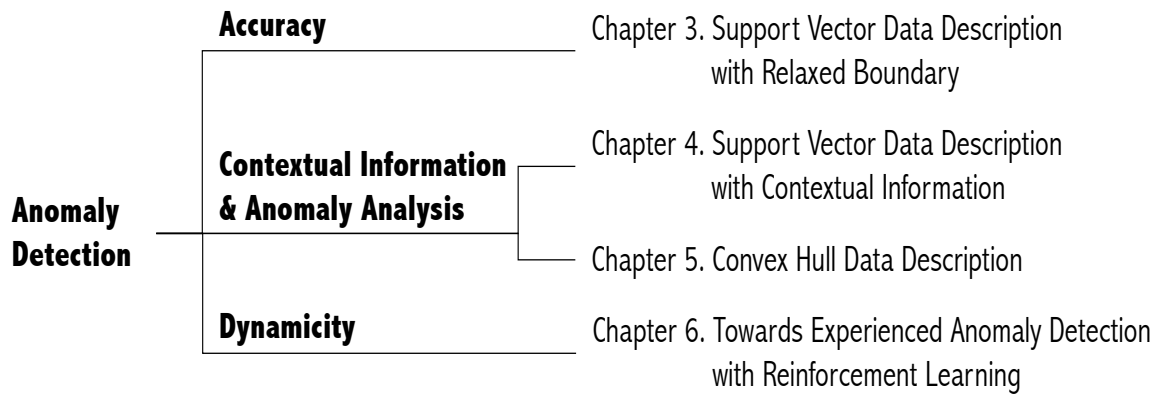


Fig. 1.4 Key contents of the thesis

1.3 Thesis Outline and Contributions

In this chapter, the introduction of the related contents of general anomaly detection has been elaborated. From the next chapter (Chapter 2), detailed related works and state-of-the-art anomaly detection methods will be thoroughly surveyed. The contributions to the family of anomaly detection methods follow from Chapter 3 to Chapter 6 (see Fig.1.4).

- Chapter 2 presents a thorough review of anomaly detection strategies and techniques. For general anomaly detection, four major strategies are identified and five categories of specific techniques are summarised. On the other hand, two basic strategies and four types of techniques are elaborated for time series anomaly detection. This review builds a solid background for the methods proposed in this thesis.

- Chapter 3 proposes an anomaly detection method that relaxes Support Vector Data Description (SVDD) with additional information for better anomaly detection performance. More specifically, the method adopts linear programming method to implement SVDD and relaxes its anomaly detection boundary for network time series anomaly detection. The experiment results demonstrate that the method greatly enhances the overall accuracy of time series anomaly detection.

- Besides the desire for better accuracy, practical applications introduce additional requirements in the actual network anomaly detection process, e.g., the needs for analysing contextual information and classifying the anomalies for different responses. Consequently, Chapter 4 considers specific contextual information in network time series and suggests contextual SVDD in order to integrate the information in anomaly detection for time series anomaly classification.

- In Chapter 3 and Chapter 4, SVDD is the leading method to be utilised in time series anomaly detection with distinct improvements. However, considering the situation where the anomaly detection method is expected to not only identify the anomalies but also cluster or classify the anomalies into different categories such that different responses could be initiated accordingly, SVDD is not the best choice. Hence, Chapter 5 develops Convex Hull Data Description (CHDD) that succeeds in one-class classification and clustering at the same time.

- In both SVDD, CHDD and many other methods, there is a significant limitation that these methods require an inconvenient process of parameter tuning. As a result, in complex problems, a large amount of human labor is required for tuning the parameters, which is undesirable and has made many methods impractical. Moreover, in situations where the data patterns are dynamically evolving, the manual tuning of the parameters is impossible. To ease the problem, Chapter 6 recommends a framework for parameter-free sequential data anomaly detection based on Reinforcement Learning (RL). The framework not only supports parameter-free anomaly detection but also dynamically evolves the anomaly detection method to learn novel data patterns.

To summarise with the key contents, this thesis researches the problem of anomaly detection, especially time series anomaly detection, and advises a number of novel methods to promote anomaly detection from diverse facets. The proposed methods are expected to contribute positively to the family of anomaly detection and prompt more valuable improvements for practical anomaly detection. In the next chapter, a comprehensive review of existing solutions for anomaly detection is provided. In addition to this, each primary chapter also makes some extra efforts to explain its contributions and most related works to make clear the contents in the chapter.

Chapter 2

Related Work

In Chapter 1, an overview of the anomaly detection related concepts, e.g., outlier detection and novelty detection, is presented coupled with the research problems, challenges and aims of this thesis. In this chapter, a comprehensive review of existing anomaly detection strategies and methods is provided. Anomaly detection strategies are high-level and abstract methodologies that guide the process of anomaly detection, while anomaly detection methods concern the detailed techniques and tools that are employed for data analysis. This chapter starts with the introduction of anomaly detection strategies (Section 2.1) and elaborate various types of anomaly detection methods afterward (Section 2.2). Because time series anomaly detection is a key topic in this thesis, related methods for time series anomaly detection are surveyed separately in Section 2.3. Section 2.4 briefly discusses the outline of the remaining contents in this thesis.

2.1 Strategies for Anomaly Detection

According to different scenarios, distinct anomaly detection strategies exist. Generally, there are four types of anomaly detection strategies: 1) **rule**-based anomaly detection; 2) **case**-based anomaly detection; 3) **expectation**-based anomaly detection; and 4) **property**-based anomaly detection. These four strategies correspond to distinct sets of anomaly detection methods and are applied under different scenarios. The rule-based strategy concentrates on identifying explicit/implicit rules to distinguish anomalies from normal data instances. The case-based strategy, on the other hand, tries to pinpoint a relevant case of the target case in order to analyse its abnormality. The expectation-based strategy generates an expected concept of the normality/abnormality which is utilised for anomaly detection, while the property-based anomaly detection investigates the latent properties among all the data and determines the abnormality based on the properties.

2.1.1 Rule-based Strategy

The rule-based strategy is generally applicable in a wide variety of domains. A typical instantiation of the rule-based strategy is a rule-based anomaly detection system that encompasses a set of *rules*, a bunch of facts and an interpreter for applying the rules and facts. In a specific application, the corresponding facts are always determined. It is the rules that are to be learned in order to support accurate anomaly detection. On one hand, the rules are usually designed by human experts who possess strong knowledge of how to determine anomalies in the specific application. On the other hand, rules can be learned according to labeled datasets through classification methods, e.g., Decision Tree (DT) [220] and Support Vector Machine (SVM) [38]. The choices of how to design the rules vary in different application domains. However, the essential idea of the rule-based anomaly detection, i.e. applying discovered rules to identifying anomalies, is identical.

The rule-based strategy is the basic standpoint of many methods discussed in later sections. The generalised concept of "a rule" includes the underlying models in various classifiers, one-class classifiers and many other anomaly detectors. From a high standpoint of view, the rule-based strategy has two potential concerns that may hinder its practical applications:

- Complicated or novel data instances may escape the rule-based anomaly detection system due to the **inaccuracy or absence of the corresponding rules**;
- Static rules in the rule-bases anomaly detection system may not adapt well to the **evolving data patterns**.

As one shall see shortly, these concerns are not severe in other strategies and different methods do not share common solutions to the above problems. Therefore, methods that root on the rule-based strategy have to consider the potential problems independently. The robustness and the capability of incremental learning in different methods are two critical research topics that are attracting more and more attention.

An Example - Rule-based Reasoning System

In network systems, conventional rule-based reasoning systems [87] are pervasive in traffic control systems and intrusion detection systems. Consider a simple paradigm as in Fig.2.1, the rule-based reasoning system contains a set of rules and a set of facts that are both determined by expert experience. Also, an additional inference engine is responsible for anomaly detection based on the reasoning according to the rules and facts. Such rule-based reasoning systems are always efficient in detecting simple network problems and easily acceptable by most engineers.

Rule 1: If A and C then Y

Rule 2: If A and X then Z

Rule 3: If B then X

Rule 4: If Z then D

Fact 1: A indicates E

Fact 2: Y collides with Z

Fig. 2.1 The rules and facts in a rule-based reasoning system

However, in traditional rule-based reasoning systems, rules are often built relying on expert knowledge. It is apparent that 1) the rules are impossible to exhaust such that new problems can circumvent the system easily; 2) with the growth of the number of rules, they are becoming increasingly hard to maintain; 3) the rules added by different experts may conflict with each other [232], which undermines the consistency of the entire system. These problems have promoted the innovations of the form of rules and the systems. Methods, such as DT, are designed to learn rules based on different models which refrain from certain aforementioned drawbacks.

2.1.2 Case-based Strategy

Distinct from the rule-based strategy which summarises a relatively concise knowledge for anomaly detection, the case-based strategy seeks relative *cases* of a target case to help with anomaly identification. At a first glance, the concept of the case-based strategy is highly related to nearest neighbor based methods which determine the abnormality of a data instance based on the analysis of its nearest neighbors. However, the case-based strategy is more powerful in the applications where the target object is complicated and concluding the rules for anomaly detection is cumbersome. For instance, a user profile dataset is rather complex in the sense that it contains multiple types of data, e.g., image information, categorical information, numerical information and etc. As a result, summarising rules for anomaly detection becomes troublesome and ineffective. The case-based strategy is, therefore, more suitable that it focuses on the identification of similar cases of the target case and largely reduces the size of the relevant data for analysis. In many complex applications, e.g., network anomaly detection [213][214], identifying the close related cases is the very first step that greatly boosts the process of anomaly detection.

The case-based strategy has driven some practical methods that are widely in use. A detailed method that implements case-based strategy is the K-Nearest Neighbor (KNN) method [2] which has numerous applications. Another high-level instance is the signature-based anomaly/object detection methods that have been adopted in many real-life scenarios [137][169]. The details of the related methods will be elaborated in Section 2.2.

Although regarded as a promising resort in complicated applications, the case-based strategy is not without its drawbacks. There always exist two primary concerns that should be taken into consideration while utilising the case-based strategy:

- The case-based strategy involves computing the similarities between the target case and all the other cases, which incurs **significant computational complexity**, especially when the number of cases is huge and the similarity measurement is relatively complex.
- The case-based anomaly detection strategy greatly relies on the similarity measurement between two cases. A good similarity measurement helps with the effective differentiation between normal and abnormal cases, while a bad similarity measurement largely degrades the performance. In many applications with complex data, such as graphs and sequences, **defining an effective measurement could be really challenging**.

Despite the concerns, the case-based strategy does maintain nice properties that, as the number of cases keep increasing, the anomaly detection process gains more and more confidence in its decisions. It naturally supports streaming data and enhances its capability with the increment of the dataset.

An Example - Case-based Reasoning System

A case-based reasoning system [124] vividly implements the case-based anomaly detection strategy and largely avoids the limitations in the rule-based reasoning systems. The essential idea of a case-based reasoning system is to represent former problem-solving experience as cases which are stored in a centralised library. Confronted with a new problem, the system retrieves similar cases and summarises valuable information to solve the current problem. The novel experience with the proposed solution is to be confirmed and added to the library for future reference. 1) With the accumulation of informative cases, the system adaptively evolves according to the experience; 2) the solutions of the previous cases can be generalised to offer solutions to unseen problems; and 3) the whole system does not require extensive maintenance experience. To be more concrete, Fig.2.2 depicts the workflow of a typical case-based reasoning system. Four major steps are involved:

- Retrieval: This step retrieves relevant cases from the case library in order to solve a target problem. A comprehensive case could contain the description of a problem, its solutions and other related information, while a simple case could be a single data instance along with its label;

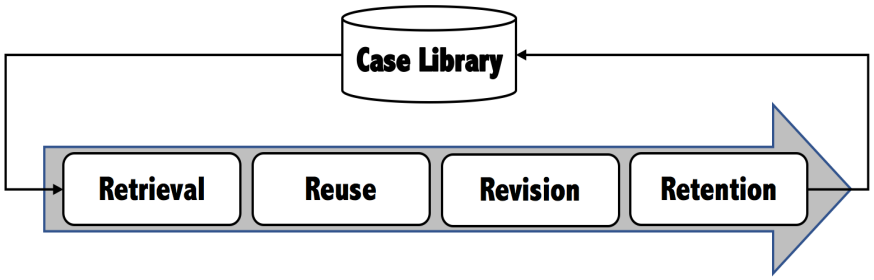


Fig. 2.2 The workflow of a case-based reasoning system

- Reuse: The reuse of the relevant cases is to summarise valuable information from previous solutions of relevant problems and determine the solution for the current problem;
- Revision: The solution is revised in this step according to the specifications of the current problem. It outputs a revised solution for testing;
- Retention: If the revised solution successfully solves the current problem, the case, i.e., the problem and its solution, is retained in the case library.

Within the four steps, the retrieval of the relevant cases is of high significance. It is also the step which consumes most computation resources and demands effective similarity measurements. The complexities of the steps of reuse and revision, however, vary under diverse scenarios. Overall the process of anomaly detection using the case-based reasoning system is greatly simplified due to the detached steps, which is suitable for different groups of experts to work on in an attempt to solve complex anomaly detection problems, e.g., network anomaly detection in which a case contains complex network information and network anomaly descriptions.

2.1.3 Expectation-based Strategy

The expectation-based strategy features in the utilisation of the expected concept of normality to determine anomalies. In other words, whenever a data instance is beyond *expectation*, it is regarded as anomalous. Generally speaking, the expected concept of normality comes in two distinct forms:

- Data probability, which connects intensely to information theoretic methods that analyse the probability of the occurrence of a specific data instance;
- Data estimation, which has been widely implemented by regression and reconstruction methods to measure the error between the actual data and the estimated data.

In data probability analysis, distributions of normal data are analysed and formulated to assign the probability of a data instance being anomalous. However, in data estimation, the expected data values are calculated directly from normal data regardless of their probabilities. These two forms can also be unified to supply probabilistic estimation of the concept of normality. The choice of these forms depends heavily on the applications. As a concrete example, in time series prediction for anomaly detection, a probabilistic estimation of a future value presents not only the expected value but also the variance so that providing the confidence for the prediction. On the other hand, in multimodal data anomaly detection, estimating the mean of the dataset is not profitable but measuring the mixed probabilistic distribution can be much more helpful.

Although different in form, data probability analysis and data estimation share the same principle that they extract knowledge from the given dataset and obtain the distances between expectations and realities. The distances are further analysed through a thresholding process to determine the eventual anomalies. This expectation-based strategy is straightforward to understand and extensively applied in applications where the data show stable patterns. For instance, in problems such as online time series anomaly detection, expectation-based anomaly detection is always the very first strategy to be considered, especially when the time series demonstrates strong patterns, e.g., periodic patterns, that are beneficial for prediction. However, the expectation-based strategy is not without its drawbacks. The two most prominent problems concerning the employment of the strategy are:

- The performance of the methods in data probability analysis and data estimation always **rely on the model used for fitting the training dataset**. An unbefitting model will consume extra computational power and result in poor performance.
- Expectation-based anomaly detection typically asks for a thresholding process to determine the eventual anomalies. Nevertheless, the **selection of the threshold is nontrivial and application-specific**, which causes potential troubles for accurate anomaly detection.

An Example - Expectation-based Anomaly Detection System

Generally, an expectation-based anomaly detection system consists of four components as shown in Fig.2.3. 1) The training dataset contains only the original data instances with no label. It is responsible for training the selected model which is expected to generalise and fit the testing dataset. 2) The testing dataset has the identical format of the training dataset and is exploited to examine the validity of the learned model. 3) The data estimation or probability analysis model is the core component of the system. During the training phase,

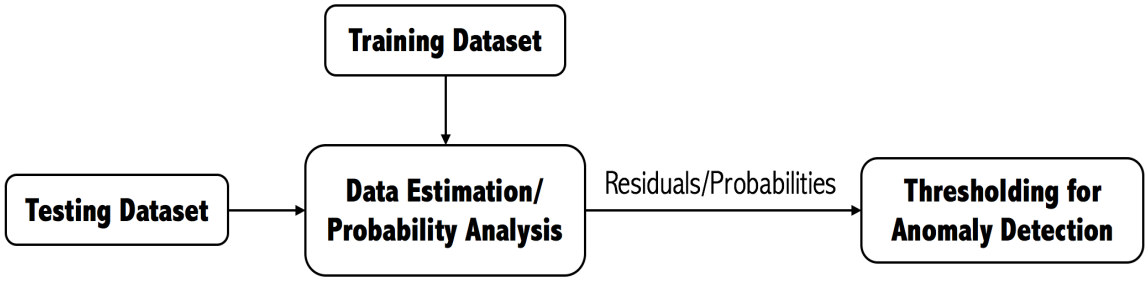


Fig. 2.3 The components of an expectation-based anomaly detection system

the model takes the training dataset as input and tunes the model parameters to fit the dataset as closely as possible, e.g., likelihood maximisation and error minimisation, while in the testing phase the model takes the testing dataset as input and outputs anomaly scores of the data instances, e.g., reconstruction errors/residuals and data probabilities. 4) The anomaly scores are examined by a thresholding component for the ultimate decisions of anomaly detection. Usually, the threshold is assigned as a byproduct of the trained model. Nonetheless, in many applications where no label is provided for the determination of anomalous data, obtaining a practical threshold can be troublesome and always desires online adjustment.

To illustrate the concrete examples of data estimation and probability analysis, consider firstly the Replicator Neural Network (RNN) [94], i.e., a fully connected neural network whose inputs and outputs are the same and the size of the hidden layer is smaller than that of the input and output layers. The neural network is trained such that its outputs mimic the inputs as closely as possible, i.e., reconstructing the inputs. During the testing phase, the reconstruction errors/residuals between the inputs and outputs are calculated with a chosen measurement to reflect the distances between the tested data and their corresponding expectations. On the other hand, consider the Gaussian Mixture Model (GMM) [14] which exploits multiple Gaussian distributions to fit a given dataset during the training phase and outputs the probability of a data instance belonging to the mixed distribution during the testing phase. GMM essentially interprets the expectation of data using the data probabilities derived from the learned distribution. The probabilities, therefore, constitute anomaly scores for ultimate anomaly detection.

2.1.4 Property-based Strategy

Besides the aforementioned strategies, a relatively novel anomaly detection strategy which is based on the latent *properties* in data instances attracts increasing attention in recent years. This property-based strategy assumes that there always exist static or stable properties in the system that generates the target dataset. With the identification of the properties, they

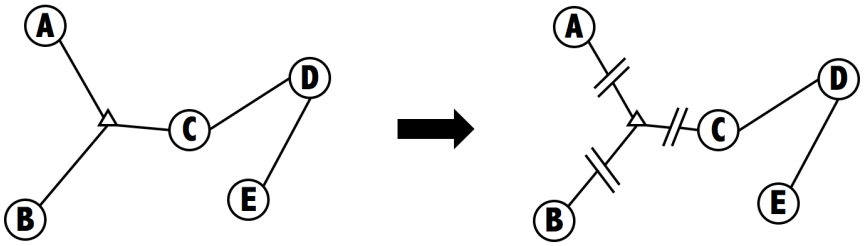


Fig. 2.4 The basic idea of a feature correlation anomaly detection method

can be employed as the criterion to differentiate normal data instances and abnormal ones, i.e., abnormal data instances will not exhibit certain properties indicating that they are not generated by the underlying system. Armed with the assumption, a number of methods have been proposed to mine the latent properties. Two representative examples are 1) feature correlation, which explicitly finds the stable internal relationship among data features; and 2) data compression, which implicitly measures the normal amount of information hidden in the data. Therefore, as long as a data instance violates the stable internal relationship or the normal amount of information, it is pinpointed as anomalous.

Feature correlation and data compression are only two inchoate methods that await further investigation. There are many other techniques that have been well developed, such as Principal Component Analysis (PCA) [102], Kernel Principal Component Analysis (KPCA) [202] and etc. All these methods recognise certain properties of the dataset and leverage these properties for the purpose of anomaly detection. If the properties are accurately identified, the methods always exhibit extraordinary performance. Unfortunately, two practical issues have made the applications of the methods difficult:

- Given a random dataset, **pinpointing the best property to extract is a troublesome task**. Typically, no preference of methods can be made without examining their actual performances.
- **Determining the violation of a certain property can be tricky.** For instance, in feature correlation, the change of the relations among the features can be hard to measure due to the varied intensity and existence of the relations.

An Example - Feature Correlation Analysis

Consider the example in Fig.2.4 where feature correlation is employed as the key property to identify anomalies [81][211][103]. The circles are features and the link between two circles represents the intense correlation between the features. The triangle and connected links indicate the relation among several features.

On the left part of Fig.2.4, a stable set of feature correlations is recognised from the training dataset as a benchmark graph to define normality. Whenever the normal set of feature correlations is largely violated in a data instance, e.g., the right part of Fig.2.4 where the correlations among features A, B and C are corrupted, it is highly suspicious that the data instance is anomalous. This workflow is intuitionistic and features in the way that 1) it achieves feature selection through the process because the features which have no relation to others are not helpful for anomaly detection and can be ignored; 2) the analysis of the feature correlations reveals the internal relationship among the features, which supports the better understanding of the data and benefits the application. For instance, with the solid feature correlations found, missing feature values can be estimated through a predictive model using their related features. Therefore, feature correlation analysis is attractive in applications that require anomaly detection with numerous irrelevant features and missing feature values.

2.1.5 Summary

In this section, four diverse strategies, i.e., rule-based, case-based, expectation-based and property-based strategies, are presented for the purpose of anomaly detection. All these strategies have their individual policies for anomaly detection and possess disparate advantages and disadvantages. In distinct applications, the most suitable strategy varies. For general datasets with no evident feature, the rule-based strategy is generally applicable and currently implemented in many industrial systems. The case-based strategy, however, works more efficiently in complex systems where the target datasets of anomaly detection carry various types of features each of which is complicated to analyse. On the other hand, the expectation-based strategy is made straightforward when dealing with sequential datasets and datasets with stable patterns. When it comes to systems that generate data with certain properties, e.g., high-dimensional data with internal feature correlation, the property-based strategy is more advisable. Besides all these strategies, it is believed that more strategies are possible in the area of anomaly detection that further exploration should be consistently made in order to discover strategies with beneficial attributes.

In Section 2.2, the representative techniques for anomaly detection are classified into several categories according to the detailed technical features. It is worth noting that a specific technique could implement various strategies. In other words, strategies are relatively high-level tactics that may overlap with each other in achieving anomaly detection.

2.2 Techniques for Anomaly Detection

Under the umbrella of anomaly detection, the solutions [90] are typically categorised into three aspects according to the type of the input (Section 1.1.3): 1) **supervised** anomaly detection, 2) **unsupervised** anomaly detection and 3) **semi-supervised** anomaly detection. As mentioned earlier, supervised anomaly detection has access to both normal and abnormal data. Therefore, the essence of a supervised anomaly detection problem is a classification problem that endeavors in distinguishing abnormality from normality. On the other hand, unsupervised anomaly detection gains no access to the exact labels of the given dataset. It achieves anomaly detection through identifying the shared patterns among the data instances and observing the outliers. Hence, the task of unsupervised anomaly detection is intensely related to **outlier detection**. In addition, Semi-supervised anomaly detection accepts normal or abnormal dataset and determines the concept of normality or abnormality for anomaly detection. Consequently, semi-supervised anomaly detection is more prone to solving the task of **novelty detection**.

With a detailed examination, it is obvious that the methods for unsupervised/semi-supervised anomaly detection are universally applicable in anomaly detection problems because 1) supervised anomaly detection can be easily divided into two semi-supervised anomaly detection problems, which model the concept of normality and abnormality, respectively; 2) by assuming that a sampled portion of the given dataset is normal or abnormal then examining the remaining data, unsupervised anomaly detection is converted to semi-supervised anomaly detection with some computational expenses. Inversely, semi-supervised anomaly detection is solvable through unsupervised anomaly detection with the aggregation of labeled and unlabeled data. As a result, this section focuses mainly on the review of unsupervised/semi-supervised anomaly detection methods, e.g., one-class classification methods, and does not make a clear distinction between the methods for outlier detection and novelty detection. The specific methods for supervised anomaly detection, i.e., classification-based anomaly detection, can be found in other related surveys [101][3][29]. Moreover, due to the fact that multi-dimensional datasets are more pervasive in applications, this section targets methods with the capability to cope with the high-dimensional datasets and omits the conventional statistical data analysis methods, e.g., 3-sigma [168], boxplot rule [96] and etc.

For the rest of this section, five categories of anomaly detection methods are detailed. Firstly, the underlying assumptions of these methods are described. Detailed methods are then elaborated with their corresponding advantages and disadvantages, which lead to the improvements upon various facets of the methods, such as robustness, effectiveness and etc. Possible research topics and directions are briefly discussed.

2.2.1 Distance-based Methods

Investigating the distances among data instances has been one of the primary approaches to reveal outliers or novelties in anomaly detection. Although the broad spectrum of distance-based anomaly detection methods always vary in the ways they measure the distance and how they calculate the anomaly score, they all share the same following assumption:

> **Assumption:** *Normal data instances are close to their neighbors, while anomalies are far away from their neighbors.*

Within the assumption, 1) the definition of the distance between two data instances and 2) the determination of the term *"close"*, i.e., the ways to assign anomaly score, according to the measurement are essential to anomaly detection.

The Definition of the Distance

Over the years, researchers have proposed numerous methods of distance measurement [49]. Typical measurements include Euclidean distance, i.e.,

$$D_{Euclidean} = \sqrt{\sum_{i=1}^{d} \|x_i - y_i\|^2},$$

and Mahalanobis distance [148], i.e.,

$$D_{Mahalanobis} = \sqrt{(x-y)^T \Sigma^{-1} (x-y)},$$

where x and y are two d-dimensional column vector data instances; x_i and y_i are the i-th elements of x and y, respectively. T denotes the transpose of the column vector. Σ is a matrix that governs the distance measurement.

Although Euclidean distance and Mahalanobis distance are widely applicable in most cases, there are situations where data features are not numerical, which requires novel distance measurement methods. For instance, categorical data ask for different distance measurement methods, such as simple matching, that are more suitable. Interested users are recommended to refer to [19] or [231] for a detailed list of the measurements for numerical, categorical and mixed data. Besides numerical data, categorical data and the mixture of these two, other data types, e.g., image, graph, sequence and etc., may demand more complex distance measurements. Investigations have been conducted individually in corresponding domains [42][114][173].

The Determination of the Anomaly Score

With an appropriate definition of the distance, the distance-based anomaly detection methods leverage the distance information to assign anomaly score for each data instance under examination. A key distinction of these methods is their determination of the anomaly score. One may use the longest distance to the K nearest neighbors of a data instance as the anomaly score, utilise the mean or median of the distances to all the other data as the anomaly score, turn the numerical scores into probabilistic scores, or analyse and summarise all the distances and simply output binary anomaly score for each data instance. Methods, such as [2], **directly** employ the information of distances to assign anomaly scores, while [98] and many other distance-based clustering methods apply the information to firstly form clusters and **indirectly** point out anomalies afterward. An informative survey about the direct and indirect usage of the distance information for anomaly detection is presented in [29].

The direct and indirect utilisations of the information of data distances help categorise distance-based anomaly detection methods into two classes. However, a more helpful taxonomy would be dividing the methods according to the general computational expense used in calculating the distances:

1. **Measure the distance from a data instance to all the other data instances.** Due to the vast number of distance calculations, this always consumes considerable computational resources. Typical examples of this kind involve nearest neighbor-based methods, such as [2][145];

2. **Measure the distance from a data instance to a concise set of targets or data instances.** The size of the concise set is normally several orders of magnitude smaller than that of the original dataset, which enormously reduces the computational expense. A very intuitive example is the K-means clustering method [98].

Example - k-th Nearest Neighbor Anomaly Detection

A traditional distance-based anomaly detection method is to use the distance from a point to its k-th nearest neighbor. For instance, in [188], the definition of an outlier is given based on the distance to the k-th nearest neighbor:

> D_n^k **Outlier:** *"Given an input data set with N points, parameters n and k, a point p is a* D_n^k *outlier if there are no more than* $n-1$ *other points* p' *such that* $D^k(p') > D^k(p)$."[1]

[1] Quoted from [188].

Note that $D^k(p)$ denotes the distance from point p to its k-th nearest neighbor. In other words, the definition picks the top $n-1$ points which have the largest distances to their k-th nearest neighbors. These $n-1$ points are called D_n^k outliers parametrised by n and k. It is worth stressing that, to pinpoint the outliers, calculating the distance from a data instance to all the other data instances are required. This inevitably incurs high computational complexity for the anomaly detection process.

Example - K-means Clustering Anomaly Detection

Due to the high computational complexity of calculating all the distances between pairs of any two data instances, it would be advisable to compute the distances from the data instances to a small set of targets. This strategy is implemented in distance-based clustering methods, such as K-means clustering [98]. In K-means, the targets are the prototypes associated with different clusters. Formally, consider the dataset $X = x_1, x_2, \cdots, x_N$, where $x_i \in \mathbb{R}^d$, $i \in \{1, 2, \cdots, N\}$, d and N are the numbers of data dimension and instances respectively. The prototypes μ_k is for the k-th cluster, where $k \in \{1, 2, \cdots, K\}$ and K is the number of clusters. The intuition behind K-means clustering is to form K clusters which minimise the sum of all the distances from any data instance to the prototype of its assigned cluster:

$$J = \sum_{i=1}^{N} \Big(\min_k ||x_i - \mu_k||^2 \Big).$$

The prototype of a cluster is typically calculated as the mean vector of all the data instances in the cluster, which can be formally represented as:

$$\mu_k = \frac{1}{N} \sum_{i \in C_k} x_i,$$

where C_k is the set of the indexes of data instances that are assigned to cluster k. To find the prototypes that minimise J, data instances are firstly clustered into k sets based on their nearest prototypes and the prototypes are updated accordingly. This process is repeated until convergence. As a result, data instances that are 1) far away from its cluster prototype or 2) in a cluster with very few data instances are considered as anomalous [29].

Distinct from k-th nearest neighbor anomaly detection, K-means clustering demands only the calculations of the distances from a data instance to K cluster prototypes, where $K \ll N$, which largely reduces the computational expense in each iteration of the clustering process. Besides K-means, many other distance-based clustering methods [13][24][219] also enjoy the same merit and further improve K-means from different aspects.

Improvements and Recent Work

The *k*-th nearest neighbor method, K-means anomaly detection and other related methods, e.g., hierarchical clustering [46], are distance-based anomaly detection methods that have been developed for a long time. Over the years, distance-based anomaly detection methods have been researched from three primary aspects: 1) the developments of new definitions of distance-based anomaly score for anomaly detection; 2) the enhancements of diverse distance measurements for various data types; 3) the improvements of the efficiency of distance-based anomaly detection methods.

In 1996, Bradley et. al. [23] had revised K-means and proposed K-medians to support a more robust data clustering and thus more accurate anomaly detection results. Instead of using medians, Zhang and Wang [261] in 2006 employed the sum of the distances from a data instance to all its K nearest neighbors as the anomaly score. More recently, authors of [170] proposed Local Distance-based Outlier Factor (LDOF) to integrate the information of the so-called outer distance and inner distance of a particular data for the definition of the anomaly score. These methods change slightly the original way of using the distances and aim at the enhancements of the robustness and effectiveness of the distance-based anomaly detection methods. While, some other approaches, such as the methods using reverse nearest neighbor [187], provide a brand new perspective of using the distances.

As a critical research topic, distance measurement consistently attracts much attention. In 2002, Xing et. al. [246] proposed distance metric learning to identify a distance measurement that respects the known relationship within a given dataset, i.e., assign small distances between similar data pairs. In the domain of anomaly detection, anomaly metric learning [60] introduced a similar approach to learn a robust Mahalanobis-like distance measurement for anomaly metric. From a different perspective, a metric that measures data dissimilarity was also developed in [217], which demonstrates better results in three existing algorithms of clustering, anomaly detection and multi-label classification. Besides the advancements of the distance measurements for standard data types, similarity measurements concerning complex data types, such as time series [63][114], images [42], graphs [173] and etc., are also under intense developments.

In order to promote the efficiency of distance-based anomaly detection methods, methods such as sampling have been under intense study. In 2006, Wu and Jermaine [233] proposed a sampling algorithm to detect outliers in domains where the distance computation is expensive. [192] also adopted sampling method and theoretically analysed the reason that brings the benefit. Other than sampling, studies concerning the utilisation of distributed computing [22], the detection efficiency of data stream [41] and the combination of optimisation strategies [166] are also hot research directions.

Advantages and Disadvantages of Distance-based Methods

The primary advantages of distance-based anomaly detection methods cover:

- Distance-based anomaly detection methods are **purely data driven** that they make no assumption about the data distribution or generating mechanism and, therefore, are widely applicable.
- Due to the **easy-to-understand concept of distance**, distance-based anomaly detection methods are always more accessible to engineers compared to other complex methods.
- The **adaptation of distance-based anomaly detection methods in diverse application domains** is straightforward because the essence of the methods can stay unchanged and only appropriate distance measures are required for new data types.

Distance-based anomaly detection methods have their disadvantages:

- Distance-based anomaly detection methods **only concern the processing of the distance information**. In datasets, where the anomalies cannot be directly reflected by the distance information, the performances of these methods may be degraded.
- As a practical implementation of the case-based strategy for anomaly detection, distance-based anomaly detection methods typically incur **high computational complexity**, especially for nearest neighbor-based methods.
- In cases where the data are complex, e.g., graphs or a minibatch dataset, **defining a good distance measurement between two instances can be really challenging**, not to mention the measurement should effectively reflect the distinction between normal and abnormal instances.

2.2.2 Density-based Methods

In distance-based anomaly detection, anomalies are determined solely according to the distance information extracted from the target dataset. Although helpful in many applications, using distance information has some potential problems, a primary one of which is that it does not take into consideration the relative position or the distribution of the neighbors of a data instance. In many cases, simply adopting the distance information for anomaly detection is not enough. To promote the methods, the density information is always employed to integrate more information. The density information implicitly contains the distance information and is expected to support better anomaly detection results.

Density-based anomaly detection methods make a slightly different assumption compared to distance-based anomaly detection methods:

> **Assumption:** *Normal data instances form dense areas, while anomalies always appear in sparse areas.*

Intuitively, whether an area is dense or sparse is determined by both the size of the area, which is highly related to the distance information, and the number of the data instances within the area. This is a straightforward way of modeling the density information. Another set of methods for extracting density information is distribution-based methods which fit a distribution to the dataset to implicitly model the density information. Both of these types of methods are under the broad umbrella of statistical analysis. More formally, from the standpoint of statistics, these methods are classified into two categories: 1) parametric techniques; and 2) non-parametric techniques.

Parametric Techniques

To extract the density information, a traditional statistical analysis method is to fit an assumed parametric distribution D to the dataset. After training, the probability density function $f(x, \theta)$ of the distribution, where x is an observation and θ is the trained parameters, is expected to reflect the density information of the target dataset for anomaly detection. As can be noticed, the key components of the parametric techniques include: 1) the assumed distribution, 2) the training of the probability density function and 3) the assignment of the anomaly scores and threshold for anomaly detection.

For many existing research and applications, Gaussian distribution is always the first candidate for the distribution assumption. Although there emerge many other diverse distributions, e.g., Student's t-distribution [133], Gaussian distribution is still the most popular one under research and utilisation. Considering the usage of a single Gaussian distribution for data analysis, the training of the distribution typically relies on the Maximum Likelihood Estimation (MLE) method [1] which identify the best parameters to maximise the overall probability of generating the entire target dataset. The anomaly scores of the data instances can be assigned as their corresponding data probabilities under the trained distribution. The threshold, however, is determined according to different application scenarios. In many cases, a single distribution cannot properly model the dataset which is generated by a mixture of models. As a result, the mixture of parametric distributions becomes a natural countermeasure [14][183]. A concrete example based on GMM is presented in next section to demonstrate the essentials of the parametric techniques.

Example - Gaussian Mixture Model

The GMM, i.e., Gaussian Mixture Model, is essentially a linear superposition of Gaussians [14]. Its probability density function can be formally represented as:

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k),$$

where $k \in \{1, 2, \cdots, K\}$; K is the number of Gaussian distribution $\mathcal{N}$; π_k specifies the probabilities that the data x belongs to the k-th Gaussian distribution and has $\sum_{k=1}^{K} \pi_k = 1$; μ_k and Σ_k are the mean and covariance matrix of the k-th Gaussian distribution, respectively. To fit the model to the training dataset, the idea of MLE is also adopted, which maximises the following target function:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\},$$

in which X is the set of N data instances, π the set of π_k, μ the set of μ_k, and Σ represents the set of Σ_k. This model, parameterised by π, μ, Σ, is trained with the help of Expectation-Maximisation algorithm (EM algorithm) [55]. After training, the model is employed as a scoring function for anomalies. Data instances with low probability are therefore considered as anomalous.

Non-Parametric Techniques

Rather than modeling the density information with certain distributions, non-parametric density-based anomaly detection methods directly measure the data density. As a concrete example, the Histogram method [14] segments the data space and counts the number of data instances located inside specific segments for density estimation. As a result, similar techniques are also referred to as frequent-based or counting-based methods. The histogram is a statistical method that dates back to very early years. Recent methods that are applicable in density estimation largely focus on Kernel Density Estimation (KDE) methods [167][132][204]. Both of these types of methods are designed to achieve density estimation. Therefore, their applications in anomaly detection demand another further step which is to leverage the density information to assign anomaly scores for data instances. Besides these, other related methods also cover Local Outlier Factor (LOF) [21], Connectivity-based Outlier Factor (COF) [206] and clustering methods, e.g., DBSCAN [65]. All these methods exploit the density information and are further formulated to achieve anomaly detection.

Example - Parzen Density Estimation

A classic method for density estimation of multivariate datasets is the Parzen Density Estimation (PDE) [167] which does not make any assumption concerning the distribution or shape of the target dataset. In PDE, the density information is directly estimated according to the number of data instances situated in a specific region of a predefined volume. For each space point or data instance x, the formulation of its density information is given as:

$$p(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{V}\phi(\frac{x-x_i}{h}),$$

where N is the size of the target/training dataset, $V = h^d$ is the volume of the d-dimensional hypercube whose edge is of length h. Most importantly, ϕ is a kernel function that satisfies $\phi(v) > 0$ and $\int \phi(v)dv = 1$. In PDE, the kernel function is defined as:

$$\phi(v) = \begin{cases} 1, \text{ if } \forall j = 1,2,\cdots,d, \ |v_j| < 1/2, \\ 0, \text{ otherwise}, \end{cases}$$

which decides whether the data instance x_i is located within the hypercube centred in x. Therefore, PDE finds the average percentage of the number of data instances in a hypercube for the density estimation. The density information can thus be employed as the anomaly score for anomaly detection.

Improvements and Recent Work

Over the years, density-based anomaly detection methods have been under intense investigation and experienced fast development. Both parametric and non-parametric methods are practically utilised in many different application domains to solve real problems. For parametric methods, the representative, i.e., GMM, has been employed in diverse areas such as flight operation monitoring [129], crowd behavior anomaly detection [153] and anomaly detection of hyperspectral images [64]. For nonparametric methods, e.g., LOF, i.e., Local Outlier Factor, KDE, i.e., Kernel Density Estimation, and DBSCAN, numerous applications and developments have been made.

Concerning LOF, it has been utilised in various areas along with other techniques for better effectiveness. Areas, such as process control [160][200], network intrusion detection [225] and traffic data outlier detection [155], all witnessed the positive effects of using LOF. Other revisions of LOF exist and always rely on the underlying applications.

To promote the accuracy of KDE, Lichman and Smyth [139] introduced the mixture of kernel densities to integrate information from a different perspective, which results in the enhanced accuracy of pattern recognition. In 2017, Zhang et. al. [257] proposed adaptive kernel density-based anomaly detection which assigns adaptive width for the kernel so as to improve the overall accuracy of the method in nonlinear systems. Moreover, the robustness of KDE is investigated in [226] and the applicability and flexibility of KDE are also explored and extended through a general framework for outlier detection in [204].

Similarly, for DBSCAN, a number of researchers have analysed the method [85][199] especially from the perspective of its efficiency. Other research concerning boosting the process of DBSCAN also include distributed computing, which utilises MapReduce [97] to promote the speed of DBSCAN, and fast neighbor searching, which accelerates the neighbor searching process for better efficiency [118]. Aside from the improvements of the efficiency, many other works also aim at revising DBSCAN to handle datasets with irregular characteristics [212] and inventing robust methods for the process of parameter selection in DBSCAN [112].

To summarise, the density-based anomaly detection methods are under fast development. Revisions to existing methods and developing novel density-based methods both attract a large amount of attention [241][218]. The emergence of more advanced methods is highly possible in this domain.

Advantages and Disadvantages of Density-based Methods

Due to the fact that the density information implicitly utilises the distance information, the density-based anomaly detection methods share many similar advantages and disadvantages with the distance-based anomaly detection methods. The primary advantages of density-based anomaly detection methods cover:

- **Non-parametric** density-based anomaly detection methods are **purely data driven** that they make no assumption about the data distribution or generating mechanism and therefore are widely applicable. And **parametric** methods are typically more efficient and provide **statistically justifiable** solutions for anomaly detection.
- Due to the **easy-to-understand concept of density**, density-based anomaly detection methods are always more accessible to engineers compared to other complex methods.
- The utilisation of density information or relative density information **supports the anomaly detection of more complex datasets** in which only the distance information is not sufficient to reflect the anomalies.

Density-based anomaly detection methods are not without their disadvantages:

- Non-parametric density-based anomaly detection methods somehow leverage the distance information and therefore have the similar problem of **high computational complexity** with distance-based anomaly detection methods.
- Parametric density-based anomaly detection methods **rely heavily on their assumptions of the underlying data distributions**. If the assumptions do not hold, which is often the case, the methods are likely to provide results that are not satisfactory.

2.2.3 Boundary-based Methods

In distance-based and density-based anomaly detection, they exploit the distance and density information to build anomaly detectors that recognise data instances far away from other instances or located in sparse areas respectively. Differently, boundary-based anomaly detection seeks a boundary that surrounds normal data instances with the expectation that the surrounded area contains only normal data. To put it more formally, boundary-based anomaly detection methods make the following assumption:

> **Assumption:** *Normal data instances are located in the normal region/regions which is/are defined by a boundary/boundaries, while anomalies are located outside the region/regions.*

Consequently, the key problem in boundary-based anomaly detection methods is how to discover the best way to define the boundary. In the situations where no label is provided for the training process of anomaly detection, i.e., unsupervised and semi-supervised scenarios, it is advisable to assume that most of the data instances in the training dataset are normal. Therefore, the training process of boundary-based anomaly detection is to utilise the training dataset to find the boundary, while the testing phase further exploits the boundary for anomaly detection. Diverse ways of determining the boundary promote different methods that have distinct features.

The most related methods cover 1) K-Centres [186], which finds K hyperspheres with minimum volumes to embrace all the data instances, 2) Elliptic Data Description [185], which utilises hyperellipsoid to encompass the dataset, 3) One-class Support Vector Machine [201], which identifies the hyperplane that has the largest distance to the origin and at the same time isolates the dataset from the origin, 4) Support Vector Data Description [208], which tightly surrounds the dataset with a hypersphere, and 5) Level Set Method [59], which directly constructs the so-called level set as the boundary function.

Example - Support Vector Data Description

In [208], Tax and Duin developed Support Vector Data Description (SVDD) as a method to achieve one-class classification. The basic idea behind the method is to seek a hypersphere with minimum volume that can encompass all the training data instances which are considered as normal data. This initial idea (with no slack variables) is formally presented as an optimisation problem:

$$\begin{aligned} &\min_{a,R} \ R^2 \\ &\text{s.t.} \ \ \|\phi(x_i) - a\|^2 \leq R^2, \ \ \forall i \in \{1,2,\cdots,N\}, \end{aligned}$$

where a, R, ϕ and N represents the centre, radius of the hypersphere, the mapping function for kernel and the number of training data instances respectively. To solve the optimisation problem, the original formulation is converted to its dual formulation which can be successfully solved by traditional Quadratic Programming (QP) solutions. The boundary function in SVDD is defined as:

$$F(z) = \|\phi(z) - a\|^2 - R^2,$$

which is made concrete with the results obtained through solving the QP problem as:

$$F(z) = \left(\Phi(z,z) - 2\sum_i \alpha_i \Phi(z,x_i) + \sum_{i,j} \alpha_i \alpha_j \Phi(x_i,x_j) \right) - R^2.$$

Note that z is a new data vector, α_i is the Lagrangian multiplier corresponding to x_i, $\Phi(\cdot,\cdot)$ is the kernel function related to the mapping function and R^2 is obtained through setting $F(z) = 0$ with z replaced with a data instance whose Lagrangian multiplier is positive. Any new data instance z that satisfies the equation $F(z) = 0$ is on the boundary. Data instance z which have $F(z) < 0$ is inside the boundary. On the other hand, if $F(z) > 0$, z is identified as an anomaly.

Improvements and Recent Work

Besides the aforementioned methods, boundary-based anomaly detection methods also include ISODEPTH [189], Convex Peeling and etc. However, many of these methods have their own defects and have been outdated since the thriving advancement of machine learning methods. The boundary-based methods that still attract much attention are one-class SVM [201] and SVDD [209][208][20], which are considered two critical branches for anomaly detection derived from the prestigious classification method SVM [17].

One-class SVM and SVDD have been proved identical under the utilisation of specific kernels, e.g., Gaussian kernel. Both of them are sparse kernel models which have sparse solutions so that maintaining a subset of the dataset, i.e., Support Vectors, for model training is sufficient for model testing. They both maintain high effectiveness and high accuracy. Due to its easy-to-understand geometrical interpretation, SVDD is heavily investigated over the last decade. SVDD has fruitful research outcomes that spread through four main research directions: 1) theoretical improvement and discovery; improving the 2) accuracy, 3) efficiency of SVDD; and exploring its 4) applications in different problems and areas.

Originally, SVDD leads to an underlying QP problem. Over the last decade, numerous research has helped better understand the problem. In [35], Chang et al. provided a thorough analysis of the QP problem and discussed the feasibility of the solution under different settings. In [31], Campbell and Bennett devised a Linear Programming (LP) SVDD instead of QP. A unified model of SVM and SVDD was proposed in [143]. Besides the developments of SVDD in terms of the basic theory, many of the previous research works focus on the developing of data description under the presence of noise or imbalance data [45][78][239]. Developing SVDD under probabilistic input is another important research direction, some works of which include Fuzzy SVDD [142] or the like [71]. In addition, Elliptical SVDD [215][238] and Multi-sphere SVDD [248][142] are investigated to enhance the accuracy in describing a dataset with a specific shape. In [128][161][210], the ideas for incremental SVDD had been fulfilled to meet the requirements of the constant processing time for each incoming data and the constant/low memory utilisation with preserved accuracy. From the perspective of applications, SVDD has been extended for clustering [20] and binary/multi-class classification problems [154]. The applications of SVDD in novelty detection and outlier detection have also led to research efforts in different domains, such as image processing [18], process control [159], machinery fault diagnosis [58], wireless sensor networks [196] and many others [198][237]. A more detailed review of SVDD related methods could be found in [99].

Advantages and Disadvantages of Boundary-based Methods

The primary advantages of boundary-based anomaly detection methods include:

- The concept of the boundary has a clear **geometric interpretation** which helps with the understanding of related methods.
- A boundary-based anomaly detection method typically maintains a **clear and concise decision function** for the determination of anomalies that the decision-making/testing process could be relatively faster compared to distance-based or density-based methods.

Boundary-based anomaly detection methods are not without their disadvantages:

- Boundary-based anomaly detection **does not take into consideration the information of data distance, density and etc.** Therefore, additional mechanisms are demanded to integrate extra information for advanced anomaly detection.
- Boundary-based anomaly detection methods are **sensitive to the parameters** whose adjustments may cause a dramatic change in the performance of anomaly detection.

2.2.4 Partition-based Methods

Distinct from boundary-based methods, which expect to find a specialised space for normal data, partition-based methods slice the data space for anomaly detection. It could be summarised that partition-based anomaly detection methods possess the following assumption:

> **Assumption:** *Normal data occur in specific areas of the data space, while anomalies are not within these areas.*

Consequently, the approaches to slice the space and determine the normal areas become the prominent sub-tasks of anomaly detection and have led to distinct partition-based methods. Generally, there are three ways to slice the space: 1) slicing the space into hypercubes according to the original coordinate; 2) partitioning the space with respect to certain dimensions; 3) using certain hyperplanes to isolate the space into subspaces. Corresponding to the three ways of slicing the space, Binary Decision Diagram (BDD) [108], Isolation Forest (iForest) [144] and Randomised Hashing [191] are three partition-based methods that were developed recently and have attracted much attention due to their superior effectiveness and efficiency in solving anomaly detection problems. In what follows, iForest is presented as an exceptional candidate to demonstrate the idea of partition-based anomaly detection.

Example - Isolation Forest

The idea of iForest, i.e., Isolation Forest, for anomaly detection arises from the observation that anomalies in a given dataset can always be easily isolated from other normal data instances. Therefore, to identify an anomaly, iteratively random divisions of the input space can be performed to construct division trees, i.e., binary trees, to isolate all the data instances. In a division tree, it is expected that a normal data instance will be located on the bottom of the tree, which shows it takes many divisions to isolate the data instance from others, i.e., normal. On the other hand, an anomaly would typically appear close to the root of the tree, which means that it is isolated with few divisions and, thus, is very different from other data instances, i.e., anomalous.

To be more specific, iForest starts with sampling sub-datasets from the original dataset. For each sample, it randomly selects a feature of the sub-dataset and randomly selects a value for the feature to form a node of a binary tree called iTree. The left child and the right child of the node are determined afterward. Thus, the whole tree is constructed iteratively till all the leaf nodes, i.e., nodes that have no children, represent a space containing only one data instance in the sub-dataset. The anomaly score for a data instance is measured according to the average length of the traversing paths in all iTrees when searching for the data.

As the process of iForest indicates, the anomaly detection in iForest doesn't require any explicit similarity measurements, e.g., distance measurement and density measurement, among data instances. This property significantly simplifies the process of anomaly detection and contributes greatly to its high efficiency. The distributed nature of iTrees helps with enabling distributed anomaly detection, which further lowers the time consumption of the method. It is also reported in [144] that iForest outperforms many other methods in terms of accuracy and robustness. However, a potential issue of iForest may arise from the size of the iTrees when the size of the given dataset is immense, and keeping updating the iTrees with incoming new data instances is another important research direction that hasn't been largely explored.

Improvements and Recent Work

Partition-based anomaly detection methods have become hot research topics in the past few years. The methods that attract much attention are iForest [144] and its related methods, BDD [108] and Hashing-based methods [191], especially methods using Local Sensitive Hashing (LSH) [234]. Among these methods, iForest has aroused the most research efforts. From the perspective of its applications, it has been utilised or modified to solve practical problems in areas such as the anomaly detection of traffic trajectories [44], time series in cloud data centers [40], building energy-consumption [116] and data retrieved from wireless sensor networks [50]. It is also witnessed that ideas similar to iForest can not only solve anomaly detection problems of general datasets but also be applicable to that of data streams. [83] proposed to use random cut forests to achieve anomaly detection on data streams, and Half-Space-Trees [216] employed tree-based space partition to cope with the problem of efficient anomaly detection in streaming data. Besides anomaly detection related problems, the idea of tree-based space partition is also extended to solve complex classification problems with advantageous properties, e.g., classifying emerging new classes [162]. In [241], the idea is exploited to construct a density estimator that succeeds in anomaly detection in data streams, which broaden the capability of the idea from a distinct angle. Additionally, in 2017, [254] introduces a generic framework to integrate iForest with LSH.

Similar to iForest, the idea of hashing, especially LSH [234], also incites lots of research interests. LSH is a type of hashing that maintains the similarity among data instances with the capability of dimension reduction [194]. Consequently, LSH is preferred by applications in which similarities among data instances of high dimensionality need to be measured. LSH related methods have also been developed to solve anomaly detection problems under distributed settings [178] to further boost the efficiency. In order to improve the accuracy, additional research has utilised the information from the training dataset for guiding the selection of the hash functions [95]. In 2016, Zhang et. al. [260] successfully applied LSH in video anomaly detection, which demonstrates its applicability in more complex datasets. Compared to iForest and LSH, BDD experienced slightly slow development. Since the publication of the method in 2010 [108], two primary works have been done to improve the applicability of BDD. Firstly, BDD is employed as a critical tool to support a parameter-free one-class classifier [109] which is both efficient and accurate. Secondly, BDD is utilised to evaluate the so-called leave-one-out density [110][111] of a datum for anomaly detection. The time complexity of the method is linear with respect to the size of the dataset.

Advantages and Disadvantages of Partition-based Methods

The primary advantages of partition-based anomaly detection methods include:

- Compare to other methods of anomaly detection, partition-based methods are known to be very **efficient** due to the fact that they do not explicitly rely on the data distance for anomaly detection.
- Partition-based methods typically partition the data space into diverse parts or divide the dataset into several portions, which promotes the **distributed processing** of anomaly detection.
- Partition-based methods, e.g., LSH, which are **capable of dimensionality reduction**, are potent in processing high dimensional datasets.

Partition-based anomaly detection methods are not without their disadvantages:

- For some of the partition-based methods, **anomaly detection over high-dimensional data can be troublesome** due to the fact that a large number of dimensions will contribute significantly to the variance of partitioning the data space, hence undermining the accuracy of anomaly detection.
- Partition-based anomaly detection normally slices the data space with hyperplanes, which **always generates unsmooth decision boundaries**. This immensely influences the anomaly detection results especially when the number of hyperplanes is limited.

2.2.5 Property-based Methods

In distance-based, density-based, boundary-based and partition-based methods, the basic strategy is to model the apparent characteristics of the target dataset in order to identify the data instances that disobey the characteristics. However, in property-based methods, it is presumed that the target dataset is generated by an underlying system with certain stable properties. Therefore, property-based methods are modeling the intrinsic mechanism, e.g., principal components, data dictionary, data components, data correlations, latent function or latent information, that is essential to generating the target datasets. Consequently, the assumption made by property-based methods are as follows:

> **Assumption:** *The set of normal data instances is generated according to a certain mechanism, while the anomalies are not.*

Depending on the type of property/mechanism monitored, anomaly detection methods vary a lot. In this section, six different types of methods are presented:

1) ***Component Analysis***: with the assumption that the target dataset is generated by combining normal data, data noise and data error, component analysis wishes to *decompose the given dataset* for anomaly detection.

2) ***Subspace Analysis***: subspace analysis holds the view that the target dataset is produced according to some predefined subspaces so that *the data that do not align well with others in the subspaces* are anomalous.

3) ***Representation Analysis***: in the systems where normal data instances are generated based on a dictionary or representatives, representation analysis strives to *find these key data instances and at the same time identify anomalies*.

4) ***Latent Information Analysis***: latent information analysis holds the assumption that *the system generates data instances with similar latent information*. Therefore, the data with different latent information are marked as anomalies.

5) ***Latent Function Analysis***: latent function analysis assumes that *the normal dataset is generated according to a set of underlying functions*. Data that are not generated by the functions are anomalies.

6) ***Correlation Analysis***: when a system generates a data instance with stable feature correlations, this property can be revealed by correlation analysis. *Anomalies are pinpointed if the data instances do not maintain the feature correlation.*

In the following subsections, these types of methods will be discussed in details concerning their technical methodologies and existing works. Each of these types has actually become a research direction for anomaly detection and some of them, e.g., correlation analysis, have been implemented in anomaly detection systems that aim at resolving high-level anomaly detection problems, such as network device fault detection [81][152].

Component Analysis

Reported as a sub-domain of spectral anomaly detection in [29], the component analysis is a critical research direction in which matrix decomposition or factorisation plays a significant role. Methods aiming at matrix component analysis, e.g., Robust Principal Component Analysis (RPCA) [36][258][245] and other matrix decomposition methods [37][151][113], have witnessed increasing number of applications for anomaly detection in recent years.

The key idea in matrix decomposition for anomaly detection is to filter the outliers in the data matrix so that the revised matrix satisfies certain criteria, e.g., low-rank. The criteria is always assumed according to the actual application. As a quick example, the introduction of the Direct Robust Matrix Factorisation (DRMF) [244] in anomaly detection is presented here. The central idea of DRMF is to find a low-rank approximation matrix L of the given matrix X with the possible outliers S excluded:

$$\begin{aligned} \min_{L,S} \quad & \|X-L-S\|_F \\ \text{s.t.} \quad & \text{rank}(L) \leq K \\ & \|S\|_0 \leq e. \end{aligned}$$

Here, $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$ is the Frobenius norm, while $\|X\|_0 = \sum_{ij} I(X_{ij} \neq 0)$ is the $L0$-norm; rank(L) represents the matrix rank of L; K and e are the upper bounds for matrix rank and the number of outliers, respectively. With the solution of the formulation, the original matrix X is decomposed into two components, i.e., the low-rank approximation matrix L and the outlier matrix S. This outlier matrix S is the primary result of anomaly detection using DRMF.

Typically, other matrix decomposition methods [255][256] use methodologies similar to DRMF. Differences are witnessed in how they formulate and solve the core optimisation problem of matrix decomposition. In general, matrix decomposition is efficient due to the availability of theoretical methods and practical hardware, e.g., Graphics Processing Units (GPU), for boosting the process of matrix calculations. It is particularly suitable for offline processing of a large number of data instances.

Nevertheless, a major drawback of matrix decomposition is that the assumptions made by the methods largely influence their performances. The performances of the methods are very sensitive to the changes of the criterion, e.g., the parameters K and e in DRMF. Furthermore, it is more an offline technique that works with static matrices. To achieve online anomaly detection, more challenging tasks, e.g., incremental matrix factorisation, await further exploration.

Subspace Analysis

Besides component analysis, another critical sub-domain of spectral anomaly detection is subspace analysis. Subspace analysis emphasises the significance of embedding the data into a lower dimensional subspace in order to identify anomalies that are not apparent in the original space. Consequently, determining such subspaces is the key in subspace analysis. A major method to determine the subspace that captures the bulk of variability in the data is the Principal Component Analysis (PCA) method [102]. The method targets at pinpointing the principal components, which capture most of the data information, of a given dataset and measures the abnormality of a data instance according to its reconstruction error using only the principal components [207]. Although the method is relatively old, it is capable of dimensionality reduction and is fast to compute. Therefore, there still are many works focusing on its development and deployment. Kernel PCA [202] is a critical improvement over the original design. The method introduces a non-linear mapping function to transfer the data from the original space to the kernel space in which the original PCA is performed to find the principal components. The dot product of two mapping functions gives rise to the utilisation of the so-called kernel function and makes the implicit utilisation of the mapping function possible. The result of Kernel PCA is the identification of the curved principal components which are much more accurate compared to original PCA. Thus, Kernel PCA empowers related anomaly detection methods, e.g., [89], with improved accuracy and succeeds in better dimensionality reduction results. Besides Kernel PCA, other research covers further theoretical analysis of PCA, e.g., applicability analysis [57], extending PCA for different anomaly detection scenarios, e.g., change detection in data streams [180][53], applying PCA in diverse application domains, e.g., social network [222], and etc.

Generally speaking, subspace analysis methods for anomaly detection are skilled in processing high-dimensional datasets due to their intrinsic capability of dimensionality reduction. They are beneficial to understanding the structure of the target dataset so that making data analysis tasks easier. However, as pointed out by [57], using subspace analysis methods, e.g., PCA, for anomaly detection has to be careful due to the conditions of detectability of the anomalies, which means the methods have certain limitations.

Representation Analysis

In boundary-based anomaly detection, a superior property is witnessed in methods, e.g., SVDD, that the original dataset can be abstracted using a set of representatives, e.g., Support Vectors, whose number is typically several magnitudes smaller than the size of the dataset. To emphasise the significance of the property in processing datasets with large volume, representation analysis methods for anomaly detection are proposed. These methods typically assume that normal data instances can be easily represented by a set of data representatives, while anomalies are hard to be expressed. Hence, there are two primary tasks/techniques in a representation analysis method: 1) the identification of the representatives and 2) the expression/representation of data instances by the representatives. Generally, these two tasks are tightly bound and have similar procedures. In this section, related methods are categorised according to how they achieve data expression into two classes: sparse coding methods and sparse representative methods.

Sparse Coding: The first type of representation analysis is to identify the representatives under the definition of a coding mechanism. For instance, a linearly independent set of data can be the representatives of all the data which can be presented by a linear combination of the representatives. Here in the example, the coding mechanism is the linear combination of representatives. Related methods of this type have been largely explored in recent years. Standout coding methods include dictionary learning [48], Nonnegative Matrix Factorisation (NMF) [140][140][240], subspace segmentation [135], sparse modeling [66][68] and etc. Some of them have been thoroughly investigated for the purpose of anomaly detection, such as [48][7][86].

As one may notice, one possible benefit brought by sparse coding is dimensionality reduction. When the number of representatives is smaller than the dimension of the data, the codes to represent data are of lower dimensions. Therefore, sparse coding is particularly interesting for high-dimensional data process problems. Nonetheless, in the cases where the number of representatives is larger than the dimension, sparse coding may not be the best choice for anomaly detection.

Sparse Representative: The second type of representative analysis is more direct. It achieves the recognition of representatives through clustering or even sampling. For example, a rough estimation of the representatives is a uniformly sampling of the original dataset. More advanced methods for representative identification are clustering-based methods, such as Self-organising Map (SOM) [107]. In SOM, the neural network succeeds in capturing the overall structure of the dataset after training while the neurons are distributed according to the density distribution. As a result, the neural network works like a skeleton of the dataset and the neurons act as the representatives.

SOM has been utilised in many applications related to anomaly detection [52][51]. Its most attractive merits are in two folds: on one hand, SOM is an unsupervised method that requires no label for anomaly detection; on the other hand, it achieves data reduction that boosts the efficiency of anomaly detection. The set of neurons in SOM works as a superb set of samples for representing the original dataset, which benefits a large number of applications. Nevertheless, most of the sparse representative methods, such as SOM, is not originally designed for anomaly detection. Therefore, they need to be further optimised for the purpose.

Latent Information Analysis

In the above-mentioned three analysis approaches, the anomaly detection methods tend to identify explicit information, e.g., data compositions, principal components and representatives, for pinpointing anomalies. In some other methods, implicit information is exploited. A classical example of methods using implicit information is the RNN, i.e., Replicator Neural Network [94]. In the initial design of RNN, a fully connected neural network with only one hidden layer is utilised to fulfill the replication of the input dataset, i.e., the input and output are the same during the training process. It is worth stressing that the size of the hidden layer is smaller than that of the input layer in RNN. Therefore, RNN is essentially a combination of a compressor/encoder and a decompressor/decoder. The compressor encodes the input data using only the key information, while the decompressor strives in recovering the input data using only the key information. It is expected that RNN learns the common way to extract the valuable information of the training data, and any datum whose information cannot be extracted through the same procedure is considered as anomalies.

Besides RNN, similar approaches cover compression-based methods [114], Adaptive Resonance Theory (ART) [34] and etc. Compression-based methods leverage latent information to compress similar data, while ART stores latent information in the neural network for further utilisation. Originally, these methods are applied in the fields other than anomaly detection, such as time series similarity measurement and general clustering problems. As similarity measurement and clustering are both key problems in specific anomaly detection methods, e.g., distance-based anomaly detection, compression-based methods and ART are candidates to fulfill novel anomaly detection methods with desirable features, e.g. efficiency and self-learning. For example, in 1996, Arning and Agrawal [5] proposed a technique called sequential exception based on the dissimilarity within the dataset for anomaly detection. The algorithm requires a function to measure the so-called implicit redundancy, i.e., which elements in a data set cause the dissimilarity of the data set to increase. The implicit redundancy techniques could implement the compression-based methods to assess the homogeneousness of the dataset after the removal of an object or a set of objects.

Ideas concerning the utilisation of latent information are currently under intense investigation. Angle-based anomaly detection methods, such as [120][177], compression-based methods and neural network-based methods are all possible directions to further extend the methodologies of anomaly detection. From the perspective of efficiency, methods utilising latent information are always superior especially during the testing phase. And methods, such as compression-based anomaly detection, are directly applicable to all kinds of datasets regardless of their size, format and context. Nonetheless, due to the implicit content of information, one is hard to mathematically explain the essential theories behind the methods, such as neural network-based methods, for anomaly detection.

Latent Function Analysis

Latent function analysis models anomaly detection as a one-class classification problem and assumes that there exists a one-class classification function that is capable of identifying anomalies, i.e., $f(x) = y$, where x is a data instance and $y \in \{0,1\}$ is the label. Therefore, the keys in the latent function analysis are the formulation of the function and the way to train the function under the one-class setting. GP for one-class classification [119] and the least squares approach [182] are two excellent examples of latent function analysis.

GP has long been famous as an approach for regression and classification [236]. Nevertheless, rare work has been reported concerning the utilisation of GP in anomaly detection. In 2013, Kemmler et. al. [119] investigated the task of using GP to achieve one-class classification and discussed theoretical connections between GP and other methods. Essentially, GP relaxes the assumption that a given dataset is generated by a certain parametric family of functions and proposes that a specific probability distribution of functions gives rise to the generation of the given dataset, where the function values are assumed to follow a Gaussian process. With a further assumption of Gaussian white noise between a real value and its corresponding function value, the predictive value for a new data instances can be estimated using another Gaussian distribution denoted here by $p(y_*|X,Y,x_*) = \mathcal{N}(y_*|\mu_*, \sigma_*^2)$, where X and Y are the feature values and labels of the training dataset respectively. x_* and y_* are the feature value and label of the target dataset respectively. μ_* and σ_*^2 are the mean and variance of y_* respectively corresponding to the feature value x_*. For one-class classification, a zero mean of Gaussian process prior is used, while labels of all training data are set to 1. Consequently, the values of the latent function are close to 1 in areas near to training data and 0 otherwise. Four criterion are suggested as scores for anomaly detection: 1) the mean of the predicted value μ_*, 2) the negative variance value $-\sigma_*^2$, 3) the probability $\mathcal{N}(y_*|\mu_*, \sigma_*^2)$ and 4) a heuristic score $\frac{\mu_*}{\sigma_*}$.

In GP, $p(y_*|X,Y,x_*) = \mathcal{N}(y_*|\mu_*,\sigma_*^2)$ is the latent function, while a simpler function is adopted in the least-squares approach for anomaly detection [182]:

$$f(x) = p(y|x,\theta) = 1 - \theta^T \phi(x),$$

where $\phi(\cdot)$ is a mapping funtion and θ is the parameter. To achieve one-class classification, the latent function is trained to be close to 1 in areas where the training data are populated and 0 otherwise. The process for training the one-class classifier is designed as the following minimisation procedure:

$$\min_{\theta} \ J(\theta) = \frac{1}{2}\int \left(1 - \theta^T \phi(x)\right)^2 p(x)\mathrm{d}x + \frac{\rho}{2}\|\theta\|^2.$$

Here, $p(x)$ is the probability distribution of data x and ρ is a parameter for balancing the regularisation term $\frac{\rho}{2}\|\theta\|^2$. Through empirical approximating of $p(x)$, the minimisation problem gives the optimal parameter θ_* which is further utilised by the latent function for one-class classification.

Although elegant, a critical potential problem that may hinder the utilisation of latent function analysis in practical applications is the assumed functions in related methods, e.g., GP presumes that the function values follow the Gaussian process. Despite this, latent function analysis methods have their own merits that are useful in practical anomaly detection problems. For instance, GP is skilled in incremental data analysis and the least-squares approach is efficient in both training and testing phase of anomaly detection. Latent function analysis also poses potential research direction that classification and regression methods are all possible to be modified and applied to support one-class classification.

Correlation Analysis

Correlation analysis is mostly mentioned and utilised in statistical analysis problems. The relations among diverse variables support the solution of various problems, such as anomaly detection. From the perspective of general datasets, the idea of correlation analysis is promoted for outlier detection problems in [175]. Unsupervised anomaly detection is generally harder than supervised anomaly detection due to the lack of label information in the training dataset. Through feature correlation analysis, Paulheim and Meusel reformulated the unsupervised anomaly detection problem as a set of supervised regression problems. To be more specific, the set of regression problems for a d-dimensional numerical data $x \in \mathbb{R}^d$ is designed as:

$$\begin{aligned} \text{regress}_1(x_2, x_3, \cdots, x_d) &\sim x_1 \\ \text{regress}_2(x_1, x_3, \cdots, x_d) &\sim x_2 \\ &\cdots \\ \text{regress}_d(x_1, x_2, \cdots, x_{d-1}) &\sim x_d \end{aligned}$$

where x_i, $i \in \{1, 2, \cdots, d\}$ is the i-th feature in the data. With the further assignment of the weights to the regression problems, the overall accuracy of anomaly detection is enhanced. Also, as a side product, the learned weights pinpoint the irrelevant features of the data, which succeeds in feature selection.

Apart from general datasets, the idea of correlation analysis has also been adopted in large-scale systems for system anomaly detection [103][8]. In recent years, the notion of invariant network has attracted large number of research interests and resulted in fruitful research outcomes [152][181][211][81][43]. Essentially, the idea of the invariant network is to identify the invariant connections/relations among the variables found in the system. A variable could be the metric of the CPU utilisation in a device or the measurement of the temperature in an area of the system. A stable correlation between two variables along the time indicates the good health of the components related to the variables. Related information about the dynamics of the correlations also provides indications for the fault localisation of the system, which has tremendous potential benefits for the management of the target system.

The idea and related techniques have been investigated in practical systems, such as wireless sensor networks [152], and it is expected that the correlation analysis methodology will contribute even more to the domain of anomaly detection and further system-level fault detection/localisation. Nonetheless, the method is not without its drawbacks. The large computational complexity, e.g., the augmented number of regression problems, is the very first issue to be addressed to promote the applicability of the method in broader application domains. The calculation of the correlations in special types of datasets, e.g., time-series dataset and textual datasets, is another practical problem to be tackled. Most importantly, the incremental measurement of the correlations should be well examined to make the methods practical in coping with problems posed by evolving datasets and systems.

2.2.6 Discussion and Other Methods

In the previous subsection, five major types of anomaly detection methods are elaborated. All of these methods have their individual properties. To determine which method to take under a specific scenario typically requires detailed examination of the critical demands in a application. Currently, there is no a well-rounded method that is suitable for all applications.

Hybrid Methods

Due to the fact that there is no best method in all applications, numerous works have been investigating the possibility of combining different methods for anomaly detection. The resulting method, always called a hybrid method, is expected to possess better accuracy and beneficial features. Early investigations of the idea of hybrid methods have spread through various domains, such as time-series prediction [174], classification [122][121] and etc. The research of hybrid methods in anomaly detection domain is also under fast development and has produced many successful applications and research outcomes [164][253][190][259]. For example, [190] presents a system with two stages in which a clustering algorithm, e.g., K-means, and an anomaly detection algorithm, e.g., iForest, are employed to provide detailed analysis of known anomalies and unlabeled data instances respectively. After the inchoate process of the complex input, another method, e.g., weighted support vector machine, is adopted to finalise the anomalies. On the other hand, in [164], a number of independent models for data prediction are used to investigate the relationship among data features. Models are evaluated according to their performance in fitting the training dataset and further combined and leveraged to implement the eventual anomaly detector. Therefore, it is witnessed that developing a hybrid method is of great benefit under the presence of a complex anomaly detection scenario and more efforts could be made to further advance this field. Nevertheless, in this thesis, due to the reason that extending specific features of an anomaly detection method is the primary target, hybrid methods are not largely explored. For more details concerning hybrid methods for anomaly detection, interested readers are recommended to refer to [3][19][90][115][176] for more details.

Automated Learning

Another major research direction for the purpose of designing a well-rounded anomaly detection method/system is the automated learning. In 2015, Yahoo proposed a generic and scalable system for automated time-series anomaly detection [125]. The system maintains a library of time-series anomaly detection methods which are examined individually to identify the best method for a specific time series. It is claimed that the system is currently utilised by many teams in Yahoo for daily monitoring of critical time series data. In machine learning, another related work that integrates algorithm selection with hyperparameter tuning is undertaken in [70] which also exploits multiple techniques, e.g., Bayesian optimisation and ensemble learning. It is expected that similar approaches can be useful in designing a generic system for automated anomaly detection. This is still a brand new research direction that few researchers have set foot on.

Other Methods

Hybrid methods and automated learning are two research directions that worth exploring. In recent years, there is another hot notion called *Big Data* which incites intense research efforts in the domain of tensor analysis [150]. A tensor is a general term for describing arrays. A 1-dimensional array is called vector and 1st-order tensor, while a 2-dimensional array is called matrix and 2nd-order tensor. To process a high-dimensional tensor, novel methods are required. From the perspective of anomaly detection, more novel techniques need to be investigated to achieve accurate and efficient tensor-based anomaly detection. A survey of tensor-based anomaly detection is presented in [69]. Due to the fact that many existing tensor-based techniques are not designed for anomaly detection, it becomes a novel research direction.

Comparison to Other Surveys

The related work in this section covers a broad view of general anomaly detection methods. It is also compared with many existing surveys of anomaly detection, the result of which is shown in Table 2.1. Note that the review proposes a new taxonomy of the existing anomaly detection methods and it investigates more categories of related methods compared to all the other related surveys.

2.2.7 Summary

This section illustrates diverse categories of techniques for general anomaly detection. Specifically, five categories of methods are discussed according to the information they utilised in determining the concept of abnormality, i.e., distance information, density information, boundary information, partition and other properties. All these classes of methods have their own pros and cons. Generally, it is the property of the target dataset that should be fully examined before picking the most appropriate method for anomaly detection. Luckily, it is witnessed that hybrid methods are becoming increasingly skilled in integrating the benefits from diverse methods under certain anomaly detection scenarios so that generating better anomaly detection performances is made possible. Overall speaking, there have been fruitful research outcomes in general anomaly detection methods and it is shown that more powerful and rounded systems with distinctive features are promising and await further exploration. In this thesis, efforts are made to support specific methods with additional features. It is expected that the additional features will greatly contribute to more powerful anomaly detection methods so as to ease the problems and challenges mentioned in the previous chapter.

Table 2.1 A comparison of related surveys

	Our	[3]	[6]	[9]	[19]	[29]	[79]	[76]	[90]	[101]	[115]	[149]	[176]	[171]
Strategies														
Rule	√		√	√	√	√	√			√			√	
Case	√									√			√	
Expect.	√					√								
Propert.	√									√				
Methods														
Distance	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Density	√	√	√	√	√	√	√	√		√	√	√	√	√
Boundary	√		√		√	√		√	√		√	√		√
Represent.	√					√			√	√		√		
Partition	√		√								√			
Subspace	√			√	√	√	√		√	√		√	√	√
Component	√					√								
Latent Fun.	√													
Latent Info.	√		√		√	√		√	√	√	√	√		√
Correlation	√		√	√	√	√	√			√		√	√	
Hybrid	√	√			√				√		√		√	

2.3 Time Series Anomaly Detection

In the previous two sections, related works concerning general anomaly detection methods are categorised and detailed. From this section, the focus is placed on the anomaly detection of a specific data type, i.e., time series. This is due to the reason that time series anomaly detection is the primary application investigated in this thesis. It is beneficial to have a clear understanding of what the actual problem formulation we are dealing with and pinpoint the related works targeting at solving the problem. A thorough discussion of all the related problem formulations and related works of time series data mining can be found in [80][30][63]. As pointed out by [80], the problem of time series anomaly detection has two main tasks. The first task deals with the detection of anomalous time series over a given time series database, whereas the second task concerns detecting anomalies within a single time series. **In this thesis, identifying anomalies within a single time series is the primary focus and it is worth noting that the time series concerned is considered to be a continuous sequence of numerical data** rather than a discrete sequence of symbolic data which requires a different set of methods [30]. In the next subsection, basic strategies for analysing time series anomalies are discussed. A broad taxonomy of time series anomaly detection methods is presented afterward. For additional information concerning time series anomaly detection methods, interested readers are recommended to refer to [47][4].

2.3.1 Strategies for Time Series Anomaly Detection

Before moving forward to introducing time series anomaly detection strategies, let us firstly clarify the problem formulation of the concerned time series anomaly detection. Imagine a time series starting from time 0 to time T, i.e., $X =< x_0, x_1, \cdots, x_T >$. The time series anomaly detection is to identify the set of anomalous points within the time series or label all the points within the time series with $Y =< y_0, y_1, \cdots, y_T >$ indicating whether their corresponding points are anomalous or not. More formally, the problem is:

Given: a time series X,
Find: anomalous points in X.

Reduction to General Anomaly Detection

Although the above problem formulation concerns the anomaly detection in a time series which possesses sequential information among the data points, general anomaly detection methods are still applicable. This is achieved through the reduction of the time series anomaly detection problem. A typical method for the problem reduction is time-delay embedding

[172] which adopts a sliding window to construct multivariate data instances from the original time series. For example, supposing that the size of the sliding window is set to E, the method of time-delay embedding would construct a new dataset from the time series X as:

$$X' = \{x_E(t)|t = 1,2,\cdots,T-E+1\},$$

where it has:

$$x_E(t) = [x_t, x_{t+1}, \cdots, x_{t+E-1}].$$

Consequently, a time series is converted to a set of multivariate data instances that enables general anomaly detection methods. Note that the sequential information within the original time series is encoded within the multivariate data. Therefore, the general anomaly detection methods also take into consideration the sequential information. It is also worth noting that, the utilisation of the sliding window has multiple tricks that support the advancement of anomaly detection performance. For example, if the periodicity of the time series is known, non-overlapping sliding windows could be used instead of traditional time-delay embedding, which is supposed to generate fewer data but better performance.

Time Series Specific Analysis Methods

Besides the methodology of reducing time series anomaly detection to general anomaly detection, there is also a broad range of time series analysis methods that are applicable in anomaly detection. Time series analysis methods model or analyse the sequential information directly to seek stable models or patterns that are essential in the original time series. With the assumption that the time series consistently obeys these models or patterns, they are utilised to check the consistency of the time series for anomalies. A very straightforward example would be the utilisation of Linear Regression (LR) [117] to model the evolving trend of the time series. Data values that are far from the expected values given by the model are likely to be anomalies. In the following subsection, the time series specific analysis methods are reviewed and categorised.

2.3.2 Techniques for Time Series Anomaly Detection

Time series anomaly detection has been an important research topic for many years. Typical approaches for solving the problem cover 1) the methods that target at analysing univariate time series and 2) the methods that model multivariate time series for anomaly detection. In this section, four categories of the time series specific anomaly detection methods are detailed and their advantages and disadvantages are also discussed.

Statistical Prediction Methods

Statistical prediction methods for anomaly detection mainly concerns the predictive models of univariate time series. These models aim at extracting the patterns in a given time series so as to accurately predict subsequent values. Traditional methods, such as Auto-regressive model (AR) [15] and Moving Average model (MA) [61], are all statistical analysis methods that are potent in modeling univariate time series for prediction. More advanced methods are developing rapidly in recent years. ARIMA [25], Seasonal ARIMA [93] and the whole class of Exponential Smoothing (ES) methods [72][73] all find numerous applications in various domains.

For univariate time series, statistical prediction models are relatively simple models that are easy to train and utilise. As a result, the prediction of the future time series is fast and suitable for working online. For stable time series that possess steady time series patterns, these models always demonstrate very good performance in practice. Nevertheless, due to the fact that they are originally designed for univariate time series, most of these methods are not applicable in multivariate time series prediction, which limits their applications. Besides, in situations where the target time series is noisy and unstable, these predictive models are too simple to capture the complex patterns and dynamics of the time series. Therefore, they always do not adapt well in sophisticated scenarios.

Time Series Decomposition Methods

Rather than dedicated methods for time series anomaly detection, time series decomposition methods are originally designed to decompose a time series into several critical components, e.g., the growing trend and the seasonal pattern. The accurate separation of these components contributes to the analysis of time series. It helps with the understanding of a time series and makes the prediction easier and more accurate. Typical time series decomposition methods include additive decomposition, multiplicative decomposition, X-12-ARIMA decomposition and STL [91][33]. The positive effect of time series decomposition has been validated in many works, e.g., STL is employed in [224] as a primary technique to achieve the decomposition of the seasonal information, which successfully supports effective long-term time series anomaly detection. Generally speaking, time series decomposition is an excellent tool for preprocessing time series for further applications. It is widely adopted especially when analysing time series with growing trend or strong periodicity. Nonetheless, for time series with no explicit periodicity, time series decomposition may incur additional work and may not have positive effects on time series anomaly detection.

State Transition Models

State transition models are methods that examine the dynamic state transitions within a system or a time series. In state transition models, a time series is assumed to maintain a steady state transition pattern that can be modeled as a stable property. This stable model is normally realised as a Markov model [250], a Hidden Markov Model (HMM) [138] or a Finite State Machine (FSM) [130]. Over the years, numerous research has been undertaken to exploit the state transition models for sequential data anomaly detection. In 2015, [77] proposed a Hidden Markov anomaly detector that is shown to outperform the one-class SVM in situations where data have latent dependency structures. Additionally, in [134], a timed automata is employed to profile the normal sequential behavior of a digital video broadcasting system for the purpose of anomaly detection. These works all demonstrate the effectiveness of state transition models in analysing sequential data.

Compared to statistical prediction methods and time series decomposition methods, state transition models are superior in processing multivariate time series. Furthermore, the models grasp the intrinsic state transitions among sequential data, which promotes the understanding of the underlying system and provides a better explanation for potential anomalies. On the other hand, these models are not without their drawbacks. An essential problem of these models is the utilisation of the states which are either not known in advance, i.e., hidden state, or discrete symbols that ignore certain information, e.g., discretising values in time series to definite states [130]. Both of the situations limit the performance of time series anomaly detection. Moreover, the trained model is always not flexible enough when the target time series or system is dynamically changing.

Regression Methods

Regression methods are critical tools for time series prediction and anomaly detection. And different from statistical prediction methods, regression methods naturally support multivariate data processing. As a result, they are widely applicable. Typical examples of regression methods, e.g., LR [117], Generalised Regression Neural Network (GRNN) [100], Support Vector Regression (SVR) [161] and GP [236], have been implemented in various fields for the purpose of time series prediction and anomaly detection. In [156], Ma and Perkins formulated the problem of time serise anomaly detection under the framework of SVR, and in [184] GP is suggested for anomaly detection and removal. LR is utilised in [11] for time series anomaly detection. Generally speaking, most of the regression methods for time series forecasting are widely applicable in time series anomaly detection problems with some minor adjustments.

As has been mentioned, regression methods are multivariate data processing methods that are naturally suitable for time series prediction. According to distinct scenarios, different regression methods could be used. Regression methods support the recognition of complex time series patterns which statistical prediction methods are always not able to capture. Another advantage that supports the utilisation of regression methods is that most of the regression methods are adaptive to model changes and therefore potent in dealing with streaming time series datasets. On the other hand, the drawbacks of using regression methods are that they are relatively complex models and normally require more data and resources for training. In environments where resources are limited, the utilisation of regression methods should be carefully examined.

2.3.3 Summary

In this section, the strategies and techniques for time series anomaly detection are examined. Compared with general anomaly detection, time series anomaly detection has to extract and analyse the sequential information within the time series in order to make anomaly detection decisions. Based on how to extract the sequential information, two strategies are identified, i.e., reducing time series anomaly detection to general anomaly detection and implementing time series specific analysis methods. On the other hand, four types of time series specific analysis methods are elaborated according to their capability of handling multivariate time series. Overall, this section presents a brief taxonomy of time series anomaly detection methods and gives a solid overview of the background in time series anomaly detection.

2.4 Conclusion

In this chapter, an organised overview of the techniques for both general and time series anomaly detection is present. Generally speaking, analysing time series is a more challenging task because of the dynamically evolving patterns and complex contextual information in time series. Although general anomaly detection techniques are applicable in time series anomaly detection problems, additional efforts are required to meet the specific challenges in processing time series. From the next chapter, four key research topics will be detailed concerning general and time series anomaly detection: 1) better accuracy, 2) integration of contextual information, 3) anomaly analysis and 4) parameter-free and dynamic anomaly detection model. Each of these topics is critical and should be carefully examined for practical anomaly detection.

Chapter 3

Support Vector Data Description with Relaxed Boundary

3.1 Introduction

Anomaly detection is widely applied in diverse fields, such as the cloud computing systems. As one of the fastest-developing technologies, cloud computing is becoming increasingly ubiquitous. Cloud services, e.g., Dropbox, Google App Engine and Windows Azure, have infiltrated into diverse aspects of the society and the performance of such services enormously affects our daily lives. Therefore, many efforts have been made by cloud service providers and researchers on anomaly detection over the cloud services [56][104][165][224]. However, with the ever-growing complexity, cloud computing systems also facilitate the emergence of Big Data [105][252], which raises huge concern in accurate anomaly detection due to the greater difficulties in accurate data collection, preprocessing, etc. More specifically, the time-series data collected from cloud services are always noisy and trigger high volume of false alarms in cloud monitoring systems. As a result, excellent time series anomaly detection methods are urgently required to reduce the false alarms in the systems.

In reality, time series anomaly detection has been a critical research topic in the domain of data analysis for decades. It has been widely applied to various areas related to the processing of sequential datasets. Over the years, diverse researches have been undertaken to detect time series anomalies aiming for low false positive rate and high efficiency [92][125][127][126][224]. This chapter, however, is particularly devoted to investigating the method of Support Vector Data Description (SVDD) [208] for time series anomaly detection with the targets of reducing false alarms while maintaining efficiency.

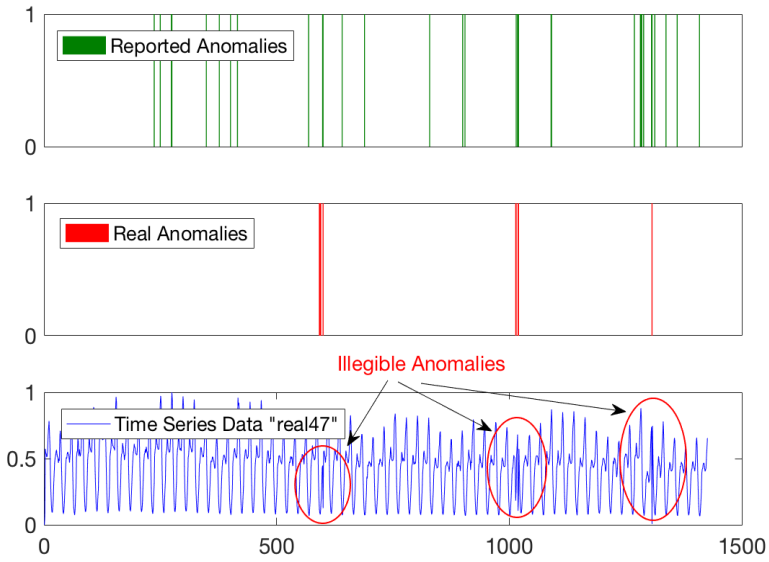


Fig. 3.1 Anomaly detection over time series “real47” using SVDD

Generally speaking, SVDD is a promising and popular method for achieving efficient, accurate, and interpretable anomaly detection. Its popularity is mainly owing to the fact that it is a non-parametric sparse model, which naturally supports multivariate one-class classification with a concise and easy-to-understand geometric interpretation of the results. Previously, a similar method of SVDD, i.e., One-class Support Vector Machine (OCSVM) [201], was invented for the purpose of one-class classification. The authors in [157] pioneered the employment of OCSVM for general time series anomaly detection with a sliding window. However, in applications such as cloud computing systems, the noisy nature and high velocity of the time series, i.e., service performance metrics, contribute to the great challenges in anomaly detection. As a result, when SVDD is used in time series anomaly detection, its drawbacks, i.e., high false alarm/positive rate and huge computational complexity, become prominent, which hinder its practical applications.

In order to illustrate the high false positive rate of SVDD in time series anomaly detection, a real-world time series “real47” selected from Yahoo benchmark datasets [249] is depicted in Fig. 3.1 with its manually labelled anomalies and the anomalies reported by conventional SVDD. It is worth stressing that SVDD identifies all the labeled anomalies, which are not easily spotted by traditional statistical methods, e.g., box-plot rule [96], because the anomalous points have normal values (between 0.2 and 0.8). It is the unusual shapes and patterns of the values that contribute to the anomalies. Nevertheless, in this example, false alarms dramatically outnumbers the correctly identified anomalies, which renders conventional SVDD impractical. Furthermore, from the perspective of computational complexity, over

a thousand of Quadratic Programming (QP) problems, i.e., the underlying mathematical problem of SVDD, need to be solved during the process (with sliding window [157]), which leads to the drain of significant computational resources. Both of the abovementioned issues are severe and can influence the performance of SVDD in time series anomaly detection significantly.

Consequently, to ensure the applicability of SVDD, this chapter introduces a more practical tool for time series anomaly detection by utilising linear programming SVDD (LPSVDD) with relaxed data description boundary to achieve better accuracy and efficiency. It is found that relaxing LPSVDD (RLPSVDD) results in a Linear Programming (LP) problem which is much easier to tackle than a nonlinear programming problem. Along with jumping window, RLPSVDD manages to boost the process of time series anomaly detection, and, at the same time, achieve high accuracy. The sufficient condition for selecting valid parameters to ensure the practicality of RLPSVDD is also provided. To conclude with, the main contributions of this chapter are:

- A novel linear programming SVDD (LPSVDD) is formulated and relaxed for detecting anomalies in time series;
- To ensure that the relaxed LPSVDD (RLPSVDD) is practical for anomaly detection, important insights of how to select its parameters have been presented. The sufficient condition of a practical RLPSVDD is given.
- Extensive experiments are conducted on Yahoo benchmark datasets which contain different metrics of various Yahoo services. The results demonstrate that RLPSVDD can achieve higher capability and accuracy in identifying various time series anomalies.

As will be shown by the experiment results, RLPSVDD performs averagely the best among all the compared methods, which firmly supports the utilisation of RLPSVDD in time series anomaly detection. Note that many other methods do not maintain a consistent performance due to their incapability of handling the diverse patterns of the time series. Consequently, RLPSVDD is able to provide more accurate results so that the reliability and trustworthiness of the applications, e.g., cloud computing systems, are more effectively preserved.

The rest of this chapter is organised as follows. Related work and background information are presented in the next section. In Section 3.3, the formulation of LPSVDD as well as the ways to relax it and ensure valid anomaly detection are elaborated. The details of the experiments and results analysis are provided in Section 3.4. Finally, conclusions are drawn in Section 3.5.

3.2 Related Work

3.2.1 Time Series Anomaly Detection

Many time series anomaly detection methods have been practically implemented and utilised. For instance, in 2013, Etsy open-sourced Skyline [62] for the passive monitoring of time-series metrics. The basic algorithms of the system cover Grubb's test [75], moving average [25] and other statistical methods. In 2014, Twitter published a method based on time series decomposition and generalised Extreme Studentised Deviate test (ESD) [224]. It is a practical approach built upon robust statistical methods and has been released under an open source license in [205]. Moreover, in 2015, Yahoo announced their framework for automatic time series anomaly detection, which is named Extensible Generic Anomaly Detection System (EGADS) [125]. Rather than merely trying a single method, EGADS includes a set of methods for time series anomaly detection, such as ARIMA [25], exponential smoothing [73], etc. Different from these methods and systems, which typically utilise statistical information, the method suggested in this chapter (RLPSVDD) can capture additional structural information of the dataset. While detecting time series anomalies, RLPSVDD takes not only the values but also the patterns of the time series into consideration, which reflects its superior capability to detect diverse types of anomalies.

Besides the well-developed statistical methods, other methods under the broad umbrella of machine learning have gained increasing popularity in recent years. For example, Principal Component Analysis (PCA) has been examined specifically for time series anomaly detection. In [127], PCA was utilised to find the normal and anomalous components of network link traffic measurements. A new link traffic measurement is then projected onto the anomalous components for the investigation of anomalies. Although elegant, PCA and many other spectral analysis methods always require a time series to be folded into a matrix according to its intrinsic period before anomaly detection. As a result, these methods are more suitable for time series with stable periodicity. In RLPSVDD, however, the unstable patterns of a time series do not interfere with the effectiveness and changing patterns can be detected online.

In [29], many other types of machine learning methods for anomaly detection in general datasets are summarised. A very powerful type of the methods is based on neural networks. As a concrete example, in 2015, Numenta implemented and open-sourced a special type of neural network for time series anomaly detection called Hierarchical Temporal Memory (HTM). The method was compared with other related methods in [126], where the results demonstrate the excellence of Numenta HTM in time series anomaly detection. In contrast to HTM, RLPSVDD shows its conciseness in implementation, as it can be implemented in several lines of code, and its competitive effectiveness, which will be shown in Section 3.4.

3.2.2 Support Vector Data Description (SVDD)

In comparison with other related work, SVDD [208] shows its uniqueness in time series anomaly detection. SVDD is essentially a one-class classification or data description method that features a nonparametric model without requiring the knowledge of explicit data distribution. The existence of the sparse support vectors in SVDD enables a computationally efficient decision function for online anomaly detection. Although it is theoretically proved to be equivalent to One-class SVM [201] with the utilisation of Gaussian kernel, its easy-to-understand geometric interpretation contributes to its popularity. More specifically, the mathematical formulation of SVDD indicates a process of searching for the minimum enclosing ball, determined by the center a and radius R, of a set of data instances $x = \{x_1, x_2, \cdots, x_N\}$, which are projected by a projection function $\phi(\cdot)$ into a high-dimensional space [208]. Note that, N is the number of the data instanes. In addition, with the introduction of the nonnegative slack variable ξ_i weighted by a constant C for each data instance $x_i \in \mathbb{R}^D$, where $i \in \{1, 2, ..., N\}$ and D is the dimension, the radius of the ball is shrunk to exclude possible outliers. The final formulation of SVDD follows:

$$\begin{aligned} &\textbf{min}_{a,R^2,\xi} \quad R^2 + C\sum_i \xi_i \\ &\textbf{s.t.} \quad \forall i, \quad \|\phi(x_i) - a\|^2 \leq R^2, \quad \xi_i \geq 0. \end{aligned} \tag{3.1}$$

There have been a number of studies on improving the accuracy and efficiency of SVDD. Many of these studies focus on enhancing the accuracy of data description under the presence of a noisy dataset. A natural way to customise the way SVDD handles noisy data is to fine tune the weights of the slack variables of all the data instances. Liu et al. [146] and Chen et al. [45] both adopted this idea and designed distinct weights for the slack variables. As the original slack variables only have the effect of shrinking the data description boundary, Chen et al. also showed in [45] that SVDD could introduce an additional constant slack variable to improve the generalisation performance through expanding the boundary. This technique has also been introduced by Wu et al. [239] and Mu et al. [154] with distinct formulations that are superior in the automatic decision of the constant slack variable. Nonetheless, both of these works considered the presence of a labeled dataset, which is not always the case in time series anomaly detection. Another popular method of handling a noisy unlabeled dataset was introduced by Lee et al. [131] where weights are assigned to the distance between a data instance and the center of the enclosing ball. They designed the weights based on the density information of the data instances, encoding the capability of capturing the dense area of the dataset into SVDD. The resulting data description boundary is expanded and shifted towards the dense area of the dataset.

The work in this chapter, i.e., RLPSVDD, also expands the description boundary in order to obtain better results in time series anomaly detection. Nevertheless, it adopts linear programming SVDD for boundary expansion, which leads to the relatively easy-to-solve LP problem rather than the nonlinear programming problem that is witnessed in [131].

Besides accuracy, research has also been intensely undertaken concerning the efficiency of SVDD. In 2000, Campbell and Bennett introduced linear programming One-class SVM [32][31] that replaces the original QP problem with the LP problem. Later, in 2004, Chu et al. [39] proposed to utilise core-sets to scale up SVDD training and reduce the training time complexity from $O(N^3)$ to $O(N)$, where N is the size of the dataset. From the perspective of SVDD testing, Liu et al. [136] seeked a way to find the preimage of the center of the enclosing ball in the feature space and reduce the time complexity of SVDD testing to constant time. Another important work aiming at boosting the efficiency of SVDD was introduced by Tax and Laskov [128] where incremental SVDD is proposed to achieve online learning of SVDD. The method solves the underlying QP at the very beginning and only updates the Lagrangian multipliers [208] when new data instances are available. It avoids the retraining of the model and maintains the validity of the solution for data description. The incremental SVDD is elegant and practical, but it works only for conventional SVDD. Whether the same technique works for the density-induced SVDD [131], the underlying problem of which is a nonlinear programming problem, remains a nontrivial problem.

The work in this chapter is inspired by linear programming One-class SVM [31] and incorporates additional information for time series anomaly detection in a way that is similar to [131]. Different from [31], the proposed method achieves a flexible boundary that helps with better data description. It essentially contains the LP problem that is much easier to solve than that in [131]. From the perspective of time series anomaly detection, it achieves higher accuracy and performs more efficiently than the conventional SVDD.

3.3 Relaxing Linear Programming Support Vector Data Description

3.3.1 Linear Programming SVDD (LPSVDD)

The essentials of LPSVDD lie firstly in the problem formulation. Intuitively, to describe a dataset, LPSVDD considers that the data instances in an enclosing ball aim to move away from the center as far as possible. As a result, LPSVDD targets at minimising the sum of the distances between data instances and the data description boundary:

$$\begin{aligned} \mathbf{min}_{a,R^2} \quad & \sum_i \left(-\|\phi(x_i)-a\|^2+R^2\right) \\ \mathbf{s.t.} \quad & \forall i, \quad \|\phi(x_i)-a\|^2 \leq R^2. \end{aligned} \tag{3.2}$$

The D-dimensional data instances are denoted by $x_i \in \mathbb{R}^D$ with index $i \in \{1,2,...,N\}$, where N is the number of data instances. a and R are the center and radius of the enclosing ball, respectively. The mapping function $\phi(\cdot)$ projects a data instance into a high-dimensional feature space and enables the utilisation of the Gaussian kernel. By expanding the squared distance and employing some underlying properties and constraints, the formulation is reformulated as:

$$\begin{aligned} \mathbf{min}_{\alpha,R^2} \quad & \sum_i \left(\sum_j 2\alpha_j K(x_i,x_j) - 2 + 2R^2 \right) \\ \mathbf{s.t.} \quad \forall i, \quad & 1 - \sum_j \alpha_j K(x_i,x_j) \leq R^2, \\ & \sum_j \alpha_j = 1, \quad \alpha_j \geq 0, \end{aligned} \tag{3.3}$$

where $j \in 1,2,...,N$ is also the index of data instances and α_j is the Lagrangian multiplier for data x_j. The properties and constraints leveraged in Eq. (3.3) contain:

$$\begin{aligned} \phi(x_i)\cdot\phi(x_j) =& K(x_i,x_j) = e^{\frac{-\|x_i-x_j\|^2}{\sigma^2}}, \\ \phi(x_i)\cdot\phi(x_i) =& K(x_i,x_i) = e^{\frac{-\|x_i-x_i\|^2}{\sigma^2}} = 1, \end{aligned} \tag{3.4}$$

$$a = \sum_j \alpha_j \phi(x_j), \tag{3.5}$$

$$a^2 = 1 - R^2. \tag{3.6}$$

Eq. (3.4) is the formulation of the Gaussian kernel with the parameter σ. Eqs. (3.5) and (3.6) display the implicit constraints that appear as a result of solving the conventional SVDD problem [208]. With all these conditions, the above two formulations are equivalent. In Fig.3.2, where the formulations are interpreted geometrically, the feature space onto which all the data instances are mapped is depicted as the surface of the bigger circle denoted by its center O, and the smaller circle denoted by its center a is the minimum enclosing ball that surrounds all the data instances. Due to the utilisation of the Gaussian kernel, O is a unit circle, i.e., the radius of O is 1. This geometric interpretation is the same as that in SVDD.

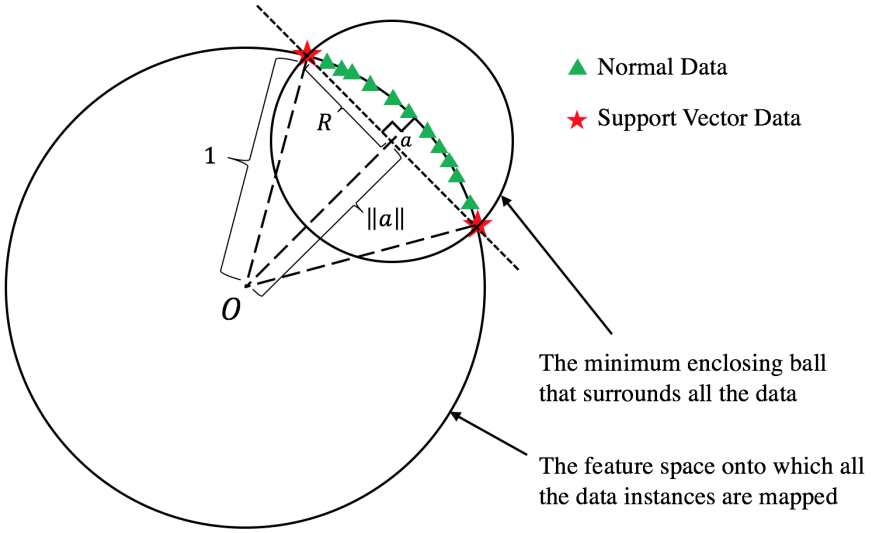


Fig. 3.2 LPSVDD in feature space with constraint $a^2 + R^2 = 1$

However, the new formulation still contains the QP problem due to the constraint in Eq. (3.6). Therefore, the constraint is relaxed to reduce the QP problem (Eqs. (3.3)-(3.6)) to the LP problem (Eqs. (3.3)-(3.5)) that can be solved directly by the existing LP solvers. After solving the problem, the data instances x_j with non-zero Lagrangian multipliers α_j, known as support vectors (SV), and the radius of the enclosing ball

$$R^2 = \max_{i \in SV} 1 - \sum_{j \in SV} \alpha_j K(x_i, x_j), \tag{3.7}$$

are facilitated to test whether a new data instance x_{new} is inside the minimum enclosing ball by checking:

$$1 - \sum_{j \in SV} \alpha_j K(x_{new}, x_j) \leq R^2. \tag{3.8}$$

If Eq. (3.8) holds, the new data instance is considered to be normal. Otherwise, an anomaly is detected.

3.3.2 Relaxing LPSVDD (RLPSVDD)

In many applications, the available data instances come only from a restricted portion of the entire dataset. Therefore, from the perspective of anomaly detection, the available data instances are so limited that a detailed description would cause too many false alarms, which would notably undermine the practice of anomaly detection. It has already been shown

that, with an extension of the data description boundary, SVDD achieves higher accuracy in one-class [45] and binary class classification [154][239]. In this work, LPSVDD is relaxed for one-class classification and a better boundary extension than that reported in [45] is accomplished. By incorporating additional information, the relaxed LPSVDD (RLPSVDD) is capable of expanding the boundary towards the preferred directions. The method used is analogous to that in [131]. However, RLPSVDD maintains its quality as the LP problem.

Concretely, to provide a flexible description of a given dataset, the formulation of RLPSVDD follows:

$$\begin{aligned} \mathbf{min}_{\alpha,R^2} \quad & \sum_i \left(\sum_j \alpha_j K(x_i,x_j) - 1 + R^2 \right) \\ \mathbf{s.t.} \quad \forall i, \quad & \rho_i \cdot \left(1 - \sum_j \alpha_j K(x_i,x_j) \right) \leq R^2, \\ & \sum_j \alpha_j = 1, \quad \alpha_j \geq 0, \end{aligned} \tag{3.9}$$

where the parameter ρ_i is determined by the additional information of data instance x_i and is always nonnegative. In contrast to other formulations of related purpose, such as the density-induced SVDD [131] in which $\rho_i \geq 1$, the range of ρ_i here is $[0,+\infty)$, which means that RLPSVDD cannot only expand the boundary for relaxed anomaly detection, but also shrink and move the boundary of the data description. As a result, it is unnecessary to introduce slack variables in this formulation. Note that, the additional information can be fundamentally different from that given by the original dataset, e.g., a textual description of the symptoms of a patient can be the additional information w.r.t. the detected quantitated measurements of the symptoms, e.g., the blood pressure. In one-class classification for anomaly detection, a natural piece of additional information is the holistic or detailed description of the original dataset [228]. The details of the additional information used in this chapter follow that in [131] and will be explained in Section 3.4.

In fact, there are also other distinct formulations that are capable of shrinking and expanding the data description boundary in LPSVDD. But focus has been placed on Eq. (3.9) for two reasons. (1) It intuitively relaxes LPSVDD while maintaining the LP problem. Consider that $\rho_i = \rho$ for all i, the formulation scales the radius R, i.e., $R^2 = \rho \cdot \max_i \left(1 - \sum_j \alpha_j K(x_i,x_j)\right)$, to directly control the boundary. (2) It leads to an easy-to-understand criterion that rules the proper selection of the parameter ρ_i.

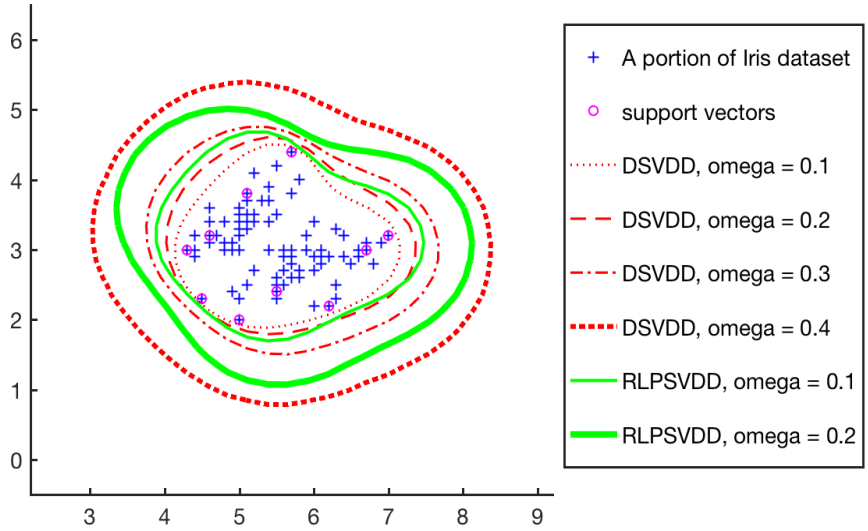


Fig. 3.3 The restriction of parameter selection in DSVDD and RLPSVDD

3.3.3 The Restriction of the Parameter ρ_i

As mentioned in Section 3.3.2, the parameter ρ_i is flexible in the range $[0, +\infty)$. Although certain flexibility has been provided in assigning the parameter, it should be careful that even a normal selection of the parameter may lead to an impractical one-class classifier for a specific dataset. Fig. 3.3, where the first 100 data instances (first 2 dimensions) of Iris dataset [123] are depicted, constitutes a good example of this. The red dashed curves are the data description boundaries given by the density-induced SVDD (DSVDD) with different parameter settings, i.e., different omega (ω in Section 3.4.1). Similarly, the green solid curves are the boundaries given by RLPSVDD with data density as the additional information weighted by the different parameter settings, i.e., different omega. Note that ρ_i is positively proportional to omega (see Section 3.4.1). Therefore, with an increasing omega, the boundary inflates gradually. Nevertheless, when omega is chosen as 0.5 in DSVDD and 0.3 in RLPSVDD, no visible boundary exists for data description, which is counterintuitive. The information about the formulation of density information and parameter ρ_i will be presented in Section 3.4 along with the detailed experiments explaining why the boundary vanishes. In this section, a formal definition of practical RLPSVDD is firstly given and the theoretical analysis on how to ensure the practicality of RLPSVDD is then presented.

> **Definition 1 (One-class Classifier):** A one-class classifier f is a function or model that is trained by a set of one-class training data, denoted as X, takes input $x \in \mathbb{R}^N$, where N is the dimension of the input data, and outputs $y = f_X(x) \in \{0, 1\}$ to indicate whether the input data x belongs to the same class as X.

Definition 2 (Practical/Impractical One-class Classifier): A one-class classifier f is impractical if $\forall x \in \mathbb{R}^N, f(x) = 1$ or $\forall x \in \mathbb{R}^N, f(x) = 0$. In other words, an impractical one-class classifier does not provide valuable information for data classification. A one-class classifier that is not impractical is called practical.

Based on the definitions, RLPSVDD is a one-class classifier with

$$f(x) = sgn\left(R^2 - \left(1 - \sum_j \alpha_j K(x, x_j)\right)\right)$$

derived from Eq. (3.8). Moreover, $sgn(z)$ is a signal function that outputs 1 if $z \geq 0$, and 0 otherwise. To ensure a RLPSVDD is a practical one-class classifier, it must suffice that:

$$1 - \max_x a \cdot \phi(x) < R^2 < 1 - \min_x a \cdot \phi(x). \tag{3.10}$$

Upper Bound

Theoretically, $a = \sum_j \alpha_j \phi(x_j)$ and $\phi(x)$ are two vectors in the feature space. The minimisation of their multiplication equals 0, if $\phi(x)$ is orthogonal to a. This happens when x is sufficiently far away from all the training data x_j. Therefore:

$$1 - \min_x a \cdot \phi(x) = 1. \tag{3.11}$$

Lower Bound

For $\max_x a \cdot \phi(x)$, it is obtained when vector $\phi(x)$ is paralell to a, i.e., their included angle $\theta = 0$:

$$1 - \max_x a \cdot \phi(x) = 1 - \|a\|\|\phi(x)\| \cos\theta = 1 - \|a\|. \tag{3.12}$$

Note that, $\|\phi(x)\| = 1$, because Eq. (3.4) induces that $\phi(x) \cdot \phi(x) = \|\phi(x)\|^2 = 1$. This lower bound for R^2 is not always tight because it is not always possible to find such an x in the input space that $\phi(x)$ is parallel to a in the feature space. To seek for a tight lower bound, the following problem is solved instead [136]:

$$\min_x \|\phi(x) - a\|^2, \tag{3.13}$$

which is equivalent to solving $\max_x a \cdot \phi(x)$, because $\|\phi(x)\|^2$ and $\|a\|^2$ are constants. Setting the derivative of the target function to 0 yields:

$$\hat{x} = \frac{\sum_j \alpha_j K(\hat{x}, x_j) x_j}{\sum_j \alpha_j K(\hat{x}, x_j)}, \tag{3.14}$$

which could help update $\hat{x}$ iteratively. Nonetheless, the initial assignment of $\hat{x}$ is very important that bad initial values will make $\hat{x}$ fall into local minima. To solve this problem, Liu et al. [136] suggested to find $\hat{x}$ directly through assuming $\phi(\hat{x}) = \Psi_a = \gamma a$, where Ψ_a is called the agent of a, and $\gamma = \frac{1}{\|a\|}$. The formulation for calculating $\hat{x}$ is obtained:

$$\hat{x} = \frac{\sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j) x_i}{\sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j)}. \tag{3.15}$$

Although this formulation of $\hat{x}$ does not always lead to the correct results in RLPSVDD due to the vague validity of the assumption, it provides a good initial value for updating $\hat{x}$ iteratively according to Eq. (3.14). This is expected to provide a good $\hat{x}$ such that a tight lower bound of R^2 is obtained. It should be noted that, in the cases where $\hat{x}$ is a local minimum, the lower bound still works although it is not tight. This is because $1 - \max_x a \cdot \phi(x) < 1 - a \cdot \phi(\hat{x})$.

Results and Remarks

For the above reasons, a practical RLPSVDD has a sufficient condition:

$$1 - a \cdot \phi(\hat{x}) < R^2 < 1. \tag{3.16}$$

In other words, the selection of the parameter ρ_i in RLPSVDD should maintain the resulting radius within the adequate range to ensure a practical solution. In practice, let us firstly consider the situation when all the parameters are the same, i.e., $\forall j, \rho_j = \rho$. In RLPSVDD, $R^2 = \rho \cdot \max_{x_{sv}}(1 - a \cdot \phi(x_{sv}))$, where x_{sv} denotes a support vector. As a result, when the following condition is met, the solution of RLPSVDD is always valid:

$$\frac{1 - a \cdot \phi(\hat{x})}{\max_{x_{sv}} 1 - a \cdot \phi(x_{sv})} < \rho < \frac{1}{\max_{x_{sv}} 1 - a \cdot \phi(x_{sv})}. \tag{3.17}$$

For the situations where distinct ρ_js are presented, a similar result is obtained:

$$\frac{1 - a \cdot \phi(\hat{x})}{1 - a \cdot \phi(x_{sv})} < \rho_{sv} < \frac{1}{1 - a \cdot \phi(x_{sv})}. \tag{3.18}$$

Here, ρ_{sv} and x_{sv} are the solutions of $\text{argmax}_{\rho_{sv},x_{sv}} \rho_{sv} \cdot (1 - a \cdot \phi(x_{sv}))$. Although checking these two results is equivalent to checking $1 - a \cdot \phi(\hat{x}) < R^2 < 1$, they do provide indirect information about how to select an appropriate ρ_j so as to make RLPSVDD a practical one-class classifier. After solving the LP problem in RLPSVDD, a, x_{sv}, and $\hat{x}$ are known.

3.3.4 Time Series Anomaly Detection

Time-delay Embedding for Data Construction

Essentially, RLPSVDD is an anomaly detection method for multivariate data. In time series anomaly detection, a single time series is univariate. Therefore, it should be converted to a multivariate dataset to enable RLPSVDD. A typical way to achieve this is using a time-delay embedding process [157] or a sliding window. Specifically, a time-delay embedding process turns a time series $x(t), t = 1, \cdots, N$, into a multivariate time series dataset:

$$X(t) = \{x_E(t) | t = 1, \cdots, N - E + 1\}, \tag{3.19}$$

where E is the size of the time-delay embedding and N is the length of the time series. Also,

$$x_E(t) = [x(t) \quad x(t+1) \quad \cdots \quad x(t+E-1)]. \tag{3.20}$$

As a result, $X(t)$ is the dataset from which RLPSVDD will detect anomalies. It is noted that although there are some other ways of constructing the multivariate time series, such as replacing $x_E(t)$ with a set of its samples, the basic time-delay embedding process works very well for the experiments presented in this chapter and setting $E = 2$ or 3 can always yield good results for anomaly detection.

Initial Window for Model Training

To achieve time series anomaly detection, RLPSVDD starts with an initial window W_{init} of data instances for model training. The size of the initial window is denoted as w_{init}. Typically, in SVDD, all the data instances in the initial window are required to be normal, which means that no anomalies are allowed in the initial window. On the contrary, in RLPSVDD, the initial window could contain anomalies, whose effect is reduced by the relaxation of the boundary.

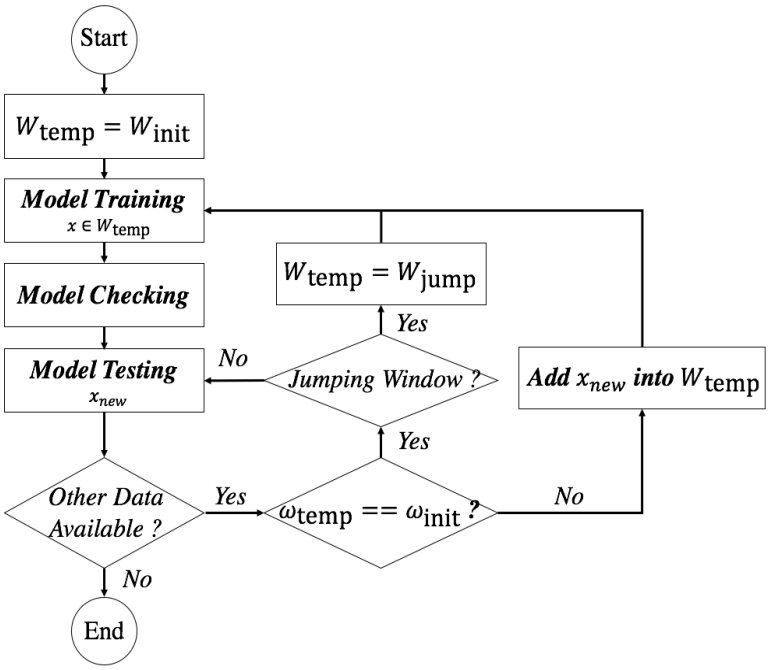


Fig. 3.4 RLPSVDD-based time series anomaly detection workflow

Jumping Window for Model Updating

In time series anomaly detection, model-based methods, for example SVDD and SVR [156], will always encounter the problem of model updating. Especially, the identification of the perfect timing for model updating is not trivial. There are typically two strategies for model updating: (1) constantly and (2) periodically. When SVDD is updated constantly, a new incoming data instance will initiate model updating for retraining SVDD and that, as a result, consumes a significant amount of time. Although solving an LP problem (RLPSVDD) is much more efficient than solving a QP problem (SVDD), constantly updating RLPSVDD is also time intensive. Therefore, in this chapter, RLPSVDD is updated periodically when a fixed quantity of continuous anomalies are detected. In other words, when all the data instances in a window are all detected as anomalies, RLPSVDD is retrained. The window is called jumping window W_{jump}, because it corresponds to a jumping model. Its size is denoted as w_{jump}.

The Workflow

To conclude with the process of employing RLPSVDD for time series anomaly detection, a conceptual workflow is given in Fig. 3.4. For a time series, the process starts with setting a temporal window W_{temp} as the initial window, i.e., $W_{\text{temp}} = W_{\text{init}}$. The size of W_{temp} is w_{temp}.

Therefore, the very initial RLPSVDD model is trained using all x in W_{init}. After model checking as presented in Section 3.3.3, model testing takes place to determine whether a new data instance x_{new} is anomalous according to the trained model. This is followed by checking whether there is any new data available. If there is any remaining data instance, the process of model updating will be initiated once a jumping window full of anomalies is witnessed or the temporal window has the size different from the initial window. When a jumping window appears, the temporal window is set as the jumping window, i.e., $W_{\text{temp}} = W_{\text{jump}}$. With model updating, RLPSVDD finds the pattern of the time-series data in the temporal window and new data instances are added to the temporal window until the size of the window matches that of the initial window. Hence, a stable model is obtained when $w_{\text{temp}} = w_{\text{init}}$. As the model is stable, the identification of the jumping window will be initiated again. The whole process ends whenever there is no new data instance available.

3.4 Experiment Results

In this section, the experiments conducted to verify the correctness of the main results in this chapter are presented. More specifically, the main results are:

- In RLPSVDD, the parameters should be carefully chosen and Eqs. (3.17) and (3.18) are the sufficient conditions to ensure the practicality of the model;
- RLPSVDD can achieve high capability and improved accuracy in identifying various types of time series anomalies in cloud service performance metrics.

3.4.1 RLPSVDD with Constrained Parameter

As mentioned in Section 3.3.3, theoretical bounds for the parameter ρ_j exist. To prove the validity of the theoretical bounds, an exemplary dataset is picked here to showcase the practicality of RLPSVDD under different parameter settings. The final results are shown in Table 3.1.

Dataset

The well-known Iris dataset from the UCI machine learning repository [123] is selected as the exemplary dataset. To facilitate data visualisation and manually label the practicality of RLPSVDD, the first 2 dimensions of the first 100 data instances, i.e., the first 2 classes of the flowers, are chosen as the targets for data description. The visualisation of the data instances is shown in Fig. 3.3.

Table 3.1 Practicality checking for different models Learned from Iris dataset

Method	Lower Bound		R^2 / ρ / ρ_{sv} / ρ'_{sv}		Upper Bound	Prac.
SVDD	0.6909	$<$	$R^2 = 0.7562$	$<$	1	1
RLPSVDD with Constant Parameter ρ	**0.5357**	$>$	$\rho = 0.1$	$<$	1.0584	**0**
	0.5389	$>$	$\rho = 0.2$	$<$	1.0935	**0**
	0.5271	$>$	$\rho = 0.3$	$<$	1.1464	**0**
	0.6002	$>$	$\rho = 0.4$	$<$	1.2127	**0**
	0.6989	$>$	$\rho = 0.5$	$<$	1.2383	**0**
	0.7486	$>$	$\rho = 0.6$	$<$	1.2512	**0**
	0.9523	$>$	$\rho = 0.7$	$<$	1.3195	**0**
	0.9576	$>$	$\rho = 0.8$	$<$	1.3224	**0**
	0.9574	$>$	$\rho = 0.9$	$<$	1.3229	**0**
	0.9574	$<$	$\rho = 1.0$	$<$	1.3231	1
	0.9572	$<$	$\rho = 1.1$	$<$	1.3238	1
	0.9572	$<$	$\rho = 1.2$	$<$	1.3238	1
	0.9565	$<$	$\rho = 1.3$	$<$	1.3242	1
	0.9558	$<$	$\rho = 1.4$	$>$	**1.3258**	**0**
	0.9544	$<$	$\rho = 1.5$	$>$	**1.3286**	**0**
	0.9542	$<$	$\rho = 1.6$	$>$	**1.3290**	**0**
	0.9515	$<$	$\rho = 1.7$	$>$	**1.3319**	**0**
	0.9515	$<$	$\rho = 1.8$	$>$	**1.3319**	**0**
	0.9514	$<$	$\rho = 1.9$	$>$	**1.3324**	**0**
	0.9524	$<$	$\rho = 2.0$	$>$	**1.3338**	**0**
RLPSVDD with Distinct Parameter ρ_j	0.9032	$<$	$\rho_{sv} = 1.0166$ $(\omega = 0.05)$	$<$	1.3082	1
	0.8279	$<$	$\rho_{sv} = 1.0335$ $(\omega = 0.10)$	$<$	1.2683	1
	0.9443	$<$	$\rho_{sv} = 1.3337$ $(\omega = 0.15)$	$<$	1.5305	1
	0.9229	$<$	$\rho_{sv} = 1.4680$ $(\omega = 0.20)$	$<$	1.5666	1
	0.9067	$<$	$\rho_{sv} = 1.6159$ $(\omega = 0.25)$	$>$	**1.6049**	**0**
	0.8453	$<$	$\rho_{sv} = 1.7787$ $(\omega = 0.30)$	$>$	**1.6531**	**0**
	0.8420	$<$	$\rho_{sv} = 1.9579$ $(\omega = 0.35)$	$>$	**1.6667**	**0**
	0.8378	$<$	$\rho_{sv} = 2.1551$ $(\omega = 0.40)$	$>$	**1.7836**	**0**
	0.8610	$<$	$\rho_{sv} = 2.3722$ $(\omega = 0.45)$	$>$	**2.0439**	**0**
	0.8610	$<$	$\rho_{sv} = 2.6111$ $(\omega = 0.50)$	$>$	**2.0439**	**0**
RLPSVDD with Distinct Parameter $\rho'_j = \rho_j - 1$	**0.4767**	$>$	$\rho'_{sv} = 0.0166$ $(\omega = 0.05)$	$<$	1.0154	**0**
	0.9973	$>$	$\rho'_{sv} = 0.0626$ $(\omega = 0.10)$	$<$	1.8269	**0**
	0.8336	$>$	$\rho'_{sv} = 0.0953$ $(\omega = 0.15)$	$<$	1.3188	**0**
	0.7030	$>$	$\rho'_{sv} = 0.1291$ $(\omega = 0.20)$	$<$	1.1446	**0**
	0.7324	$>$	$\rho'_{sv} = 0.2394$ $(\omega = 0.25)$	$<$	1.2593	**0**
	0.9505	$>$	$\rho'_{sv} = 0.7787$ $(\omega = 0.30)$	$<$	1.3686	**0**
	0.9692 (0.9370)	$>$	$\rho'_{sv} = 0.9579$ $(\omega = 0.35)$	$<$	1.5496	**0** (1)
	0.9170	$<$	$\rho'_{sv} = 1.1551$ $(\omega = 0.40)$	$<$	1.5710	1
	0.9168	$<$	$\rho'_{sv} = 1.3722$ $(\omega = 0.45)$	$<$	1.5712	1
	0.9067	$<$	$\rho'_{sv} = 1.6111$ $(\omega = 0.50)$	$>$	**1.6049**	**0**

RLPSVDD with a Constant Parameter

RLPSVDD is firstly tested using constant parameters, i.e., $\forall j,\ \rho_j = \rho$. From $\rho = 0.1$ to $\rho = 2.0$, 20 instances are tested. Only 4 out of 20 experiment instances ($\rho = 1.0, 1.1, 1.2, 1.3$) yield practical RLPSVDD, which confirms the significance of parameter selection. A random selection of parameter would easily lead to an impractical RLPSVDD. Among the impractical RLPSVDD, a small $\rho \leq 0.9$ causes $\frac{1-a\cdot\phi(\hat{x})}{\max_{x_{sv}} 1-a\cdot\phi(x_{sv})} > \rho$ and $\rho \geq 1.4$ results in $\rho > \frac{1}{\max_{x_{sv}} 1-a\cdot\phi(x_{sv})}$, both of which violate the criterion in Eq. (3.17). On the other hand, for practical RLPSVDD the criteria are all met. As one may argue, the practicality can be checked directly according to Eq. (3.16). It is not necessary to compute the upper bound and the lower bound for ρ. However, clear bounds of ρ help with achieving a better parameter selection. It is noticed that for practical RLPSVDD, the upper bounds are roughly around 1.32, while the lower bounds are approximately around 0.95. These results provide very good indications for parameter selection, i.e., for $\rho \in [0.96, 1.32]$ it is optimistic that a practical RLPSVDD can be obtained.

RLPSVDD with Distinct Parameters

To assign distinct weights/parameters ρ_j for data instances of the exemplary dataset, the density information of the dataset is extracted according to [131]:

$$\begin{aligned} \rho_j &= \exp\{\omega \times \frac{\Im^k}{d(x_j, x_j^k)}\}, \\ \Im^k &= \frac{1}{n}\sum_{j=1}^{n} d(x_j, x_j^k), \end{aligned} \tag{3.21}$$

where x_j is a data instance, x_j^k represents the kth neareat neighbour of x_j, $d(\cdot,\cdot)$ stands for a function measuring the distance between two data instances, and ω is a weighting parameter. Originally, ω is limited to the range of $[0, 1]$ [131], which means that $\forall j\ 1 \leq \rho_j$, and this only expands the boundary of RLPSVDD. For the completeness of the experiment, another set of experiments with $\rho_j' = \rho_j - 1$ is conducted to investigate the behavior of RLPSVDD with shrunk boundary. Note that $k = 3$ and $\sigma = 1$ (see Section 3.3.1) are assigned to all related experiments in Table 3.1 and Fig. 3.3.

According to the experiment results, only when the condition in Eq. (3.18) is satisfied a practical RLPSVDD exists. It also confirms that the selection of the parameter is critical especially when RLPSVDD is used to shrink the boundary of data description. In one of the experiments with $\rho_j' = \rho_j - 1$ and $\omega = 0.35$, the method for identifying the lower bound of ρ_j' fails to find bounds tight enough for accurately determining the practicality of

the corresponding model. This is due to the selection of an inappropriate initial value for calculating the lower bound. After the manual tuning of the initial value, the corrections are given in the brackets. Although this is unsatisfactory, it still maintains the validity of the results in Section 3.3.3 because the criteria in Eqs. (3.17) and (3.18) are actually conservative ones.

Besides the given results, the experiments also reveal that with the increment of ρ or ω from a low value to a high one the practicality of RLPSVDD changes from impractical (violation of $\frac{1-a\cdot\phi(\hat{x})}{\max_{x_{sv}} 1-a\cdot\phi(x_{sv})} < \rho, \rho_{sv}, \rho'_{sv}$) to practical and then back to impractical (violation of $\rho, \rho_{sv}, \rho'_{sv} < \frac{1}{\max_{x_{sv}} 1-a\cdot\phi(x_{sv})}$). This indicates that the parameters for the practical RLPSVDD are within a concentrated range and, according to the position of ρ or ρ_{sv}, the parameters can be tuned in order to find an appropriate setting.

Remark: Other than the density information, RLPSVDD can also incorporate various kinds of information. Generally speaking, the additional information is responsible for weighting the original data instances so as to get a better data description for anomaly detection. Therefore, the methods that provide a probabilistic or continuous score for measuring the abnormality of each data instance, e.g., Gaussian Mixture Model (GMM) [14] and Principal Component Analysis (PCA) [102], and the methods that extract the holistic information of the given dataset [228] are all feasible approaches to supply additional information. Nonetheless, the resulting effect of RLPSVDD largely depends on whether the selected method correctly models or measures the abnormality of the given dataset. Hence, selecting the optimal information, which reflects key features of the dataset, is critical and it always requires expert knowledge.

3.4.2 Time Series Anomaly Detection

To demonstrate the effectiveness of RLPSVDD in time series anomaly detection, the real world datasets in Yahoo benchmark [249] are used to compare the anomaly detection accuracy of RLPSVDD to some other anomaly detection methods used by Etsy [62], Twitter [224], Numenta [126], and Yahoo [125]. All these methods being compared are well-known and utilised practically in industry. It has been publicly accepted that there is not a single time series anomaly detection method that outperforms all the others in all time-series [125]. Therefore, the experiments aim at testing the general capability of RLPSVDD in time series anomaly detection and emphasising that RLPSVDD is a promising method that performs generally better than compared methods. The content and process of the experiments are detailed below.

Dataset

More specifically, the A1Benchmark of Yahoo benchmark datasets [249] is selected as the group of target datasets for time series anomaly detection. The A1Benchmark is reported to be based on the real metrics concerning various Yahoo cloud services, e.g., Yahoo Membership Login (YML) [125]. It covers 67 time-series with various seasonality, distinct changing patterns, and diverse type of anomalies. Compared to other benchmark datasets in [249] and [126], A1Benchmark shows better diversity and is much more realistic and harder to analyse. The datasets are labeled by humans and, therefore, the marked anomalies reflect how service/network administrators would like the anomalies to be reported. Despite the uncertain consistency of the labeled anomalies, A1Benchmark is undoubtedly a good time-series dataset for testing the general effectiveness (precision and recall) of an anomaly detection method.

Models for Comparison

The methods and systems to be compared with RLPSVDD are Twitter AnomalyDetection [224], Etsy Skyline [62], Numenta Hierarchical Temporal Memory (HTM) [126] and EGADS [125] from Yahoo. It is worth stressing that Skyline, AnomalyDetection and EGADS heavily depend on statistical anomaly detection methods. Skyline and EGADS are integrated systems rather than a single method. While Skyline implements the majority vote strategy for anomaly detection, EGADS uses an individual method with the best performance for analysing a specific time series. Therefore, for EGADS, the Olympic model (Seasonal Naive model) is employed to model the time series and two anomaly detection methods, namely the Extreme Low Density model and the Simple Threshold model, are compared with other methods. These models are selected according to the reported information in EGADS [125]. It was reported that the Olympic model performs the best on average among the time series models and the Extreme Low Density model and the Simple Threshold model are two standard anomaly detection methods in EGADS. Besides these, Numenta HTM, a machine learning technique, is also compared. It is based on a special form of neural network that arranges neurons in a hierarchical way. The neural network is trained by the time series and their patterns are recognised for anomaly detection. All the methods compared are also presented in Table 3.2. The method "BEST" is not a new method but a notation of the best method among the above methods in terms of the accuracy of anomaly detection over a specific time series.

Table 3.2 Methods to be compared with RLPSVDD

Method	Description
Twitter, AnomalyDetection (AD) [224]	Twitter's method for anomaly detection, which uses STL, i.e., a seasonal-trend decomposition procedure, piecewise median, and Extreme Student Deviate test.
Numenta, Hierarchy Temporal Memory (HTM) [126]	Numenta's machine intelligence technique for analysing spatial and temporal patterns of a dataset. A special form of neural network lies in the heart of the technique.
Etsy, Skyline [62]	Skyline is an anomaly detection system built by Etsy. It relies on an ensemble of algorithms that vote for anomalies, i.e., majority vote. The default algorithms cover Grubb's test, etc.
Yahoo, EGADS, SN/SO [125]	Yahoo's anomaly detection system with SN/SO method. SN/SO stands for **S**imple Threshold anomaly detection model with **N**ull/**O**lympic model for time series fitting.
Yahoo, EGADS, EN/EO [125]	Yahoo's anomaly detection system with EN/EO method. EN/EO stands for **E**xtreme Low Density anomaly detection model with **N**ull/**O**lympic model for time series fitting.
BEST	The best anomaly detection method (among the above) for a specific time series.

Metric for Comparison

For comparing the methods, the stardard F1-score is calculated:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}, \tag{3.22}$$

where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$. TP, FP, FN are standard notations for true positive, false positive, and false negative respectively. Hence, $0 \leq F_1 \leq 1$. To refrain from the situation where the denominator equals 0, F_1 is set to 0 whenever $TP = 0$, $TP + FP = 0$, or $TP + FN = 0$. This indicates that if an anomaly detector fails to detect any anomaly the performance of the anomaly detector is considered to be unsatisfactory. Also, if the time series does not contain any anomaly, all the anomaly detectors would be treated equally and marked as useless.

Parameter Tuning

Parameter tuning is a significant step that would tremendously influence the results of anomaly detection. In this section, the way of parameter tuning for all these methods is elaborated. (1) To use RLPSVDD, one would need to tune three kinds of parameters: k and ω to extract density information (Section 3.4.1); σ to control the width of the Gaussian kernel (Section 3.3.1); and the size of the windows E, w_{init} and w_{jump} (Section 3.3.4). It seems troublesome to tune these parameters simultaneously for the best result, yet it turns out that in general only ω is critical and setting $k = 10$, $\sigma = 0.5$, $E = 2$, $w_{init} = 200$, and $w_{jump} = 25$ can always result in satisfactory outcomes. Therefore, in the experiments, the above parameters are firstly fixed and ω is manually tuned, which is relatively easy. After that, the simulated annealing algorithm (100 iterations maximum) is utilised to identify the best set of parameters to minimise $FP + FN$ for each time series in A1Benchmark. (2) Similarly, for Twitter AnomalyDetection, a set of proper parameters is manually identify and then the simulated annealing is employed to find the best parameters to achieve the minimisation of $FP + FN$. Because Twitter AnomalyDetection is more of a statistical method and faster than RLPSVDD, its maximum iteration number of simulated annealing is set to 1000. (3) As discussed in [126], Numenta HTM is robust to parameter settings and there is no need and interface for parameter tuning. (4) Skyline shares the same advantage with Numenta and achieves anomaly detection without model/threshold configuration [62]. (5) In terms of the methods in EGADS, the Olympic model requires 4 parameters, the Extreme Low Density model requires 2 parameters, and the Simple Threshold model asks for 1 parameter. For Olympic model, setting the parameters to default values always gives better results and

Table 3.3 RLPSVDD V.S. other methods in terms of F1-Score

Comparison	Win	Draw	Loss
RLPSVDD V.S. **Twitter AD**	20	39	8
RLPSVDD V.S. **Numenta**	33	30	4
RLPSVDD V.S. **Skyline**	37	28	2
RLPSVDD V.S. **EGADS, SO**	35	28	4
RLPSVDD V.S. **EGADS, EO**	40	20	7
RLPSVDD V.S. **EGADS, SN**	35	28	4
RLPSVDD V.S. **EGADS, EN**	51	15	1
RLPSVDD V.S. **BEST**	10	41	16

one of its parameters can be dynamically optimised to get the best model. Therefore, the parameters of the Olympic model are set to default values and its dynamic parameter tuning is enabled. For Extreme Low Density model and Simple Threshold model, the parameters are concerned with the percentage of the reported anomalies. Thus, the real percentage of anomalies in each time series is used to assign the parameters. After the initial setting of the parameters in EGADS, they are further manually checked and tuned to minimise $FP + FN$.

Results and Discussion

As the F1-score is calculated using the precision (true alarm rate) and the recall (detection rate) of a method, it is an integrated metric that represents the general effectiveness of the method. Accordingly, a method with lower false alarm rate and higher detection capability typically has higher F1-score. The final anomaly detection results of all the methods are compared in terms of the F1-score in Fig. 3.5. The Y-axis represents the indexes of the time series, while the X-axis helps with the comparison of the percentage of the F1-score of all the compared methods. The figure is essentially a 100% stacked bar chart. The different colors and styles correspond to the different methods, which are all shown at the bottom of the figure.

At first glance, it is hard to identify a method that outperforms the others over all the 67 time-series. This result reflects the claim in [125], which underlines that anomalies are use-case specific and an individual method can hardly cover all the types of anomalies. In other words, there is no perfect anomaly detection method. Nevertheless, the result shows the holistic performance of the methods. For instance, it is clear that the overall performance of EGADS EN is not satisfactory because its performance is always below average and the performance of EGADS EO indicates high variance. The methods from Twitter, Numenta and Etsy are moderate methods because they do not achieve excellence

compared to other methods. RLPSVDD has a relatively stable performance and, for some time series, it noticeably transcends other methods. All this information is beneficial for understanding the overall performance of the methods and helps with the selection of the methods.

A more concise but informative comparison of the results is presented in Table 3.3. It shows the comparison of the results of RLPSVDD against all the other methods in terms of F1-score over all the 67 time-series (67 matches, in A1Benchmark). Concerning the competition between RLPSVDD and any other individual method, RLPSVDD always beats the other method with more than 20 wins and less than 10 losses. Although a large number of the matches result in a draw, the overall outcome of the comparison justifies the preference for RLPSVDD. Twitter AnomalyDetection is a very competitive method that performs nicely on average. However, the results in Fig. 3.5 reveal that for some specific time series, e.g. no. 7, 20, 38, 40, its capability of detecting the anomalies is less prominent. A similar situation stands for the methods used by Numenta (time series no. 14, 20, 26, 40, etc.) and Etsy (time series no. 18, 20, 38, 49, etc.). As the results indicate, the methods from EGADS, namely SN/SO and EN/EO, do not perform well as expected. This is partially due to the manual parameter tuning that is not capable of finding the best parameter setting. Moreover, as in the inchoate version of EGADS, SN/SO and EN/EO are relatively simple methods that do not perform very well in the presence of special anomalies, for example change points.

A further comparison between RLPSVDD and the BEST model reveals that RLPSVDD is a very promising method that works well on average in general time series (41 Draws) and it outperforms all the other methods in some specific time series (10 Wins in Fig. 3.6a with the index of the time series as the horizontal axis and the F1-score as the vertical axis). As a result of a detailed investigation, it is known that these specific time series are all with change points or pattern anomalies, such as time-series “real47” in Fig. 3.1. Therefore, it is concluded that RPLSVDD possesses stronger capability to detect anomalous patterns [29] in time series. Regarding the time series over which RPLSVDD does not win the competition (16 Losses in Fig. 3.6b with the same axises as that in Fig. 3.6a), it is partially due to the reason that RLPSVDD focuses more on detecting local pattern anomalies and are not skilled in detecting long-term contextual anomalies. Nonetheless, RLPSVDD averagely gets high F1-scores and is better than most of the compared methods. To conclude with the results, RLPSVDD is an excellent method for time series anomaly detection in terms of its strong capability to detect various types of time series anomalies.

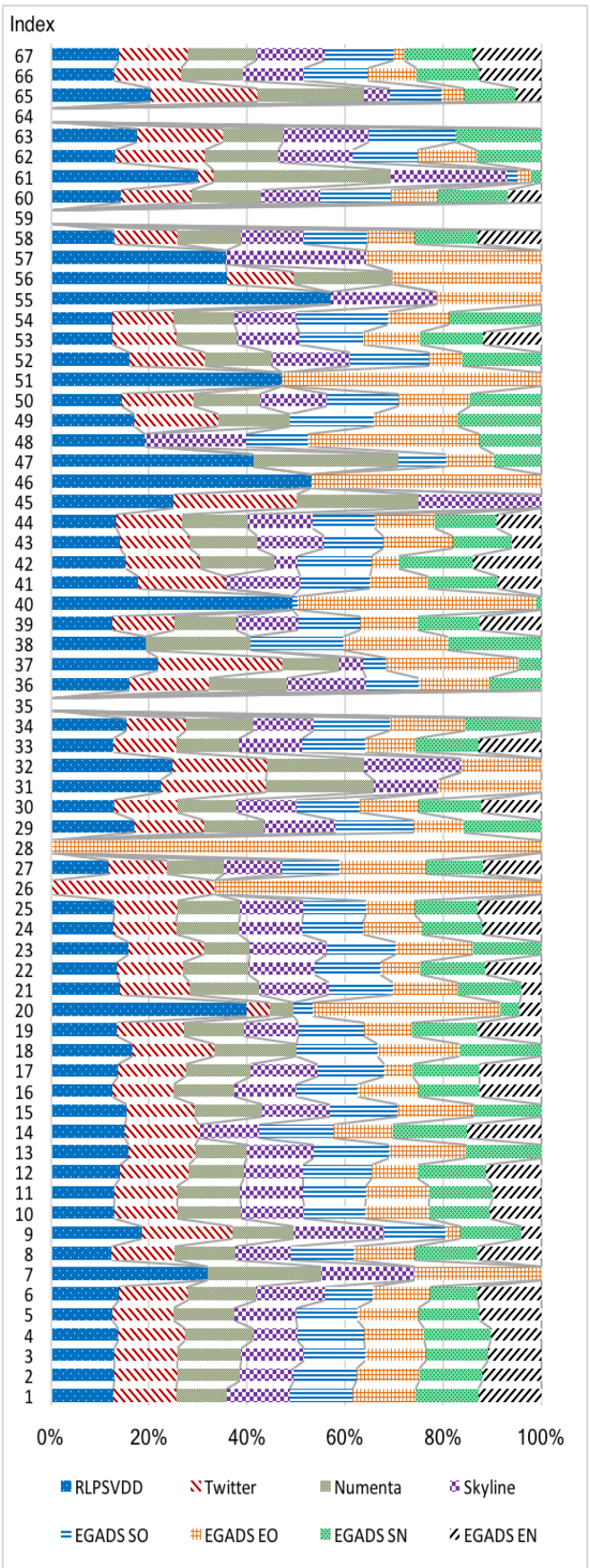
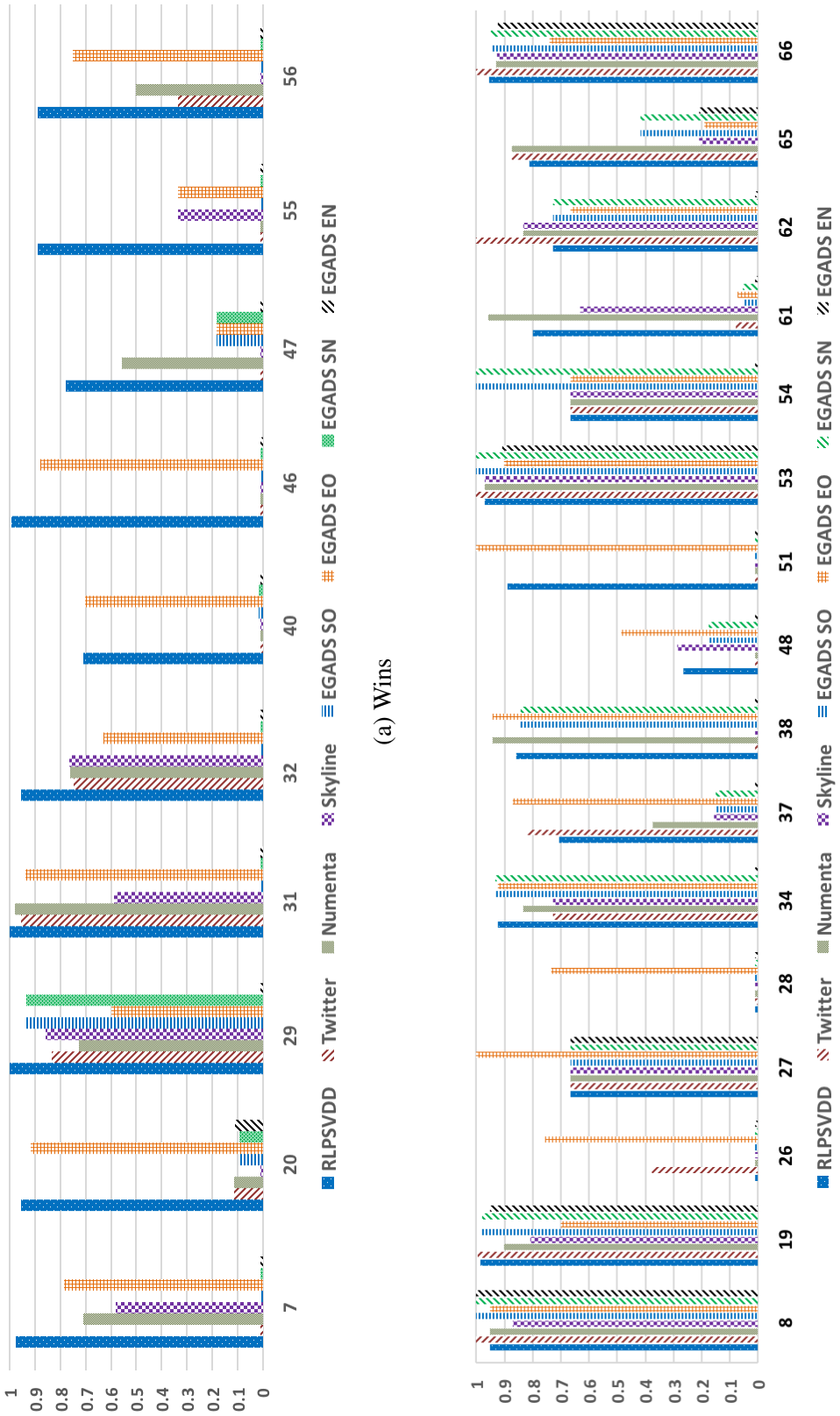


Fig. 3.5 The comparison of all methods over Yahoo A1Benchmark



(a) Wins

(b) Losses

Fig. 3.6 The cases in which RLPSVDD wins/losses the competition

In order to fully unleash the capability of RLPSVDD, parameter tuning is an unavoidable process that needs to be carefully conducted according to different time series. EGADS has given a great example of leveraging alarm filtering to fulfill online parameter adjustment. Simply tuning the weight for the additional information in RLPSVDD fits the spirit of alarm filtering, while tuning all the related parameters is so much more complex that further research efforts are required.

3.5 Conclusion

In this chapter, the utilisation of Support Vector Data Description (SVDD) for time series anomaly detection of service performance metrics in cloud computing systems has been investigated. Due to the high false alarm rate and low time efficiency of the original method, a relaxed linear programming SVDD (RLPSVDD) has been proposed to cope with the problems. RLPSVDD solves a linear programming problem to provide a flexible data description for time series anomaly detection. With the proper selection of the parameters, a practical RLPSVDD can be guaranteed and enjoys a generally stronger capability to detect various types of anomalies compared to other methods of similar purpose. Experiments on well-known benchmark datasets have confirmed the validity of the analysis and RLPSVDD has been shown to be a method full of potentials in time series anomaly detection for cloud service metrics. As a next step, contextual information of time series data, e.g., periodic patterns, linear trends, etc., will be integrated with RLPSVDD to test its capability of detecting contextual time series anomalies.

Chapter 4

Support Vector Data Description with Contextual Information

4.1 Introduction

In the previous chapter, specific methods have been developed to supply accurate and efficient time series anomaly detection capability that is required in diverse systems. Although a method may be superior in performance under a specific condition, current anomaly detection systems usually implement a set of anomaly detection methods to support the best possible system performance. For instance, the EGADS [125] is an outstanding example in which many anomaly detection methods serve as candidates for analysing cloud computing services. These methods monitor the user and system behaviors, model the normal operations and report anomalies whenever a significant deviation from the expected status of the system or actions of the user is witnessed. For most anomaly detection methods, e.g., box-plot method [96], conventional Support Vector Data Description (SVDD) [208] and Replicator Neural Network (RNN) [94], they solely focus on detecting the anomalies through analysing the primary information, yet do not explicitly process related contextual information. Consequently, these methods provide little clue within the method for interpreting the anomalies, such as a further classification of the detected anomalies or the potential reasons that cause the anomalies. I would like to argue that the explicit processing of contextual information is of great benefit for better understanding the anomalies. Therefore, in this chapter, I propose a way of anomaly detection with contextual information that is accurate, capable of distinguishing contextual anomalies from typical/point anomalies and thus achieves interpretable anomaly analysis. The practical application of this method will largely benefit the anomaly detection systems where contextual information plays a vital role.

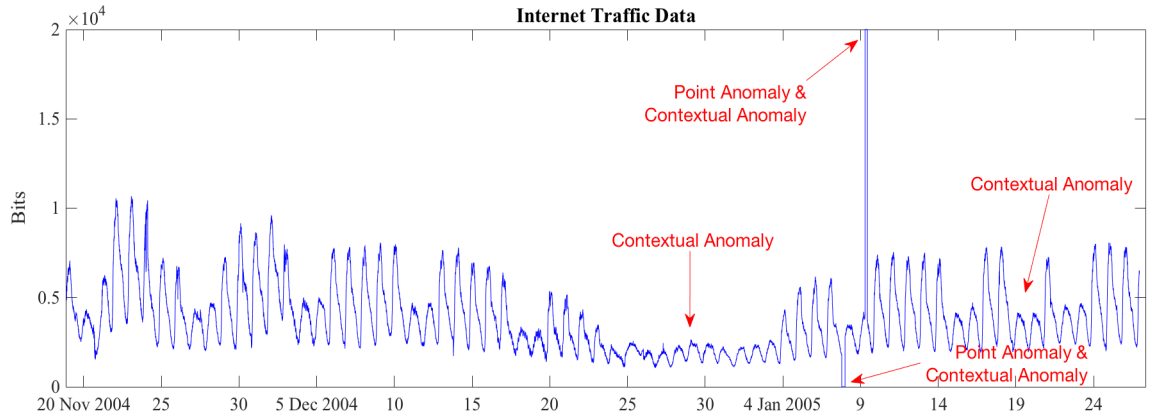


Fig. 4.1 An example of different anomalies

A key target in this chapter is to address anomaly interpretation. As a concrete example of anomaly interpretation, let us consider the situation in Fig. 4.1, where a time series of Internet traffic is recorded with marked anomalies. In the depicted time series, the metric has two types of anomalies, which are point anomaly and contextual anomaly. The point anomaly refers to the rare values that deviate greatly from typical values, while the definition of the contextual anomaly depends on the context. In Fig. 4.1, the time series has a clear periodic pattern, i.e., a single period contains 5 high peaks followed by 2 low peaks. Considering the periodic pattern as the contextual information, the contextual anomalies in the time series are the data points that are normal in terms of their data values, but abnormal because they do not follow the periodic pattern. Therefore, the anomaly interpretation in this time series anomaly detection task is to differentiate the point anomalies from the contextual anomalies. And it is preferable if the differences of the anomalies are identified within a single anomaly detection method. To summarise, this chapter makes the following contributions:

- An anomaly detection method with the capability of integrating additional contextual information is introduced for better understanding the cause of anomalies, e.g., to determine whether the anomaly is triggered by primary information or contextual information.
- Experiments are conducted using Yahoo benchmark datasets. It is demonstrated that the proposed method successfully identifies anomalies and provides useful information to help explain the cause of the anomalies.

Before heading to the next section, there are two critical points that are to be clarified: 1) why is it better to treat contextual information individually rather than combining it with primary information; and 2) what is the granularity of the anomaly interpretation in this chapter? They are presented in the following two subsections.

4.1.1 Why is it Better to Treat Contextual Information Separately?

In typical anomaly detection problems, only a single source of information is witnessed and processed. The contextual information is very often embedded within the primary source of information. For instance, in a periodic time series, the context of periodicity is embedded within the time series and it is not explicitly processed by most anomaly detection methods. Therefore, there exist three basic strategies of handling the contexts: 1) using primary information but implicitly process contexts within the methods, e.g., exploiting neural networks for time series anomaly detection without explicit identification of time series periodicity; 2) explicitly identifying contexts and combining with primary information for anomaly detection, e.g., constructing a new dataset with primary information and contexts for anomaly detection; 3) processing contextual information separately from primary information, e.g., treating primary information and contextual information separately.

In this work, I adopt the third strategy but leverage a single method to integrate the processing of the two types of information. The reasons why the third strategy is preferred in this work are that: 1) although in some cases the contextual information is embedded in the primary information, they are fundamentally different information that aims at explaining distinct aspects of the target; 2) primary information is always available, however, the existence of contextual information during training and testing time may vary according to different applications. As a result, the separated processing of the two types of information is more practical; 3) considering primary information and contextual information separately supports the identification of the anomalies from the specific source of information, therefore enhancing the interpretability of the anomalies.

4.1.2 What is the Granularity of the Anomaly Interpretation?

The interpretability of an anomaly detection method describes the capability of the method in explaining the decisions it makes in related tasks. A well-known method that is skilled in interpretability is the Decision Tree (DT) method for classification. The granularity of the explanation made by DT is feature level. In other words, DT makes a decision because the values in specific features meet certain criteria. In this work, the targeted granularity is information level, which means the anomaly detection process targets at identifying the anomalous source of information rather than the anomalous features. Consequently, the results of the anomaly detection give clear differentiation of the point/group anomalies and the contextual anomalies. With this capability, applications are empowered to initiate diverse operations accordingly.

4.2 Related Work

As discussed in [29], there exist two basic strategies of handling contextual information in anomaly detection: 1) reducing the problem of contextual anomaly detection into point anomaly detection problem; 2) utilising the structure in data for context analysis. Note that these two strategies aim at analysing the contextual information while do not emphasise the process of the primary information. As a result, either the primary information and the contextual information are combined in related tasks, or the utilised anomaly detection methods solely detect contextual anomalies. For instance, ARIMA [25] identifies the time series anomalies which deviate from the periodic pattern of the time series, but it has the difficulty in detecting anomalous patterns of the shape of the time series. In this work, the focus is to achieve point anomaly detection and contextual anomaly detection at the same time. Therefore, the strategies for only contextual anomaly detection are not satisfactory.

Another related topic of this work is Learning using Privileged Information (LUPI) [228]. In LUPI, the objective is to achieve better learning results with the help from task-specific privileged information. For instance, in the task of image classification [197], additional textual description of the images in the training set can be exploited as the privilege information to support the differentiation of the different types of images. To process the privileged information, certain modifications in related learning methods are inevitable. A very famous method in the field is SVM+ [179] [227] whose training problem is formulated as:

$$\begin{aligned} \textbf{min}_{\omega\in\mathbb{R}^d, b\in\mathbb{R}, \omega^*\in\mathbb{R}^d, b^*\in\mathbb{R}} \quad & \frac{1}{2}\Big(\|\omega\|^2+\gamma\|\omega^*\|^2\Big)+C\sum_{i=1}^{N}\Big(\omega^* x_i^*+b^*\Big) \\ \textbf{s.t.} \quad \forall i\in\{1,2,\cdots,N\}, \quad & y_i\Big(\omega x_i+b\Big)\geq 1-\Big(\omega^* x_i^*+b^*\Big), \\ & \omega^* x_i^*+b^*\geq 0, \end{aligned} \tag{4.1}$$

where ω, b, ω^*, b^* represent the model parameters; N is the number of the data instances; x_i, y_i are the ith data instance and its label respectively; x_i^* stands for the privileged information of the ith data instance; and γ, C are the hyperparameters provided in advance. The method succeeds better results in classification tasks through integrating the primary information, i.e., x_i and y_i, with the privilege information, i.e., x_i^*, in the process of classification. In recent years, advancements have been made to improve the original SVM+ method, e.g., [106], and the idea has also been adopted in anomaly detection [262]. In this chapter, LUPI is generalised for the processing of contextual information and the proposed method implements a Linear Programming (LP) problem under the framework of SVDD, rather than a Quadratic Programming (QP) problem in SVM+, to fulfill anomaly detection.

4.3 Anomaly Detection with Interpretation

4.3.1 Linear Programming Support Vector Data Description

In the conventional SVDD method, a QP problem is solved to identify the anomaly detector. While in LPSVDD, an LP problem is adopted for describing the methods and the resulting optimisation problem is designed as:

$$\begin{aligned} \mathbf{min}_{\alpha,b} \quad & \sum_{i=1}^{N}\Big(\sum_{j=1}^{N}\alpha_j K(x_i,x_j)+b\Big) \\ \mathbf{s.t.} \quad & \sum_{j=1}^{N}\alpha_j K(x_i,x_j)+b\geq 0 \quad \forall i, \\ & \sum_{i=1}^{N}\alpha_i = 1, \quad \alpha_i \geq 0, \end{aligned} \tag{4.2}$$

where x_i and x_j represent data instances; N is the number of the data instances; b is a scalor and $\alpha = [\alpha_1 \alpha_2 \cdots \alpha_N]^T$ is a column vector with N elements; and $K(*,*)$ is a kernel function. This formulation is essentially an LP problem that is simpler in the form compared to a QP problem.

4.3.2 Linear Programming Support Vector Data Description Plus

To integrate contextual information with the formulation of SVDD and to provide detailed information about the detected anomalies, i.e., whether the anomalies relate intensively to their contexts, Linear Programming Support Vector Data Description Plus (LPSVDD+) is proposed to process selected contextual information [228], which is expected to supply anomaly detection systems with more flexibility of reporting anomalies. The formulation of the anomaly detection method over a set of data instances $X = \{x_1, x_2, \cdots, x_N\}$ with their corresponding contextual information $X^* = \{x_1^*, x_2^*, \cdots, x_N^*\}$ is as follows:

$$\mathbf{min}_{\alpha,b,\alpha^*,b^*} \quad \sum_i \left(\Big(\sum_j \alpha_j K(x_i,x_j)+b\Big)+\lambda\cdot\Big(\sum_j \alpha_j^* K(x_i^*,x_j^*)+b^*\Big)\right), \tag{4.3}$$

$$\mathbf{s.t.} \quad \left(\Big(\sum_j \alpha_j K(x_i,x_j)+b\Big)+\lambda\cdot\Big(\sum_j \alpha_j^* K(x_i^*,x_j^*)+b^*\Big)\right)\geq 0 \quad \forall i, \tag{4.4}$$

$$\sum_j \alpha_j^* K(x_i^*,x_j^*)+b^*\geq 0 \quad \forall i, \tag{4.5}$$

$$\begin{aligned} &\sum_j \alpha_j = 1, \quad \sum_j \alpha_j^* = 1, \\ &\forall j, \quad \alpha_j \geq 0, \quad \alpha_j^* \geq 0, \end{aligned} \tag{4.6}$$

where $x_i, x_j \in \mathbb{R}^D$ are D-dimensional data with index $i, j \in \{1, 2, \cdots, N\}$; $x_i^*, x_j^* \in \mathbb{R}^{D^*}$ are D^*-dimensional data with the same index; N is the number of data instances; λ is a hyper-parameter. Function $K(*,*)$ denotes the famous kernel function that enables the mapping of a data instance to a high-dimensional space for better generalisation of the method. In this chapter, the Gaussian kernel is selected as the kernel function for the experiments, i.e., with parameter σ, $K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{\sigma^2}}$.

Essentially, the formulation tries to integrate two LPSVDDs for training two types of information concerning a same object. The solution of the formulation leads to a description of the dataset that is helpful in anomaly detection. However, different from typical SVDD, this formulation gains two discriminants that are capable of detecting different types of anomalies. As has been mentioned, X is set as the main data information and X^* is the contextual information. Therefore, Eq. (4.5) mainly concerns the identification of the contextual anomalies, while Eq. (4.4) is applicable in detecting the overall normality of a data instance. To be more specific, the overall normality of a new data x_{new} with contextual information x_{new}^* is determined by:

$$\left(\left(\sum_j \alpha_j K(x_{new}, x_j) + b \right) + \lambda \cdot \left(\sum_j \alpha_j^* K(x_{new}^*, x_j^*) + b^* \right) \right) \geq 0. \tag{4.7}$$

If Eq. (4.7) holds, it is believed that x_{new} is normal. Otherwise, a general anomaly will be reported. On the other hand, the discriminant of whether the data has contextual anomaly is:

$$\sum_j \alpha_j^* K(x_{new}^*, x_j^*) + b^* \geq \min_i \sum_j \alpha_j^* K(x_i^*, x_j^*) + b^*. \tag{4.8}$$

If Eq. (4.8) holds, it means that x_{new}^* is normal. Otherwise, the contextual anomaly is confirmed. From the above two discriminants, a third one is made possible considering the enforcements of the constraints in Eqs. (4.4) and (4.5). This third discriminant, i.e.,

$$\sum_j \alpha_j K(x_{new}, x_j) + b \geq \min_i \sum_j \alpha_j K(x_i, x_j) + b, \tag{4.9}$$

demonstrates a practical way of measuring the normality of the primary information X.

To summarise, the new formulation introduces three different discriminants for identifying distinct types of anomalies. This novel capability enables contextual anomaly detection and supplies strong interpretations of the detected anomalies. In other words, the anomaly detection method can provide more details about the reason why a data instance is detected as anomalous, e.g., its contextual information deviates from the normal condition. Through leveraging this anomaly detection method, practical anomaly detection systems, such as intrusion detection systems, would be able to tell the contextual anomalies from other anomalies, and response actions could be initiated accordingly. To illustrate a concrete example, let us consider a set of web servers that will attract billions of requests on a particular day of the year, e.g., the Double 11 Festival (11.11) in Taobao. The high-rocketing number of the requests from the very beginning of the day would trigger lots of alarms in a typical intrusion detection system, indicating that the network performance indicators have shown abnormal behaviors that could be considered as suffering a large-scale DDoS attack. With the help of the contextual information, which tells the intrusion detection system that the abnormal request rate is actually normal on that day, the false alarms of the system will be significantly reduced according to the interpretations of the witnessed anomalies.

4.4 Experiment Results

4.4.1 Datasets

To demonstrate the effectiveness of the proposed method, two families of datasets, i.e., Yahoo A2Benchmark and A3Benchmark, are selected from Yahoo time series dataset repository [249] as the target datasets in the experiments. Yahoo A2Benchmark contains 100 time series datasets, each of which is a univariate time series that includes roughly 2000 data points. It is worth stressing that all the 100 time series in A2Benchmark have clear periodic patterns which are utilised in the experiments for designing contexts. Besides distinct periodic patterns, the time series also differ from each other in growth trends, data noise as well as anomalies. A3Benchmark is essentially similar to A2Benchmark. However, the 100 time series in A3Benchmark preserve much more complex time series periodicity and anomaly patterns, e.g., point anomaly, contextual anomaly and etc. Essentially, anomalies in time series of A3Benchmark are much harder to detect. Yet, as one will see later, the proposed method successfully facilitates the analysis of the detected anomalies in A3Benchmark, which supports better reaction towards the anomalies.

4.4.2 General Settings

Before heading to anomaly detection, a time series has to be preprocessed. In the experiments, four preprocessing operations are performed:

- **Detrending**: In some of the time series, there exist growing trends of the time series values. To ensure the accuracy of anomaly detection, the trends of the time series are removed using standard detrending methods supplied by Matlab [147].
- **Period Identification**: In order to formulate contextual information, the period of each time series should be identified. In this work, Fast Fourier Transform (FFT) [229] is adopted for period identification.
- **Normalisation**: For normalising the time series, the following equation is utilised:

$$\hat{x} = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{4.10}$$

  where $x \in X$ represents a value in time series X; $\hat{x}$ is the normalised value of x; $\min(x)$ and $\max(x)$ are the minimum and maximum values in the time series.
- **Vectorisation**: To detect complex anomalous time series patterns, the vectorisation of time series is required. In this work, two consecutive time series points are aggregated to form a 2-dimensional vector.

After the preprocessing operations, for each time series, the anomaly detection process can take two modes: 1) use the normal data in the first two periods of the time series to train a model that test all the other data for anomalies; 2) employ non-overlapping sliding windows to divide the time series into several testing parts and, for each window, use a part of the data before the window to train the model. One could consider the first mode as batch anomaly detection mode because after the first training the model is not updated. However, the second mode is considered as an online anomaly detection process due to the fact that the anomaly detection model is updated with the movement of the sliding window. Note that the data for training include the primary information, i.e., the time series, and the contextual information which is designed as the difference, i.e., δ, between the time series point at time t, i.e., x_t, and that at time $t-T$, i.e., x_{t-T}. T is the period of the given time series and $t > T$. In the experiments, the **online anomaly detection mode** is adopted. In addition, due to the presence of the noise in the time series, the discriminants, i.e., Eqs.(4.7)(4.8)(4.9), are adjusted with the additions of a negative constant on the right part of the equations.

Table 4.1 The parameters of using LPSVDD+ over Yahoo benchmarks

Parameters	**A2Benchmark**	**A3Benchmark**
Dimension of Data	2	2
Dimension of Context	2	2
Training Window Size	1.5 periods	4 periods
Testing Window Size	0.5 periods	2 periods
Kernel	Gaussian kernel	Gaussian kernel
Kernel Parameter	0.075	0.05
Hyperparameter λ	1	1
Relax Eq. (4.7)	- 0.3	- 0.27
Relax Eq. (4.8)	- 0.15	- 0.20
Relax Eq. (4.9)	- 0.25	- 0.19

4.4.3 Results

In this section, the general performance of LPSVDD+ is reported based on its anomaly detection results over Yahoo A2Benchmark and A3Benchmark datasets. It is emphasised that LPSVDD+ succeeds in accurately detecting point and contextual anomalies at the same time. And the detecting results shed light on the detailed reasons why an anomaly is reported. In general, the anomaly detection results of A2Benchmark demonstrates the overall capability of LPSVDD+ in anomaly detection, while that of A3Benchmark support the claim that LPSVDD+ is helpful in anomaly interpretation.

Yahoo A2Benchmark

For the experiments over all the time series in Yahoo A2Benchmark, a fixed set of parameters is utlised. As shown in Table 4.1, the dimensions of the data and the corresponding contexts are both fixed to 2, which means the inputs to LPSVDD+ are two series of 2-dimensional vectors. In the online anomaly detection mode, the non-overlapping testing windows divide a time series into several parts. To test the data and contexts in a particular testing window, the data in the training window, which is ahead of the testing window, are used to train LPSVDD+. The testing window is of the size of half a period of the target time series. While the training window contains the data of a whole period before the corresponding testing window. Concerning the kernel used in LPSVDD+, Gaussian kernel is selected with kernel parameter $\sigma = 0.075$. In practical utilisation of LPSVDD+, the hyperparameter λ in Eqs. (4.3)(4.4)(4.7) is set to 1 and, to negate the influence of noise in time series, the right hand side of Eqs. (4.7)(4.8)(4.9) are relaxed with the additions of three constants, i.e., -0.3, -0.15 and -0.25, respectively.

Table 4.2 The overall accuracy of using LPSVDD+ over Yahoo A2Benchmark

Anomaly Type	**True Positive (TP)**	**False Positive (FP)**	**False Negative (FN)**	**F1-score**
Overall	429	0	5	0.994
Point	406	16	28	0.948
Contextual	417	10	17	0.961

Note that, the datasets in Yahoo A2Benchmark are 100 synthetic time series that possess simple linear trend, clear periodic patterns and uncomplex anomaly types. Therefore, it is expected that most anomaly detection methods should perform nicely, e.g., near 100% precision and recall, over all the time series in A2Benchmark. Table 4.2 presents the results of exploiting LPSVDD+ over A2Benchmark datasets for anomaly detection. As is depicted, the overall performance of LPSVDD+ is satisfactory. Using Eq. (4.7), LPSVDD+ accurately detects 429 out of all the 434 anomalies with no false alarm. The discriminant of Eq. (4.9) detect 422 point anomalies, 16 of which are erroneous. While 427 contextual anomalies are identified with Eq. (4.8) and 10 of them are incorrect. It is worth stressing that all the 434 anomalies in the datasets are point anomalies and contextual anomalies at the same time. This is because all the anomalous data not only have anomalous values or patterns but also contain corresponding abnormal contexts. Either the correct identification of point anomalies or that of contextual anomalies will contribute to the eventual anomaly detection result, which has a F1-score of 0.994. Note again that, to calculate F1-score:

$$\begin{aligned} F_1 &= 2 \times \frac{precision \times recall}{precision + recall}, \\ precision &= \frac{TP}{TP+FP}, \quad recall = \frac{TP}{TP+FN}. \end{aligned} \tag{4.11}$$

In fact, if parameters are to be tuned for anomaly detection over each time series of Yahoo A2Benchmark, better results are possible. However, in this part, the purpose is to demonstrate the general capability of LPSVDD+. Thus, only a fixed parameter setting is employed. Next, some details concerning the unsatisfactory anomaly detection results are examined for pinpointing the reasons. As depicted in Fig.4.2, LPSVDD+ does not detect any anomaly in time series “syn54”, because these is no anomaly score that is below the thresholds, i.e., red dash lines in the first three subfigures. In this case, adjusting the thresholds accordingly will support the identification of the most likely anomalies. However, the adjustments will also lead to false positives which are contributed by the intrinsic noise within the time series. The time series is too noisy that some normal points, e.g., the points marked by the red arrow, can be considered as anomalous even by human judgments.

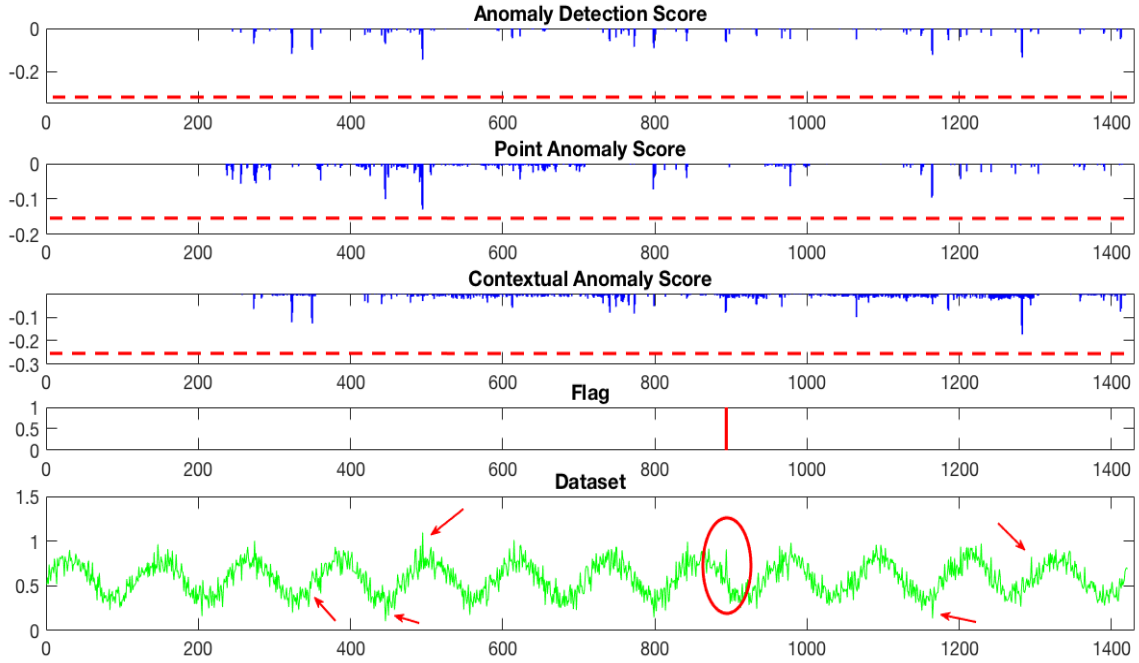


Fig. 4.2 The performance of LPSVDD+ over time series "syn54" in Yahoo A2Benchmark

Yahoo A3Benchmark

This subsection presents the results of the experiments conducted to evaluate the proposed approach over Yahoo A3Benchmark. The exact parameters are provided in Table 4.1. All the other settings are the same as those utilised in last subsection. Compared with A2Benchmark, datasets in A3Benchmark contain much more complex periodicity and time series patterns. The usual methods, which are used to eliminate the trendings and identify the periodicity of time series, may not work as expected. As a result, the performances of different anomaly detection methods could be largely degraded in A3Benchmark. Table 4.3 presents the overall accuracy of using LPSVDD+ over A3Benchmark. It is shown that, concerning the overall anomaly detection accuracy, the F1-score is 0.936 and the numbers of false positives and false negatives increase. The utilisation of discriminant Eq.(4.9) correctly identifies 774 point anomalies, while 766 contextual anomalies are pinpointed accurately through Eq.(4.8). The degraded F1-scores of point anomaly and contextual anomaly identification are largely due to the fact that not all the anomalies in A3Benchmark are point anomaly and contextual anomaly at the same time. Some anomalies are just anomalous in terms of data patterns, while their contexts are normal. On the other hand, some anomalies are normal in data patterns but have abnormal contexts. Because of the lack of labels of the types of anomalies, the time series "TS63" from A3Benchmark datasets is chosen to demonstrate the anomaly interpretation process of LPSVDD+.

Table 4.3 The overall accuracy of using LPSVDD+ over Yahoo A3Benchmark

Anomaly Type	**True Positive (TP)**	**False Positive (FP)**	**False Negative (FN)**	**F1-score**
Overall	858	49	68	0.936
Point	774	74	152	0.872
Contextual	766	94	160	0.857

Fig.4.3 presents the experimental results of using LPSVDD+ over "TS63". In the 5th subfigure on the bottom, the original time series is demonstrated with manually marked anomalies, which are also depicted in the 4th subfigure. From the 1st subfigure on the top, it is clear that LPSVDD+ detects all the anomalies without false alarms and miss alarms. The results in the 1st subfigure are obtained through checking Eq.(4.7), while the results in the 2nd and 3rd subfigures are generated with the discriminant functions for anomaly detection over primary information (Eq.(4.9)) and contextual information (Eq.(4.8)) respectively. Note that the 2nd subfigure also identifies all the anomalies, but further interpret them as point anomalies. This is because these anomalies show strange patterns, e.g., an abnormal combination of data instances or an abrupt spike. On the other hand, the results in the 3rd subfigure identify 3 parts of contextual anomalies, stressing that the abnormality of the corresponding data is also due to their anomalous contextual information, i.e., the abrupt increment of the data value over that in the last period. With the identification of the point anomalies and the contextual anomalies, the anomaly detection process provides more informative details about why a data is marked as anomalous. Consequently, one would be able to treat anomalies differently according to the additional information.

The experiments over all the other time series in Yahoo A3Benchmark obtain an average F1-score of 0.93 and also demonstrate similar results as that in Fig.4.3, which reflects the overall effectiveness of the proposed method in anomaly detection. More specifically, according to the experiment results, the proposed method is effective for distinguishing the contextual anomalies from typical point anomalies and, therefore, achieves better anomaly interpretation for practical anomaly detection systems, e.g., intrusion detection systems.

4.5 Conclusion

In this chapter, an anomaly detection method, which can distinguish different types of anomalies, is proposed for interpreting the anomalies. The method is based on integrating two LPSVDDs to support the training of two different types of information, e.g. primary information and contextual information. Experimental results on all the time series in Yahoo

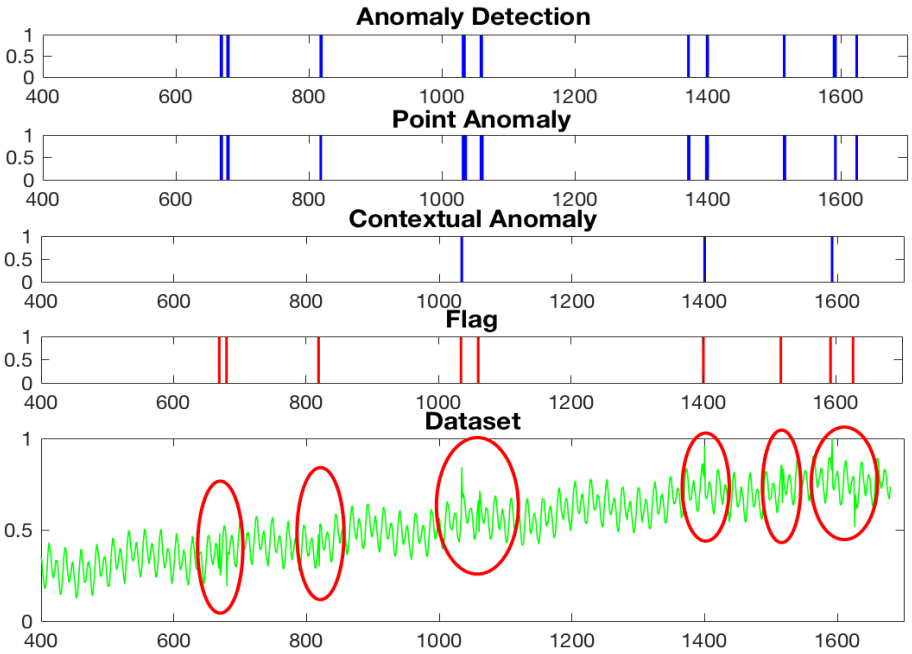


Fig. 4.3 The performance of LPSVDD+ over time series "TS63" in Yahoo A3Benchmark

A2Benchmark and A3Benchmark datasets demonstrate that the proposed method is of high accuracy and capable of identifying different anomalies, thus enabling better interpretation of the detected anomalies. As a result, the utilisation of the method in anomaly detection systems will largely benefit the underlying decision-making systems in choosing the proper reaction when an anomaly is witnessed.

Chapter 5

Convex Hull Data Description

5.1 Introduction

In Chapter 4, the proposed method succeeds in anomaly interpretation with the utilisation of two discriminants for the identification of two different types of anomalies, i.e., point anomaly and contextual anomaly. While, in many situations, the analysis of anomalies are much more complex that point anomalies or contextual anomalies are required to be further categorised for detailed analysis. This requirement desires the combination of the one-class classification technologies, e.g., [209][157][45][196], and the clustering methodologies, e.g., [82][221], especially in situations where no label is provided for anomaly classification. Nevertheless, for a long time, one-class classification and clustering are two research topics that are treated separately. Related projects that consist of these two tasks normally solve them separately with specific methods, which is comparatively complex and costs additional resources. Therefore, methods that could solve both of these tasks under a consistent framework are more appreciated and required to fill the research gap.

To this end, this chapter proposes Convex Hull Data Description (CHDD) to achieve one-class classification and clustering under a same problem formulation. Specifically, CHDD approximates the convex hull of a dataset by recognising the data representatives. The description of the dataset using the representatives not only results in the criteria for one-class classification but also reveals the internal relations among data instances which enable data clustering. Therefore, CHDD solves the tasks of one-class classification and clustering at the same time. To further summarise, this chapter makes the following contributions:

- A novel formulation of convex hull approximation is proposed and solved to find the approximated extreme points in §5.3.2 and §5.3.3. The utilised solver, i.e., Semi-

Nonnegative Matrix Factorisation (Semi-NMF), enables the kernel trick and is proved to guarantee the valid solutions under the utilisation of the Gaussian kernel.

- In §5.3.4, Convex Hull Data Description (CHDD) is developed to address general one-class classification and clustering tasks under the same problem formulation.

- The performance of CHDD is illustrated in one-class classification and clustering of a variety of datasets with distinct features in §5.4. It is shown that CHDD is promising in both of the tasks.

5.2 Related Work

5.2.1 Data Description

A significant related work of data description is the comprehensive toolbox provided by Tax [207] for the purpose of one-class classification. In the toolbox, five different classes of methods were implemented: 1) **statistical methods** that analyse the statistical properties of the dataset, e.g., Gaussian Mixture Model (GMM); 2) **distance-based methods** that regard distance as the most relevant factor in measuring data similarity, e.g., K-Nearest Neighbor (KNN); 3) **density-based methods** which leverage the density information of normal/abnormal data in a specific region for anomaly detection, e.g., Parzen density estimation; 4) **model-based methods**, typical examples of which include Self-organisation Map (SOM) and SVDD; and 5) **spectral analysis methods** that investigate the attributes of the space on which the data lie, for instance, Principal Component Analysis (PCA). Among all these well-developed one-class classification methods, only few methods, e.g., GMM and SOM, are also capable of clustering. However, the assumption of the data distribution made by GMM and the difficulty of initialising SOM make them not practical under some scenarios that more applicable methods are required. Moreover, although the up-to-date methods of one-class classification, such as the Binary Decision Diagram-based one-class classifier (BDD) [108], one-class classification with Gaussian Process (one-class GP) [119] and Isolation Forest (iForest) [144], and the recent methods of clustering, e.g., mst_clustering [221], Kernel K-means [82] and sparse subspace clustering [68], all exhibit their novelties and good performances in one-class classification and clustering respectively, all these methods cannot be immediately applied in both one-class classification and clustering problems. Therefore, new methods are required to fill the gap.

5.2.2 Convex Hull Analysis

Identifying the convex hull of a multivariate dataset has been a research topic for a long time. Over the years, geometricians and many other researchers have developed numerous related methods. The early idea of computing an approximated convex hull was proposed in 1982 when Bentley and Faust [26] introduced a set of algorithms for the problem. Since then, convex hull identification and approximation have both experienced speedy development due to their broad applications in data mining and machine learning [16][235]. Besides the conventional methods, most of which find the convex hull from a geometric perspective, a recent work [67] introduced the utilisation of spectral analysis for convex hull identification. The method leverages the definition of the convex hull and solves an optimisation problem to obtain a sparse coefficient matrix which encodes the identities of the extreme points. A related formulation was also adopted in [203], where the authors proposed a greedy search approach for selecting the extreme points according to some predefined constraints derived from the formulation. It is worth noting that the convex hulls of general datasets are normally not tight enough for data description. As a result, although both of these methods are elegant in convex hull identification, they are not practically applicable in data description tasks.

5.3 Convex Hull Data Description

5.3.1 Problem Formulation

Let $S = \{x_1, x_2, \cdots, x_N\}$ be a dataset consisting of N D-dimensional data instances $x_i \in \mathbb{R}^D$, $i \in \{1, 2, \cdots, N\}$. From the perspective of simple geometry, there must be some data instances in S that can be used as the representatives, e.g., extreme points, to describe any data instance in the dataset. According to the concept of the convex hull, the process is expressed as:

$$\begin{aligned} & x_i = X_{ep} c_i, \\ \forall i, \quad & c_i \geq 0, \\ & \mathbf{1}^T c_i = 1, \end{aligned} \tag{5.1}$$

where $X_{ep} \triangleq [x_{ep,1} \cdots x_{ep,n}]$ is a matrix that comprises n column vectors representing the extreme points, and $c_i \in \mathbb{R}^n$ denotes the coefficient vector that is employed in reconstructing x_i. Note that the $\mathbf{1}$ in this context is a n-dimensional column vector whose elements are all 1. Essentially, the formulation tries to describe x_i using a weighted linear combination of the extreme points.

In the task of convex hull identification or approximation, the essential problem is to determine the extreme points X_{ep}. A key observation of the problem is that the coefficient matrix $C \triangleq [c_1 \cdots c_N]$ encodes valuable information for the identification of the extreme points while considering the following formulation with $X \triangleq [x_1 \cdots x_N]$:

$$\begin{aligned} X &= XC, \\ C \geq 0, \quad \mathbf{1}^T C &= \mathbf{1}^T. \end{aligned} \tag{5.2}$$

For an extreme point x_j, its corresponding coefficient vector has to be of the form: $c_j = (0,\ldots,1,\ldots,0)^T$. The index of the position that 1 appears has to be j. In other words, the diagonal elements of C indicate whether their corresponding data instances are extreme points. As a concrete example, let us consider the situation where the coefficient matrix is calculated for 4 data instances according to Eq.(5.2):

$$\begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0.25 \\ 0 & 1 & 0 & 0.25 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.3}$$

As the coefficient matrix indicates, the first three data instances, i.e., $(0,0)^T$, $(2,0)^T$, and $(1,2)^T$ are extreme points, because it is not possible to find other linear combinations to reconstruct the data without using themselves. However, $(1,1)^T$ can be described by the other data, which reflects that it is not an extreme point.

In the next subsection, a novel method is proposed to find an approximated coefficient matrix for the identification of extreme points in general datasets, through which I develop Convex Hull Data Description.

5.3.2 Convex Hull Approximation

Based on the problem formulation, the essentials of CHDD rest on the convex hull approximation through solving an optimisation problem:

$$\begin{aligned} \min_{C} \quad & \|X - XC\|^2, \\ \text{s.t.} \quad & C \geq 0, \quad \mathbf{1}^T C = \mathbf{1}^T. \end{aligned} \tag{5.4}$$

This formulation is derived from Eq.(5.2). It relaxes the constraint of the equality, i.e., $X = XC$, and tries to describe the original dataset using itself in a best-effort way, i.e., minimising the squared differences between X and XC. To solve the formulation and

enable wider classes of applicable algorithms, the equality constraint in Eq.(5.4) is merged into the target function through appending a constant weight to each data instances. In other words, a data instance x_i is actually constructed as $x_i = [x_i^T \ \omega]^T$, where ω is the weight. Thus, the target becomes $\min_C \|[X^T \ \omega\mathbf{1}]^T - [X^T \ \omega\mathbf{1}]^T C\|^2$ which is equivalent to $\min_C \|X - XC\|^2 + \omega^2\|\mathbf{1}^T - \mathbf{1}^T C\|^2$. It appends $\min_C \omega^2\|\mathbf{1}^T - \mathbf{1}^T C\|^2$ to the original target function and realises the requirement of $\mathbf{1}^T C = \mathbf{1}^T$ with certain relaxation. Consequently, the actual optimisation problem is simpler:

$$\min_{C\geq 0} \quad \|X - XC\|^2. \tag{5.5}$$

As will be theoretically proved in Theorem 1, a good way to solve this problem is to regard it as a Semi-NMF problem [54] and adopt the corresponding slover, i.e., the multiplicative updating rule:

$$C^{k+1} = C^k \circ \sqrt{\frac{[X^T X]_+ + [X^T X]_- C^k}{[X^T X]_- + [X^T X]_+ C^k}}, \tag{5.6}$$

where C^k denotes the matrix C after the kth iteration of the updating. The notation $A \circ B$ and $\frac{A}{B}$ represent the element-wise multiplication and division between matrices A and B respectively. The operations $[\cdot]_+$ and $[\cdot]_-$ are defined as:

$$[A]_+ = \frac{A + |A|}{2}, \quad [A]_- = \frac{A - |A|}{2}, \tag{5.7}$$

where $|A|$ is a matrix consisting of all the absolut values of the elements in matrix A.

Theorem 1. *The multiplicative updating rule in Eq.(5.6) solves Eq.(5.5) with the guarantee that the solution c_j of an extreme point x_j is of the form $c_j = (0,\ldots,1,\ldots,0)^T$, where the 1 is the jth element.*

Proof. Without loss of generality, consider only the solution c_j of $x_j = Xc_j$ and $X \geq 0$. The updating rule is simplified as $c_{ij}^{k+1} = c_{ij}^k \cdot \sqrt{\frac{x_i^T x_j}{x_i^T X c_j^k}}$ because $[X^T X]_- = 0$ and c_{ij} denotes the ith element in the vector c_j. Semi-NMF is guaranteed to converge [54] and when it converges, i.e., $c_{ij}^{k+1} = c_{ij}^k$, it must suffice that $c_j \geq 0$ because $C \geq 0$ and $\sum_i c_{ij} = 1$ because of the minimisation of the target function $\min_C \omega^2\|\mathbf{1}^T - \mathbf{1}^T C\|^2$. Also, the convergence means $\forall i, \sqrt{\frac{x_i^T x_j}{x_i^T X c_j^k}} = 1$. Therefore,

$$x_j = Xc_j, \tag{5.8}$$

which leads to:

$$x_j(1-c_{jj}) = \sum_{i \neq j} x_i c_{ij}. \tag{5.9}$$

If x_j is an extreme point, it cannot be expressed by a convex combination of other data instances, i.e., x_i with $i \neq j$. Thus, $c_{jj} = 1$ and $c_{ij} = 0$ for $i \neq j$. □

5.3.3 Convex Hull Approximation with Gaussian Kernel

When utilising the algorithm of Semi-NMF for convex hull approximation, the updating rule only depends on the inner product of the original data matrix, i.e., $X^T X$. Therefore, the kernel trick can be employed to generalise the algorithm for the kernel convex hull approximation. In this paper, I focus on the use of Gaussian kernel $K(X,X) = \phi(X)^T \phi(X)$, whose elements are defined as:

$$\forall i, j, \quad K(x_i, x_j) = \phi(x_i)^T \phi(x_j) = \exp^{\frac{-\|x_i - x_j\|^2}{\sigma^2}}, \tag{5.10}$$

where $\phi(\cdot)$ denotes the projection function. Hence, the optimisation problem in Eq.(5.5) is changed as:

$$\min_{C \geq 0} \quad \|\phi(X) - \phi(X)C\|^2. \tag{5.11}$$

Due to the reasons that $K(x,x) = 1$ and $K(x,y) \geq 0$, the updating rule of Semi-NMF algorithm can be modified accordingly:

$$C^{k+1} = C^k \circ \sqrt{\frac{K(X,X)}{K(X,X)C^k}}. \tag{5.12}$$

Note that this Semi-NMF updating rule under the utilisation of Gaussian kernel has roughly the same form as that in NMF [141]. With a further analysis of the algorithm, it is noted that for each element c_{ij} in C, the updating rule works as:

$$c_{ij}^{k+1} = c_{ij}^k \cdot \sqrt{\frac{\phi(x_i)^T \phi(x_j)}{\phi(x_i)^T \phi(X) c_j^k}}. \tag{5.13}$$

Theorem 2. *The multiplicative updating rule in Eq.(5.13) solves Eq.(5.11) with the guarantees that the solution c_j of an extreme point $\phi(x_j)$ is of the form $c_j = (0,\ldots,1,\ldots,0)^T$, where the 1 is the jth element, and the solution of an non-extreme point is of a different form, i.e., $c_{jj} \neq 1$, with the assumption that $\|\phi(X)c_j\| \leq 1$ always holds.*

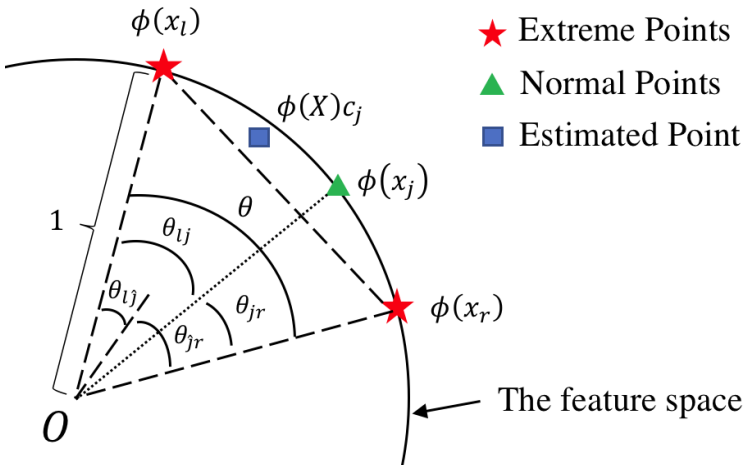


Fig. 5.1 Gaussian kernel space

Proof. It is straightforward from Theorem 1 that, for an extreme point $\phi(x_j)$, $c_{jj} = 1$ and $c_{ij} = 0$ for $i \neq j$. Therefore, the first part of Theorem 2 is proved. For a normal point $\phi(x_j)$ and its estimation $\phi(X)c_j$, consider two known extreme points $\phi(x_l)$ and $\phi(x_r)$ in the Gaussian kernel space (in Gaussian kernel space the boundary points are considered as extreme points, see Fig.5.1) and $\|\phi(X)c_j\| \leq 1$, it holds that if $\frac{\phi(x_l)^T\phi(x_j)}{\phi(x_l)^T\phi(X)c_j} < 1$:

$$\begin{aligned}
& \|\phi(x_l)\|\|\phi(x_j)\|\cos\theta_{lj} < \|\phi(x_l)\|\|\phi(X)c_j\|\cos\theta_{l\hat{j}} \\
\Rightarrow\quad & \cos\theta_{lj} < \cos\theta_{l\hat{j}} \\
\Rightarrow\quad & \cos\theta - \theta_{lj} > \cos\theta - \theta_{l\hat{j}} \\
\Rightarrow\quad & \cos\theta_{jr} > \cos\theta_{\hat{j}r} \\
\Rightarrow\quad & \|\phi(x_r)\|\|\phi(x_j)\|\cos\theta_{jr} > \|\phi(x_r)\|\|\phi(X)c_j\|\cos\theta_{\hat{j}r},
\end{aligned} \tag{5.14}$$

then it suffices that $\frac{\phi(x_r)^T\phi(x_j)}{\phi(x_r)^T\phi(X)c_j} > 1$. The property, i.e., $\|\phi(x)\| = 1$, of the Gaussian kernel is used above. And $\theta = \theta_{lj} + \theta_{jr} = \theta_{l\hat{j}} + \theta_{\hat{j}r}$, where $\theta_{lj}, \theta_{jr}, \theta_{l\hat{j}}, \theta_{\hat{j}r}$ and θ indicate the acute angles between the vectors in the pairs $\big(\phi(x_l), \phi(x_j)\big)$, $\big(\phi(x_j), \phi(x_r)\big)$, $\big(\phi(x_l), \phi(X)c_j\big)$, $\big(\phi(Xc_j), \phi(x_r)\big)$ and $\big(\phi(x_l), \phi(x_r)\big)$, respectively. Hence, the above inference shows that, to construct a normal data instance $\phi(x_j)$, the weights contributed by the extreme points will not all reduce to 0, because when c_{lj} decreases, i.e., $\sqrt{\frac{\phi(x_l)^T\phi(x_j)}{\phi(x_l)^T\phi(X)c_j}} < 1$, c_{rj} will guarantee to increase because of $\sqrt{\frac{\phi(x_r)^T\phi(x_j)}{\phi(x_r)^T\phi(X)c_j}} > 1$. That means, there is at least one weight c_{lj} or c_{rj} that is not 0 when Semi-NMF converges. In other words, $\sum_{i\neq j} c_{ij} \neq 0$ and $c_{jj} \neq 1$. □

Algorithm 1 (Kernel) Convex Hull Approximation

Input:

The dataset of D-dimensional N data instances, X;
The parameter of Gaussian kernel, σ;
The expected number of extreme points, n;
The convergence criteria;

Output:

The approximated extreme points, X_{ep};

1: initialise C, where $c_j = \frac{1}{N} \cdot \mathbf{1}$ and $j \in \{1, 2, \cdots, N\}$;
2: append $\omega = \sqrt{D}$ to each data instance in X;
3: set $K = K(X, X)$ using Gaussian kernel with σ;
4: **repeat**
5: set $C = C \circ \sqrt{\frac{K}{KC}}$;
6: **until** convergence criteria are met
7: set X_{ep} as the set (matrix) of n data points whose values in $diag(C)$ are among the top n.
8: **return** X_{ep}.

Comments: to proof the theorem, the assumption is made that $\forall c_j, \|\phi(X)c_j\| \leq 1$. In other words, if $C^T\phi(X)^T\phi(X)C = C^TK(X,X)C \leq 1$ holds during the optimisation process, it is guaranteed that Semi-NMF can successfully identify the extreme points of the target dataset. In practice, the elements in C are initialised to be all $\frac{1}{N}$ and it is optimistic that $C^TK(X,X)C \leq 1$ always hold as $\phi(X)c_j$ moves slowly to $\phi(x_j)$.

Therefore, it is summarised that the multiplicative updating rule of Semi-NMF achieves convex hull approximation through identifying the extreme points based on the importance of a data instance in describing itself, which is reflected by the diagonal elements in C, i.e., $diag(C)$. The algorithm for identifying the extreme points of a dataset is formally presented in Algorithm 1. In practice, convex hull approximation is adopted rather than identification due to the reason that Semi-NMF takes too long to converge and approximated convex hull are sufficiently useful in related tasks (Section 5.4).

Practically, the initialisation of C in Algorithm 1 is to set every element in C to $\frac{1}{N}$, i.e., $c_j = \frac{1}{N} \cdot \mathbf{1}$. For the assignment of the weight ω (see Section 5.3.2), I empirically adopt $\omega = \sqrt{D}$ to make sure that the constraint $\mathbf{1}^TC = \mathbf{1}^T$ will not be neglected when the dimension of the data is high. Concerning the convergence criteria, the standard stopping criteria of NMF is utilised, i.e., whenever the maximum change of the elements in C or that of the value of the target function in Eq.(5.5) is below a certain threshold, the updating stops. These settings are utilised in all our experiments for convex hull approximation, which is the basic building block of the Convex Hull Data Description.

Algorithm 2 Reconstruction Coefficient and Error

Input:

The target dataset, X;
The source dataset, X_{ep};
The parameter of Gaussian kernel, σ;
The convergence criteria;

Output:

The reconstruction coefficient, C_{ep};
The reconstruction error, E;

initialise C_{ep};
append a weight ω to each data instance in X and X_{ep};
set $K_1 = K(X,X)$ using Gaussian kernel with σ;
set $K_2 = K(X,X_{ep})$ using Gaussian kernel with σ;
set $K_3 = K(X_{ep},X_{ep})$ using Gaussian kernel with σ;
repeat
    set $C_{ep} = C_{ep} \circ \sqrt{\frac{K_2}{K_3 C_{ep}}}$;
until convergence criteria are met
set $E = diag(K_1) - 2 \cdot diag(K_2 C_{ep}) + diag(C_{ep}^T K_3 C_{ep})$.
return C_{ep} and E.

5.3.4 Convex Hull Data Description (CHDD)

Relying on convex hull approximation, CHDD is to extract the key features of a dataset in order to **(I)** determine whether a new data instance belongs to the dataset, i.e., one-class classification, and **(II)** separate the different groups in the dataset to form clusters, i.e., clustering. To this end, the whole dataset is once again described by only the approximated extreme points using the same optimisation formulation:

$$\min_{C_{ep} \geq 0} \|\phi(X) - \phi(X_{ep})C_{ep}\|^2, \tag{5.15}$$

where X_{ep} is a matrix obtained by running Algorithm 1 over the original dataset X and contains the approximated extreme points, while C_{ep} is the new coefficient matrix.

One-class Classification

To achieve one-class classification, the reconstructed error of each data instance is exploited as the key feature. And it is defined that:

$$\varepsilon = \max_i \|\phi(x_i) - \phi(X_{ep})c_{ep,i}\|^2, \tag{5.16}$$

Algorithm 3 Convex Hull One-class Classification

Input:

The training dataset, X_{trn};
The testing dataset, X_{tst};
The estimated number of extreme points, n;
The parameter of Gaussian kernel, σ;
The convergence criteria;

Output:

The reconstruction errors of X_{tst}, E_{tst};
The threshold, ε;

1: run Algorithm 1 with X_{trn}, n, σ and the convergence criteria: extract approximated extreme points X_{ep};
2: run Algorithm 2 with X_{trn}, X_{ep}, σ and the convergence criteria: obtain the threshold $\varepsilon = \max E$;
3: run Algorithm 2 with X_{tst}, X_{ep}, σ and the convergence criteria: measure the anomality of the data instances in X_{tst} using $E_{tst} = E$;
4: **return** ε and E_{tst}.

which is the upper threshold of all the reconstructed errors of the original data instances. Note that, $c_{ep,i}$ is the ith column vector in C_{ep}. Hence, any new data instance whose reconstructed error exceeds the threshold is considered to be a novelty. Formally, for a new data instance x_{new}, it is regarded as a member of the original dataset if it satifies:

$$\min_{c_{new}} \|\phi(x_{new}) - \phi(X_{ep})c_{new}\|^2 \leq \varepsilon. \tag{5.17}$$

The algorithms used to solve Eqs.(5.15) - (5.17) are essentially the same and formally given in Algorithm 2. When the target and source datasets are X and X_{ep}, respectively, the output E maintains the reconstruction errors for all the original data. Therefore, ε is the maximum element in E. While the target and source datasets are X_{new} and X_{ep}, respectively, the output E maintains the reconstruction errors of all the new data instances. A data instance with $e \in E$ and $e > \varepsilon$ is considered as an anomaly. It is summarised that the process of one-class classification in CHDD follows three key steps as shown in Algorithm 3. The detailed process starts with the utilisation of Algorithm 1 to obtain the approximated extreme points. Afterward, Algorithm 2 is followed in order to extract the feature of the training dataset, i.e., the reconstruction errors of all the known data instances, assuming that they are within the same class. The maximum reconstruction error is chosen as the threshold for identifying anomalies. The second run of Algorithm 2 with a testing dataset will reveal the construction errors of the data in the testing dataset. Therefore, a further comparison between the errors and the threshold determines the result of one-class classification.

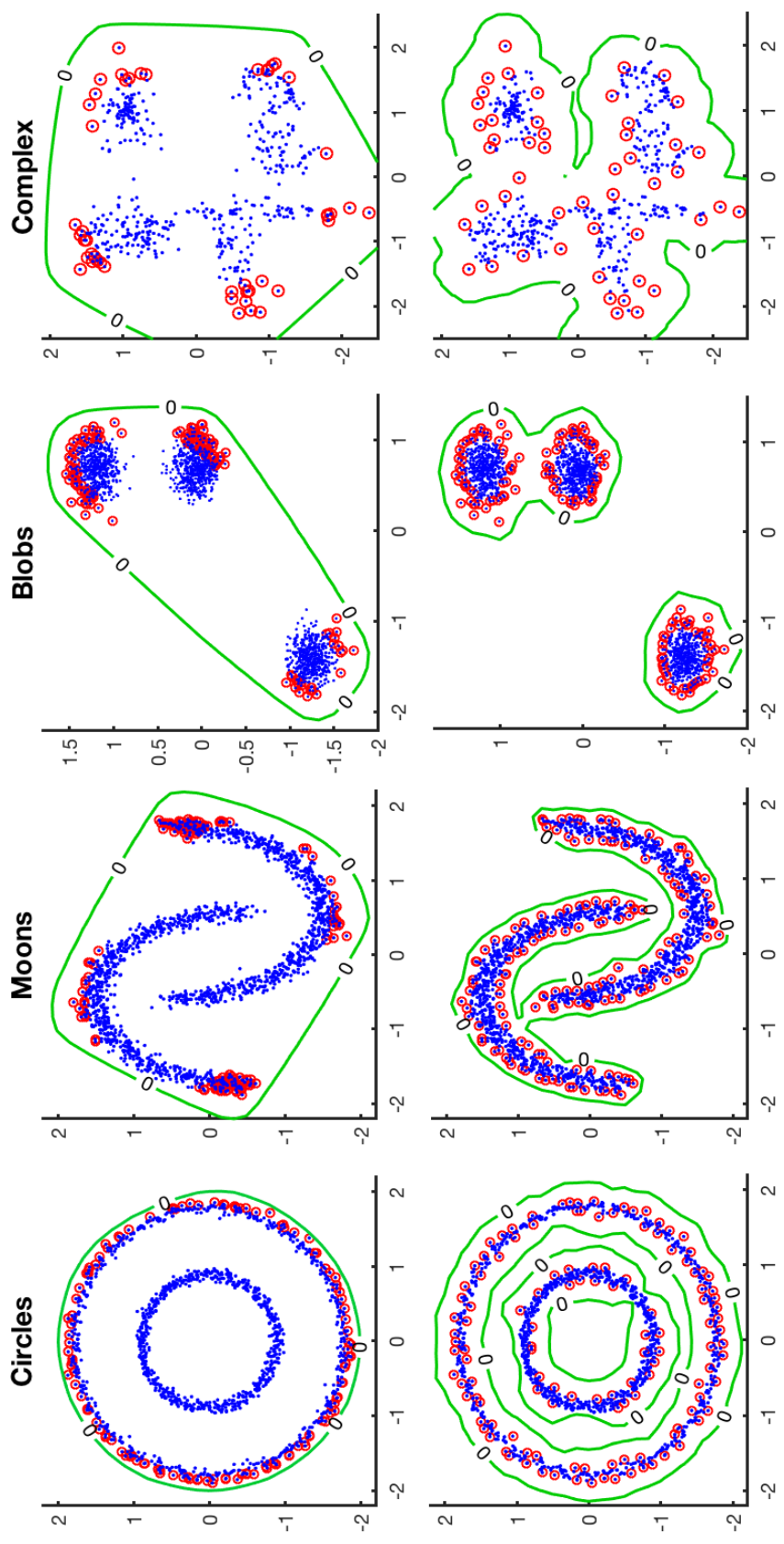


Fig. 5.2 The performance of convex hull one-class classification in four toy datasets. ($n = 0.1 \times N$, Gaussian kernel $\sigma = 0.3$, better view in color)

In Fig.5.2, the performances and characteristics of convex hull one-class classification in four toy datasets are displayed. The blue dots are the original data instances from the datasets whereas the red circles emphasise the approximated extreme points. The green boundaries are the decision boundaries for anomaly detection. Data instances outside the boundaries will be detected as anomalies, while normal data instances should rest inside the boundaries. The four subfigures on the top of Fig.5.2 firstly demonstrate convex hull one-class classification without using the kernel trick. The four subfigures on the bottom exhibit the effects of the Gaussian kernel with $\sigma = 0.3$. And the expected number of extreme points in all the tasks are selected as $n = 0.1 \times N$. It is apparent that the employment of Gaussian kernel significantly enhances the performance of convex hull one-class classification and generalises the method to be applicable in various datasets.

Clustering

Besides one-class classification, another application of convex hull approximation, i.e., clustering, is made possible with a careful examination of the reconstruction coefficient matrix C_{ep} after the resolution of Eq.(5.15). Theoretically, C_{ep} encodes the coefficients of the extreme points for constructing the original dataset. Each column of C_{ep} holds the coefficients for constructing a specific data, while each row of C_{ep} reveals how much the data instances are dependent on the corresponding extreme point. According to the definition of the convex hull, data instances that profoundly rely on a same extreme point are expected to be in the same cluster. Therefore, based on this intuition, a thorough investigation of C_{ep} can identify clusters of the original dataset. This process is called convex hull clustering. The details of convex hull clustering are given in Algorithm 4. Specifically, after the executions of Algorithm 1 and 2, convex hull clustering examines each row of C_{ep} to identify new clusters or integrated known clusters that have intense connections. The intensity of the connections is governed by a threshold ε. It is noted that the first two steps of convex hull clustering are exactly the same as that in convex hull one-class classification. The actual work that forms the clusters is inside the loop, which goes through the rows of C_{ep}.

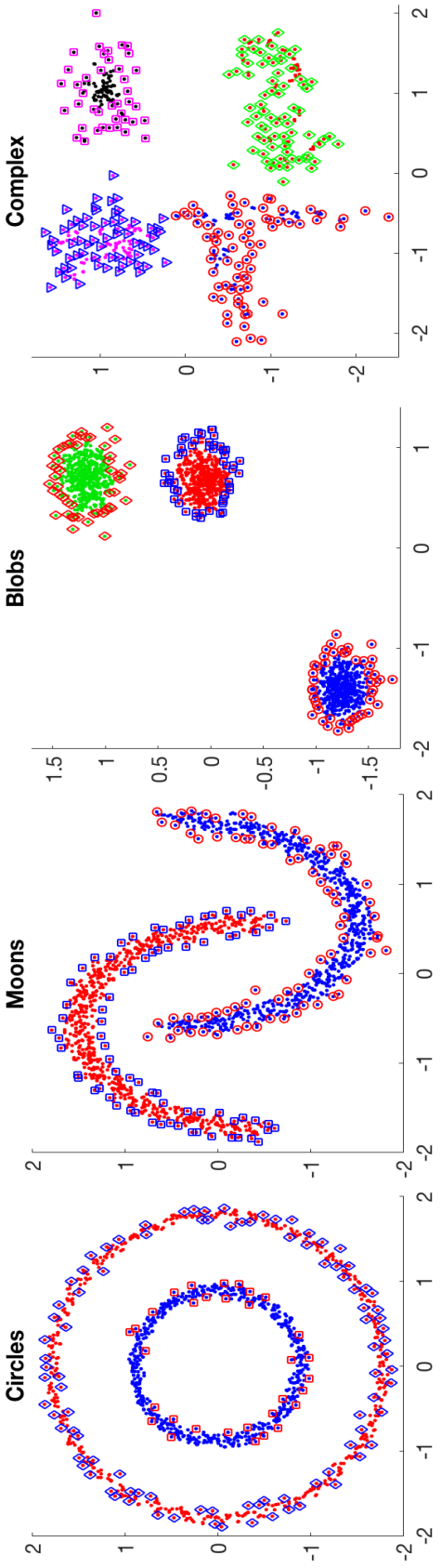


Fig. 5.3 The performance of convex hull clustering in four toy datasets. (The first three subfigures use $n = 0.1 \times N$ and Gaussian kernel $\sigma = 0.2$, while the last subfigure uses $n = 0.5 \times N$ and Guassian kernel $\sigma = 0.18$, better view in color)

Algorithm 4 Convex Hull Clustering

Input:

The target dataset, X;
The estimated number of extreme points, n;
The parameter of Gaussian kernel, σ;
The convergence criteria;
The threshold of data relationship, ε;

Output:

The clustering label, L;

1: run Algorithm 1 with X, n, σ and the convergence criteria: extract approximated extreme points X_{ep};
2: run Algorithm 2 with X, X_{ep}, σ and the convergence criteria: obtain the coefficient C_{ep};
3: initialise L, set the label of all data to 0;
4: **for** each row in C_{ep} **do**
5: get the label set L_ε of the elements in the row whose value is greater than ε;
6: **if** $\max L_\varepsilon == 0$ **then**
7: $\forall l \in L_\varepsilon$, set l to a new label;
8: **else**
9: $\forall l \in L$ whose value are in L_ε, set $l = \max L_\varepsilon$;
10: **end if**
11: **end for**
12: **return** L.

The performance of convex hull clustering in the same four toy datasets used in one-class classification is demonstrated in Fig.5.3. The data instances are shown as dots and assigned different colors according to the clusters they belong to. The circles, squares, diamonds and triangles are emphasising the approximated extreme points in the corresponding clusters. It is shown that it correctly identifies all the clusters of the toy datasets with the utilisation of the Gaussian kernel and proper tuning of the parameters. In some situations, due to the fact that CHDD chooses a set of approximated extreme points rather than the real ones, convex hull clustering may generate clusters that have few data instances. These data instances are normally outliers. In this paper, it is assumed that the training dataset does not contain outliers. Therefore, in our experiments, all the clusters with few data are merged into their most related clusters.

Table 5.1 The datasets for one-class classification

Name	#Dim	#Class	#Target	#Outlier
iris	4	3	50	100
wine	13	3	59	119
breast	9	2	458	241
car	6	4	384	1344
biomed	5	2	67	127
diabetes	8	2	268	500
sonar	60	2	111	97
breastdiag	30	2	357	212
glass	9	4	70	214
liver	6	2	145	345
ionosphere	34	2	225	351
imox	8	4	48	192
auto_mpg	6	2	229	398
chromo	8	24	42	1143
ecoli	7	8	143	336

5.4 Experiment Results

In this section, experiment results[1] are presented to demonstrate the capability of Convex Hull Data Description (CHDD). The experiments are arranged in two parts: 1) the effectiveness of CHDD in one-class classification and 2) The effectiveness of CHDD in clustering. Note that in one-class classification and clustering all the CHDD experiments use the Gaussian kernel for convex hull approximation. The tunable parameters are the parameter σ of the Gaussian kernel, the estimated number of the approximated extreme points n, the convergence criteria and the threshold ε for data clustering. All the parameters of CHDD and other compared methods are selected by grid search from a set of candidates, which will be specified later, to get the best possible results in the tasks.

5.4.1 One-class Classification

Datasets and Methods

To examine the effectiveness of CHDD in one-class classification, 15 different datasets from the UCI data repository [123] are tested. The details of the selected datasets are illustrated in Table 5.1. For each dataset, the data are firstly normalised and then the first class is picked as the target class. All the data in other classes are considered as anomalies.

[1]The source codes are available in https://github.com/ chengqianghuang/convex-hull-data-description.

Table 5.2 The results (AUC) of one-class classification in UCI datasets

	CHDD	SVDD	GMM	Parzen	PCA	K-means	KNN	LOF	iForest
iris	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
wine	**0.99**	**0.99**	**0.99**	**0.99**	0.98	**0.99**	**0.99**	**0.99**	0.98
breast	**0.99**	**0.99**	0.98	**0.99**	0.98	**0.99**	**0.99**	0.70	**0.99**
car	**0.99**	**0.99**	0.98	0.98	0.92	0.96	**0.99**	0.97	0.58
biomed	**0.73**	0.71	0.65	0.70	0.65	0.71	0.56	0.62	0.59
diab.	**0.72**	0.63	0.53	0.53	0.55	0.55	0.46	0.54	0.56
sonar	0.73	0.73	0.71	0.65	0.66	0.68	0.72	**0.81**	0.65
breast.	0.92	0.94	0.93	0.90	0.94	0.92	0.90	0.93	**0.96**
glass	0.81	0.82	0.83	0.72	0.80	0.79	0.81	**0.87**	0.82
liver	0.55	**0.60**	**0.60**	0.59	**0.60**	0.58	**0.60**	0.59	**0.60**
iono.	0.97	0.97	0.96	0.89	**0.98**	0.95	0.97	0.94	0.92
imox	0.91	0.97	**0.98**	0.96	0.88	0.94	0.97	0.90	0.92
auto.	0.84	0.88	0.92	0.92	0.81	0.91	**0.93**	0.64	0.92
chromo	0.95	**0.96**	0.94	**0.96**	0.94	0.95	0.95	0.95	0.95
ecoli	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**

In each experiment, 90% of the data instances are randomly sampled from the target class for model training. The remaining 10% of the target data and all the anomalies are used as the testing dataset. The detailed list of the tested methods is provided in Table 5.2. Apart from CHDD, 8 different methods from different categories are tested: (1) Support Vector Data Description (SVDD); (2) Gaussian Mixture Model (GMM); (3) Parzen-Window Density Estimation (Parzen); (4) Principal Component Analysis (PCA); (5) K-means; (6) K-Nearest Neighbour (KNN); (7) Local Outlier Factor (LOF); and (8) Isolation Forest (iForest). In each of the methods, there is one or several parameters. A set of candidates is prepared for each of the parameters and a grid search is employed to find the best model parameter for each method. During the grid search, for each parameter setting, a 5-folds cross-validation is performed upon the training dataset to gain the average performance. For the methods that use kernels, e.g., CHDD and SVDD, I employ Gaussian kernel and adopt the same parameter $\sigma^2 \in [0.1, 0.3, \cdots, 1.9] \cdot D$, where D indicates the dimension of the training dataset. The parameter σ^2 is also adopted as the width parameter in Parzen. For GMM, K-means, KNN and LOF, $k \in [1, 2, \cdots, 10]$ is utilised as the number of clusters or the number of neighbors. PCA picks the number of primary component from the set $c \in [1, 2, \cdots, \sqrt{D}]$. While, for iForest, the default values in the Python scikit-learn library are adopted. The rejection fraction (outlier fraction in training dataset) of all the methods are set to 0 because the training datasets do not contain outliers. Additionally, CHDD considers the fraction of the extreme points from the set $n \in [0.05, 0.1, \cdots, 0.5]$ and uses the same convergence criteria as K-means.

Results

The performances of all the one-class classification methods in the selected UCI datasets are presented in Table 5.2. Each result is the average performance of AUC, i.e. area under the curve, over 10 runs of the corresponding method. It is indicated by the results that there is no "best" method in one-class classification because no method outperforms all the other methods in all the datasets. For the datasets of "iris", "wine", "breast", "car" and "ecoli", CHDD is among the methods that demonstrate the best performance. For the datasets of "biomed" and "diabetes", CHDD outperforms all the other methods. While, in "sonar", "ionosphere" and "chromo", CHDD ranks the 2nd place in the performance. For the other datasets, CHDD ranks 3rd to 5th places. Generally speaking, CHDD is competitive in the effectiveness of one-class classification compared to other methods. From the perspective of time efficiency, the computation complexity of Semi-NMF dominates the overall computation complexity of CHDD. Due to the fact that Semi-NMF updates the entire coefficient matrix in CHDD, the computation complexity is $O(N^3)$ in each iteration, where N is the size of the original dataset.

5.4.2 Clustering

Datasets and Methods

Similar to one-class classification, 7 datasets are selected for the purpose of measuring the effectiveness of CHDD in clustering tasks. The datasets are from UCI [123] and Comprehensive R Archive Network (CRAN) repositories [27]. The first two rows in Table 5.3 present the names of the datasets as well as their basic information, i.e., the number and dimension of the data instances and the number of classes in each dataset. The compared methods contain Kernel K-means [82], DBSCAN, and the mst_clustering[221] method. All these methods are experimented after the normalisation of the datasets. A number of parameter settings are tested for each method to find the best clustering results, which are measured by Adjusted Mutual Information (AMI) [223].

To be more specific, CHDD selects the fraction of the extreme points from the set $[0.05, 1, \cdots, 1]$ and adopts Gaussian kernel with $\sigma^2 \in [0.1, 0.3, \cdots, 1.9] \cdot D \cup [0.05, 0.1, \cdots, 1]$, where D is the dimension of the dataset. Kernel K-means has the same kernel setting and sets the number of clusters $k \in [1, 2, \cdots, 15]$. DBSCAN picks the distance parameter from $[0.05, 0.1, \cdots, 1]$ and the minimum number of data points from the set $[2, 3, 7, 10, 15, 20, 25, 50, 75]$. And, linkage clustering utilises default distance settings and pick its cutoff parameter and neighbor parameter also from $[0.05, 0.1, \cdots, 1]$ and $[2, 3, \cdots, 75]$ respectively.

Table 5.3 The results (AMI) of clustering in UCI and CRAN datasets

(num,dim,#class)	**Motor** (94,3,3)	**Prest.** (98,5,3)	**Maps** (429,3,10)	**Image** (210,19,7)	**Pen** (1000,16,10)	**DrivFaceD** (606,6400,3)	**Libras** (360,90,15)
CHDD	**1**	**0.55**	**0.84**	0.57	0.62	**0.44**	0.53
Kernel K-means	0.70	0.53	0.76	**0.69**	**0.76**	0.08	**0.64**
DBSCAN	**1**	0.52	0.82	0.52	0.53	0.00	0.20
MST Clustering	**1**	0.51	0.81	0.52	0.64	0.10	0.56

Results

As indicated in Table 5.3, the clustering performance of CHDD is among the best in the dataset "Motor" and it outperforms all the compared methods in 3 datasets, i.e., "Prest.", "Maps" and "DrivFaceD". For datasets "Image", "Pen" and "Libra", Kernel K-means shows outstanding performance. The performances of CHDD rank the second or third places for these three datasets. To conclude with, the results demonstrate that CHDD has a good capability in data clustering. On the other hand, with the solution of the coefficient matrix, the clustering result of CHDD can be realised through a single scan over the coefficient matrix which is of size $N * n$, where N is the size of the dataset, n is the number of extreme points and $n << N$. Thus, the computation complexity of clustering is $O(N * n)$.

5.5 Conclusion

In this chapter, a novel method called Convex Hull Data Description (CHDD) is proposed. Three aspects of CHDD, i.e., convex hull approximation, one-class classification and clustering are elaborated. The convex hull approximation is innovatively achieved through Semi-Nonnegative Matrix Factorisation (Semi-NMF), which enables the utilisation of the kernel trick to support one-class classification and clustering. Our experiment results have demonstrated that CHDD successfully pinpoints the approximated extreme points and with the Gaussian kernel it is highly competitive in both one-class classification and clustering tasks in terms of the effectiveness. Consequently, CHDD is considered promising in anomaly detection tasks where further anomaly categorisation is desired.

Chapter 6

Towards Experienced Anomaly Detector with Reinforcement Learning

6.1 Introduction

Anomaly detection is a pervasive topic in various fields. In industry, it always serves as the first messenger to trigger more complicated procedures such as anomaly localisation. As a result, anomaly detection is very significant and, ideally, it should be highly applicable to different scenarios and easily accessible by engineers. However, existing anomaly detection methods, including the ones proposed in the previous chapters, do not necessarily satisfy the requirements. **1)** A thorough survey of anomaly detection methods is nicely presented in [29]. It clarifies the assumptions made by different types of anomaly detection methods, which reveals that methods with strong assumptions of the anomaly patterns, e.g., distribution-based methods, may not produce satisfactory results under scenarios where the assumptions do not hold. **2)** On the other hand, the anomaly detection methods are not always easily accessible. In 2015, Yahoo published their time series anomaly detection system EGADS [125]. Within the system, a set of methods are implemented and integrated to generate anomaly detection results. Such a complex system requires the engineers to not only understand the components but also comprehend the set of methods so that being able to tune the parameters for each of them. **3)** Additionally, few methods used in the industry consider the evolvement of the anomaly patterns, which leads to static anomaly detection parameters that perform poorly under dynamic scenarios. In this chapter, I consider the specific problem of time series anomaly detection and emphasise that a satisfactory anomaly detector should have the following features:

- The anomaly detector makes **no assumption about the concept of the anomaly**, i.e., the definition of the anomaly, but it learns the concept solely from the training datasets;
- The anomaly detector is **threshold-free**, which means the anomaly detector is a logical classifier with no tuning threshold. Preferably, except hyperparameters, e.g., the number of layers in a neural network, the detector does not have other tunable parameters.
- The anomaly detector is **dynamically improving** with the accumulation of the anomaly detection experience. In other words, the detector learns new anomalies and consistently enhances its knowledge for anomaly detection.

Although a large number of anomaly detectors have been proposed in related works, previous chapters show that few methods have accomplished all the expected features. Due to this fact, a new problem formulation is proposed to change the problem of time series anomaly detection to the problem of sequential decision making in an attempt to meet the criteria. Related details will be provided in the following sections with further discussions and extensive experiments.

6.2 Related Work

In typical anomaly detection problems, a group of data instances with no label information is provided for direct outlier detection or building a normal behavior model in order to achieve novelty detection. In either case, no clear information is witnessed to support the differentiation of the concept of normality and abnormality. Consequently, assumptions are made by various methods [29], which results in distinct implementations of anomaly detection methods. On the other hand, due to the vague distinction between the concepts, thresholds are required and tuned for practical fulfillment of an anomaly detector. These two issues are fundamental in designing accurate anomaly detectors in various domains. However, it seems that they could not be essentially solved without label information. Consider the situation that labels are supplied to clearly mark the normal and abnormal data instances, assumptions and thresholds could be inferred from the label information thus negating the needs of assumption making and threshold tuning. With label information, the problem of designing an effective anomaly identifier is converted into the problem of training an accurate anomaly classifier which is much easier to solve. Nevertheless, the implementation of an anomaly classifier is greatly hindered due to the lack of label information in diverse domains. Luckily, owing to recent research outcomes in data generation, e.g., GAN [84], the strategy of composing labeled data instances for anomaly classification is made possible.

Besides the identifications of assumptions and thresholds, the dynamic evolvement of the concept of normality is another critical issue that has a profound influence on accurate detection of anomalies in an application. Incremental methods, such as incremental SVDD [128] and sequential Bayesian learning [14], have been devised to efficiently process the evolvement of different models in diverse tasks. In anomaly detection, however, relatively few research work has been conducted to achieve model evolvement. One general framework that naturally possesses the concept of model updating is Reinforcement Learning (RL) [243] which has been applied in some application domains closely related to anomaly detection. In [195], RL is utilised for detection and categorisation of Distributed Denial of Service (DDoS) attacks. While in [28] the concept of RL is adopted to promote the training of a neural network in order to rapidly learn new network attacks in network intrusion detection. RL is also used in adaptive learning of parameters for sequential anomaly detection [242]. All these works are attempts at using RL to solve practical problems concerning anomaly detection. However they do not target general time series data anomaly detection which is the key topic in this chapter.

6.3 Problem Formulation

Time series anomaly detection is a sequential decision-making process. To concretely understand this, let us consider that an anomaly is reported/detected at a time step t. This action of anomaly detection changes the environment by stating that an anomaly happened in time t. The anomaly detection in the following time steps has to take the environment into consideration and performs appropriate actions according to system preferences, e.g., reporting duplicated anomalies or reporting only the first anomaly within a period of time. In other words, the environment encompasses not only the target time series but also the previous anomaly detection actions. Therefore, this decision-making process is sequential and naturally fit into the framework of RL. This opens a novel path to solving time series anomaly detection.

More specifically, in this chapter, a Recurrent Neural Network (RNN) based anomaly detector is proposed that it is trained consistently through RL to meet the objectives mentioned in Section 6.1. Following the framework of RL, the problem of time series anomaly detection is casted as a Markov Decision Process (MDP) [193] and then the corresponding concepts are defined mathematically.

Definition 1: Anomaly Detector π

An anomaly detector is defined as a conditional probability distribution:

$$\pi := p(A|S), \tag{6.1}$$

where S and A denote the sets of states and the set of actions in the target system respectively. Typically, $A = \{0, 1\}$ in which 1 means the given state is anomalous and 0 otherwise. And note that $\pi(s,a) = p(A = a|S = s)$ is the probability of taking action a given state s.

Definition 2: Anomaly Detector Performance V_π

The performance of an anomaly detector is measured through its capability of time series anomaly detection, which is formalised as:

$$V_\pi = \sum_{s \in S} d^\pi(s) \sum_{a \in A} Q(s,a) \cdot \pi(s,a), \tag{6.2}$$

where $d^\pi(s)$ is the probability of the target system being in the state s under the utilisation of the anomaly detector π, and $Q(s,a)$ represents the accumulated reward started from state s with action a. In other words, the performance is the average accumulated reward in anomaly detection following the anomaly detector π.

Definition 3: Optimal Anomaly Detector π^*

The optimal anomaly detector is the detector that satisfies:

$$\pi^* = \arg\max_\pi V_\pi. \tag{6.3}$$

Considering a deterministic optimal anomaly detector, it should maximise the performance, and, under the cases where $d^\pi(s)$ is roughly the same for all $s \in S$ and $|S|$ is the number of states in S, it has:

$$V^*_\pi = \max_\pi V_\pi = \frac{1}{|S|} \sum_{s \in S} \max_a Q(s,a), \tag{6.4}$$

In other words, the optimal anomaly detector π^* is fully determined by the accumulated reward function $Q(s,a)$, i.e., $\pi(s,a) = 1$ if $a = \arg\max_a Q(s,a)$. It is worth noting that here two assumptions are made: 1) the preferred anomaly detector is deterministic; and 2) $d^\pi(s) = \frac{1}{|S|}$ for all $s \in S$.

Definition 4: Experience E

The experience E is a set of tuples each of which is defined as $\langle s, a, r, s' \rangle$. $s, s' \in S$ indicate the states of the target system before and after the action a, respectively. r is the instant reward obtained under the state s with the action a. In an anomaly detection system, the actions are picked by the anomaly detector π. Therefore, the experience records all the behaviors of the anomaly detector.

According to the definitions, an anomaly detector is to be improved consistently by learning from the experience, which in principle is to gain a better estimation of $Q(s,a)$. This process is actually a key target of RL systems and can be achieved by existing RL solutions. Specifically, one could adopt Q-learning method to train a model for estimating $Q(s,a)$. Under this problem formulation, it is worth stressing that the anomaly detector π makes no assumption for anomaly detection, refrains from the cumbersome work of threshold selection and is capable of consistently improving its capability through the advancement of the estimation of $Q(s,a)$.

6.4 System Architecture

According to the problem formulation of anomaly detection, the utilisation of the framework of RL and the label information are the keys to construct the anomaly detector which features the advantages mentioned in the introduction. To vividly demonstrate how RL helps shape the anomaly detector, the overall architecture of the system is presented in Fig. 6.1 following the general architecture of RL. Generally, the purpose of the architecture is to train an time series anomaly detector, i.e., the agent, that could be leveraged in diverse applications and also enhanced according to specific needs. In what follows, I will dive into every detail of the architecture and discuss the validity of the designs.

Agent

As the most critical part of the architecture, the agent in the RL framework works as the time series anomaly detector that takes the target time series and previous related decisions as input, i.e., state s, and outputs the new decision made for the target time series, i.e., action a. The agent obtains the feedback for its decision, i.e., reward r, from the environment E and updates its model accordingly to enhance its accuracy in decision-making. This whole process continues until the decisions made by the agent are satisfactory. Thereafter, the agent could be applied in similar time series anomaly detection tasks.

In principle, the agent consists of three key components:

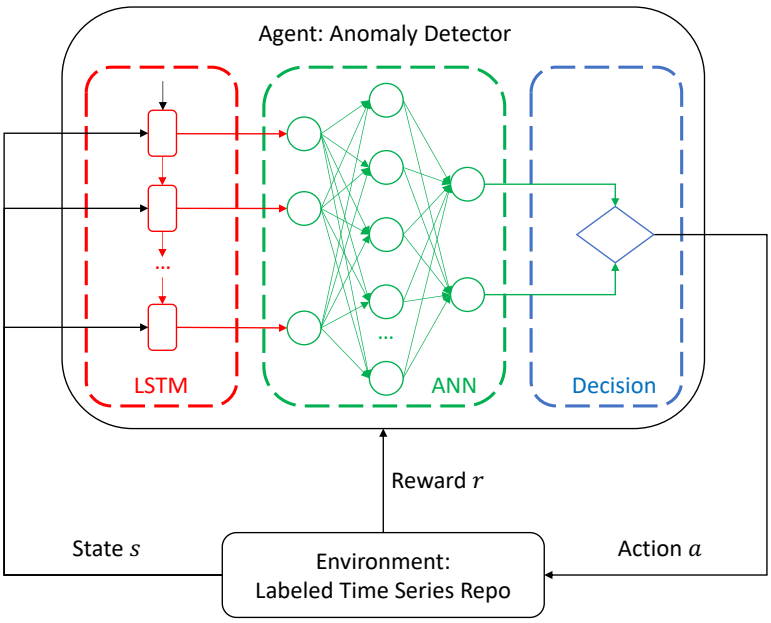


Fig. 6.1 The architecture of the proposed system

- a RNN implemented using Long-Short Term Memory (LSTM). The purpose of the RNN is to extract the sequential information within the input, i.e., state s, and output encoded information to the next model. It could be regarded as a model that learns effective representations of the inputs for better model performance;
- a fully-connected Neural Network (ANN) that takes the outputs from RNN as inputs and yields two values, i.e., $Q(s,a=1)$ and $Q(s,a=0)$, indicating the preferences for the action a. Therefore, the ANN endeavors approximating the action-value function $Q(s,a)$ which is critical in RL;
- a simple decision-maker that compares the outputs, i.e., $Q(s,a=1)$ and $Q(s,a=0)$, from the ANN and gives the final output of the agent. Its output is the final decision of the action for the current state s, i.e., $a=1$ or $a=0$.

These three components work closely together to support accurate action-value estimation. While the final output of the agent is the action a, the RNN and ANN are trained through utilising the feedback, i.e., reward r, to correct the action-value function $Q(s,a)$. Formally, the updating process follows Q-learning [74]:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot \Big(r_t + \gamma \cdot \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) \Big), \tag{6.5}$$

where t is the time step index, α is the learning rate, and γ is the discount factor.

Environment

Another key component of the architecture is the environment that governs the training of the agent. Essentially, the environment takes the action a produced by the agent as its input and generates a reward r and the next environment state s for the agent. To train the time series anomaly detector, the environment is a time series repository that maintains a large population of **time series with labeled anomalies**. Using the labeled time series, the environment is able to generate specific states for training the agent and determine the goodness of the actions taken by the agent. The detailed formulation of the state and reward are given in the following sub-sections.

State

The state s is the input of the agent. It maintains two sequences:

- the sequence of the target time series values, i.e., $s_{time} =< x_t, x_{t+1}, \cdots, x_{t+n} >$;
- the sequence of the previous actions, i.e., $s_{action} =< a_{t-1}, a_t, \cdots, a_{t+n-1} >$.

Note that n here is the size of the sliding window which is leveraged to obtain a section of the original time series for time series embedding. And it is worth stressing that the size of s_{action} is equal to that of s_{time}. The action a_{t+n} is the target output of the agent in time step $t+n$ and is to be determined.

Based on the design, it is clear that the state s is dependent on the current time series and also the previous actions. In other words, the previous actions made by the agent will change the state and, thus, affect the decisions for subsequent actions. This design turns the time series anomaly detection process into a MDP that enables the utilisation of the RL framework.

Action

The action a is the output of the agent. It is determined by the comparison between the two action-values, i.e., $Q(s, a = 0)$ and $Q(s, a = 1)$, produced by the RNN and ANN. More formally, the action space is $A = \{0, 1\}$ where 1 stands for the detection of an anomaly and 0 otherwise. Given a state s, the action of the agent is selected as:

$$a = \arg\max_{a} Q(s, a). \tag{6.6}$$

Reward

Based on the labeled time series, the reward r of the action a taken under the state s is defined according to its correctness. From the perspective of anomaly detection, if the action correctly identifies an anomaly, i.e., True Positive (TP), a positive reward will be granted. On the other hand, if the action erroneously identifies a normal state as an anomaly, i.e., False Positive (FP), or identifies an anomaly as normal state, i.e., False Negative (FN), a negative reward is given to the agent. And for actions that correctly identify normal state, i.e., True Negative (TN), a tiny reward is given. To summarise, the reward function is designed as:

$$r = \begin{cases} A, & \text{if the action is a TP} \\ B, & \text{if the action is a TN} \\ -C, & \text{if the action is a FP} \\ -D, & \text{if the action is a FN} \end{cases} \tag{6.7}$$

where A, B, C, D are all positive numbers and $A > B$. The different assignments of the parameters A, B, C, D reveal the diverse preferences of the anomaly detector. For instance, in situations where no false alarm is allowed, C is set to a large value for suppressing the false positive of the anomaly detector. Therefore, in specific applications, A, B, C, D should be adjusted for training a decent anomaly detector.

6.5 Discussion

Concerning the aforementioned architecture, there are some vital points that are worth highlighting to clarify the initial motivation and ultimate purpose of the design. The points mainly cover:

1. The relationship between time series anomaly detection and different types of MDP, i.e., one-step MDP and multi-step MDP;
2. The current difficulties of online learning;
3. The potential utilisation of active learning.

These points respectively emphasise the reason why RL is adopted instead of typical classification solutions, the difficulties of real-world online learning under the framework of RL and the potential benefits and solutions of active learning. The first point focus on explaining the validity of the current design, while the latter two points are issues that have to considered in practical applications and await future research efforts.

6.5.1 Time Series Anomaly Detection and Markov Decision Process

The process of time series anomaly detection is a decision process that typically takes the current time series as input and decides whether the time series is anomalous. In real-world applications, the decision process in a time step may also involve the consideration of the previous decisions. For instance, in some situations where alarms are rare, a single alert of the anomaly is sufficient to raise attention. Therefore, during an anomalous period, as long as the very first anomaly is reported, it is not necessary to report the same anomaly again. Nevertheless, in some other situations, users may want to report the entire period of the anomaly to gain more detailed information. As a result, after the witness of the first anomaly, a list of anomalies are reported to show the duration and intensity of the anomalies.

One-step MDP and Multi-step MDP

The key difference between the above-mentioned situations concerns how to report the anomalies which features in the delay, the duration, the frequency and etc. of the alerts. These features of reporting anomalies can be nicely captured by using multi-step MDP rather than one-step MDP. In **one-step MDP**, the anomaly detector solely focuses on the current decision of anomaly detection without taking into account its previous decisions, i.e., each decision process has only one step. While in **multi-step MDP**, the decision process has multiple steps and the previous decisions affect the later states and hence the later decisions. Compared to one-step MDP, multi-step MDP is a better way to formulate time series anomaly detection because it also captures the ways of reporting anomalies in the time series. The utilisation of multi-step MDP converts the problem of time series anomaly detection into a solid RL problem in which the agent, i.e., anomaly detector, is trained to perform nicely in the environment, i.e., correctly identifying anomalies in the labeled time series repository. Furthermore, the adoption of the framework of RL naturally empowers the anomaly detector to improve consistently.

6.5.2 Online Learning and Manual Time Series Labeling

Generally speaking, the RL framework is a natural architecture to support the online learning of the agent through interacting with the environment. The agent consistently searches over the state space and action space in the environment to find the optimal policy. The searching process is typically online, meaning that the states provided by the environment is available in a sequential order and the agent has to learn sequentially to improve itself. Additionally, in the setting of the time series anomaly detection problem, the notion of online learning refers to another critical issue, i.e., learning from the environment which evolves online.

Temporarily, the environment in the RL framework is stable, i.e., the labeled time series repository is static and prepared beforehand. To support the online learning of a universal agent, labeled time series datasets should be generated dynamically and added to the labeled time series repository. This process is crucial for the agent, i.e., anomaly detector, to adapt well in new application domains. Although this process is desired, currently there are several difficulties. The primary difficulty comes from the cumbersome work of manual anomaly labeling. This work typically requires solid experience of time series analysis that is not easy to obtain, and the work itself is time-consuming and exhausting. In addition, the process becomes even more difficult that, in some scenarios, few engineers can determine the anomalies by simply observing the real-time time series. Therefore, the process is troublesome if no strategy is employed to help engineers to improve the accuracy and efficiency. At the first glance, active learning may help ease the problem. Active learning will be discussed in the next sub-section. Note that, in the source code provided[1], the anomaly detector is currently trained with a static labeled time series repository. The online evolvement of the environment is an issue that awaits practical solution.

6.5.3 Active Learning and Automatic Labeled Time Series Generation

Due to the difficulty of the labeling task, the introduction of active learning, which tries to optimise the learning process with less training data, is of great benefit. The basic logic of active learning under the scenario of time series anomaly detection is to preferentially label the time series in which the anomaly detector produces unconfident results in order to support the training with less labeled data. This could be a decent strategy when there are bunches of unlabeled time series on hand. In situations where no further time series is available, additional time series could be generated according to the existing time series in which the anomaly detector performs poorly. In both of these situations, the unlabeled time series are sent to human experts for labeling. To further reduce the burden of human experts, the time series could be firstly labeled by the anomaly detector and the experts could focus on the correction of the labeling results.

Another way to get rid of the manual labeling task is to generate labeled time series automatically. This could be done by injecting anomalies into normal time series. To achieve this, the easiest way is to find repositories of normal time series and anomalies respectively and compose a new time series based on the known time series. A more complex way is to train models for generating normal time series and time series anomalies. Both of these are feasible solutions and will be examined in the future work.

[1] https://github.com/chengqianghuang/exp-anomaly-detector

Table 6.1 The specification of the prototype

Item	Value	Description
n_steps	25	The number of LSTM cells in a RNN layer.
n_input_dim	2	The dimension of the input for each LSTM cell.
n_hidden_dim	64	The dimension of the hidden state in each LSTM cell.
n_rnn_layers	1	The number of RNN layers.
n_ann_layers	0	The number of ANN hidden layers.
n_output_dim	2	The dimension of the output in ANN.
A	5	The immediate reward for TP.
B	1	The immediate reward for TN.
-C	-1	The immediate reward for FP.
-D	-5	The immediate reward for FN.
γ	0.9	The discount factor for Q-learning
Learning method	Adam	The optimiser used to train the networks.
Learning rate	0.01	The initial learning rate to train the networks.

6.6 Experiment Results

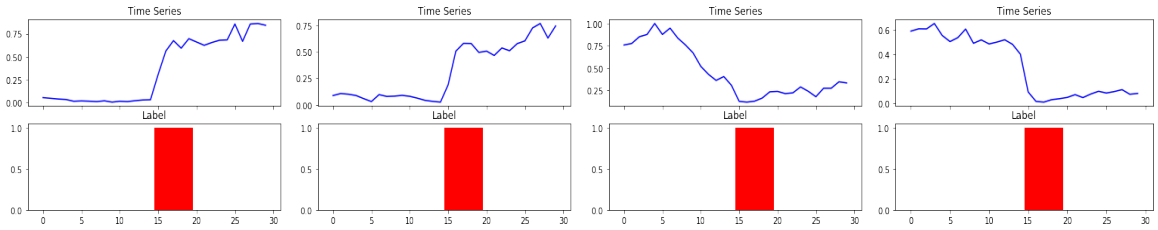
To prove the validity of the design, a prototype of the architecture is implemented. Specifically, in the experiments, the RNN in the agent is realised using one layer of LSTM cells. The last output of the LSTM cells is sent to the ANN which acts as a linear transformation function, i.e., no hidden layer and activation function are implemented in the experiments. A detailed specification is provided in Table 6.1. For more details, please check the Github[2].

Based on the specification, several sets of experiments are designed targeting at 1) proving the advantage of multi-step MDP over one-step MDP in anomaly detection tasks and 2) examining the performance of the multi-step MDP anomaly detector in terms of F1-score in two scenarios, i.e., anomaly detection with similar types of time series and that with different types of time series.

6.6.1 One-step MDP and Multi-step MDP

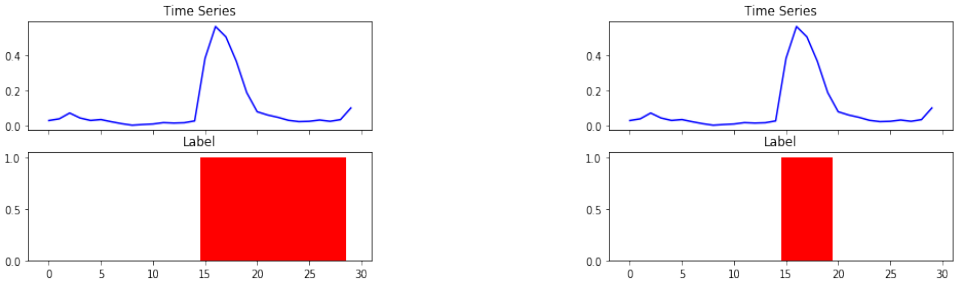
In Section 6.5.1, a concise discussion concerning the relationship between time series anomaly detection and MDP is presented. Generally speaking, the time series anomaly detection problem is more a multi-step MDP rather than a one-step MDP. In one-step MDP, the anomaly detection problem falls back to traditional classification, while multi-step MDP captures the connections among anomaly detection decisions and supports more decision-making policies.

[2]https://github.com/chengqianghuang/exp-anomaly-detector



(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4

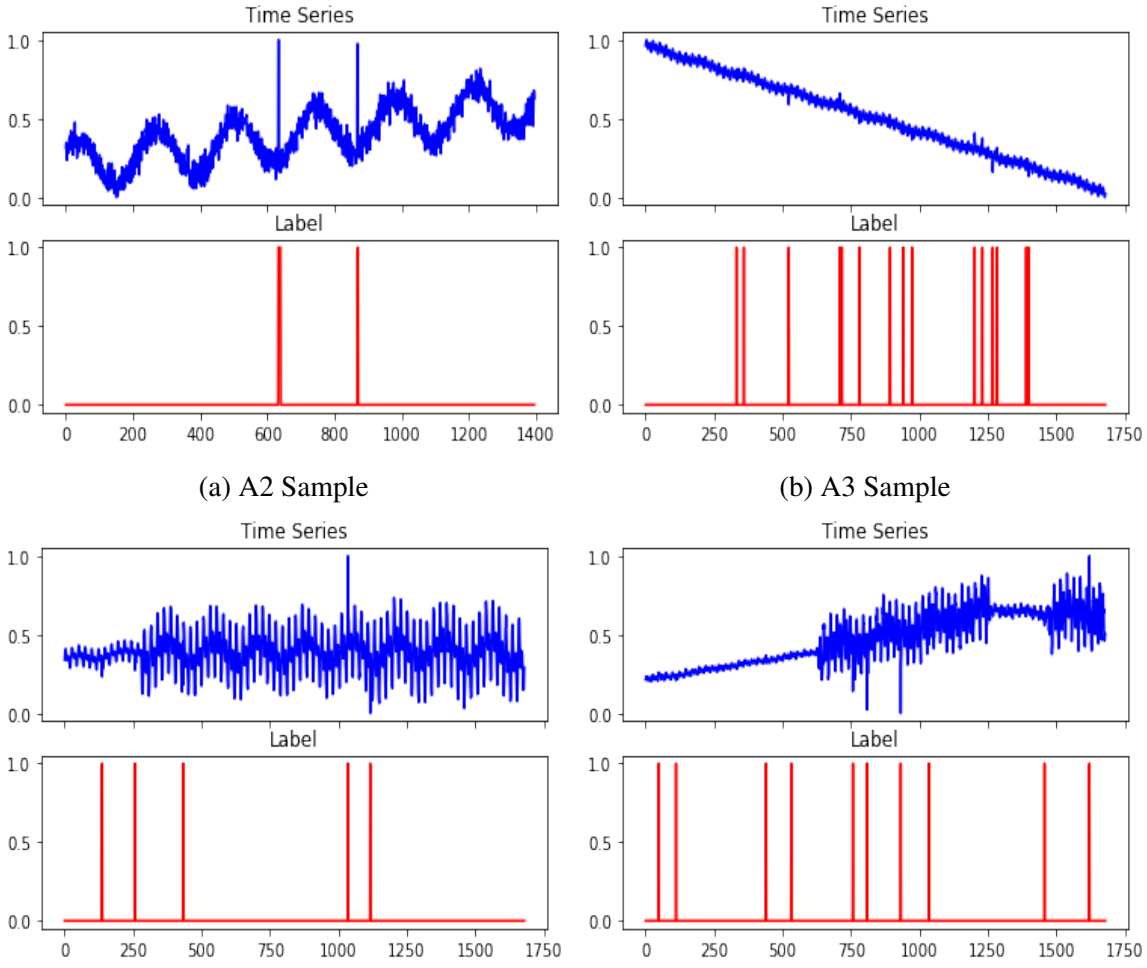
Fig. 6.2 Labeled time series datasets



(a) One-step MDP (b) Multi-step MDP

Fig. 6.3 Sample anomaly detection results by different MDPs

In what follows, an experimental example will be illustrated to demonstrate the advantage of multi-step MDP over one-step MDP. Theoretically, the one-step MDP does not consider the decisions made by previous steps and solely focus on the present time series for decision-making, while the multi-step MDP takes previous decisions as inputs and decisions are made accordingly to reflect the preference of reporting anomalies in training datasets. Consider a training dataset in which all the time series are labeled as that in Fig.6.2. Note that anomalies in the labeled datasets will not be reported consecutively for more than 5 times, which is a potential rule used in labeling the datasets. The datasets are used to train one-step MDP model and multi-MDP model respectively, which results in two styles of anomaly detection shown in Fig.6.3. Specifically, when one-step MDP observes an abrupt change in the time series, it reports anomalies and does not consider the previous actions. As a result, in Fig.6.3, one-step MDP reports 5 anomalies for the sudden increase of the time series values and another 5 anomalies for the sudden drop of the time series values. Multi-step MDP, on the other hand, takes the previous actions into consideration and decides to stop reporting anomalies when 5 consecutive anomalies have been reported. This simple experiment confirms that the multi-step MDP has its advantages over one-step MDP and is a better way to formulate time series anomaly detection especially when people have expectations of how anomalies are reported.



(a) A2 Sample

(b) A3 Sample

(c) A4 Sample 1

(d) A4 Sample 2

Fig. 6.4 Labeled time series datasets in A2Benchmark, A3Benchmark and A4Benchmark

6.6.2 Anomaly Detection with Similar Types of Time Series

To examine the capability of the anomaly detector trained through the proposed architecture, in this part the performance of the anomaly detector in analysing similar types of time series datasets is first considered. To this end, the training dataset and testing dataset are of similar types in the experiments.

Datasets

More specifically, Yahoo A2Benchmark, A3Benchmark and A4Benchmark datasets are selected as the target datasets for time series anomaly detection. Any of these datasets includes 100 time series each of which is similar to the others in the pattern but different in the anomalies. In the A2Benchmark dataset, most of the anomalies are point anomalies and

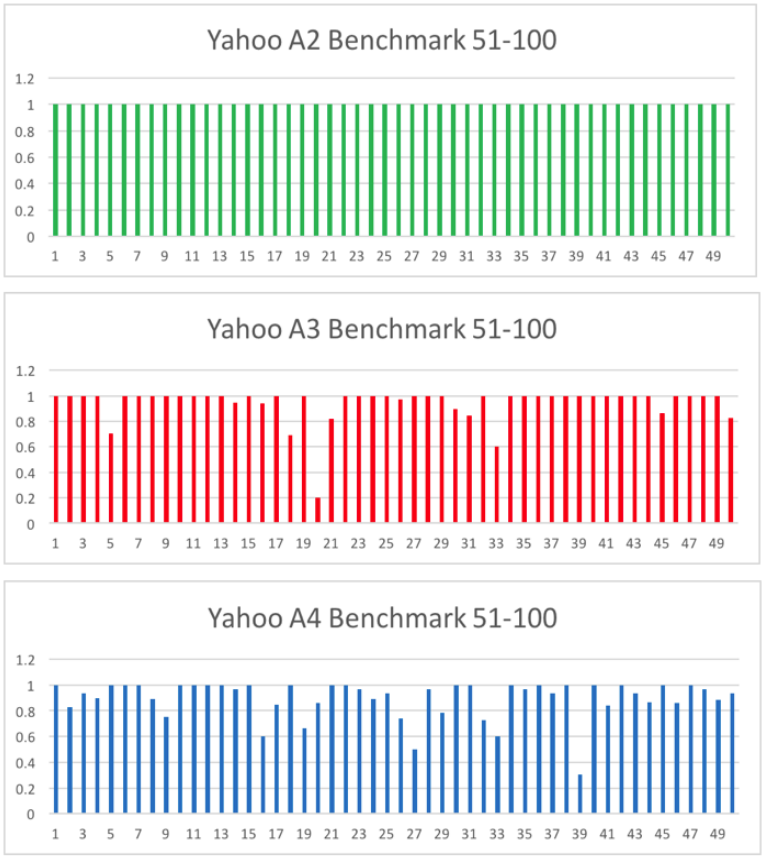
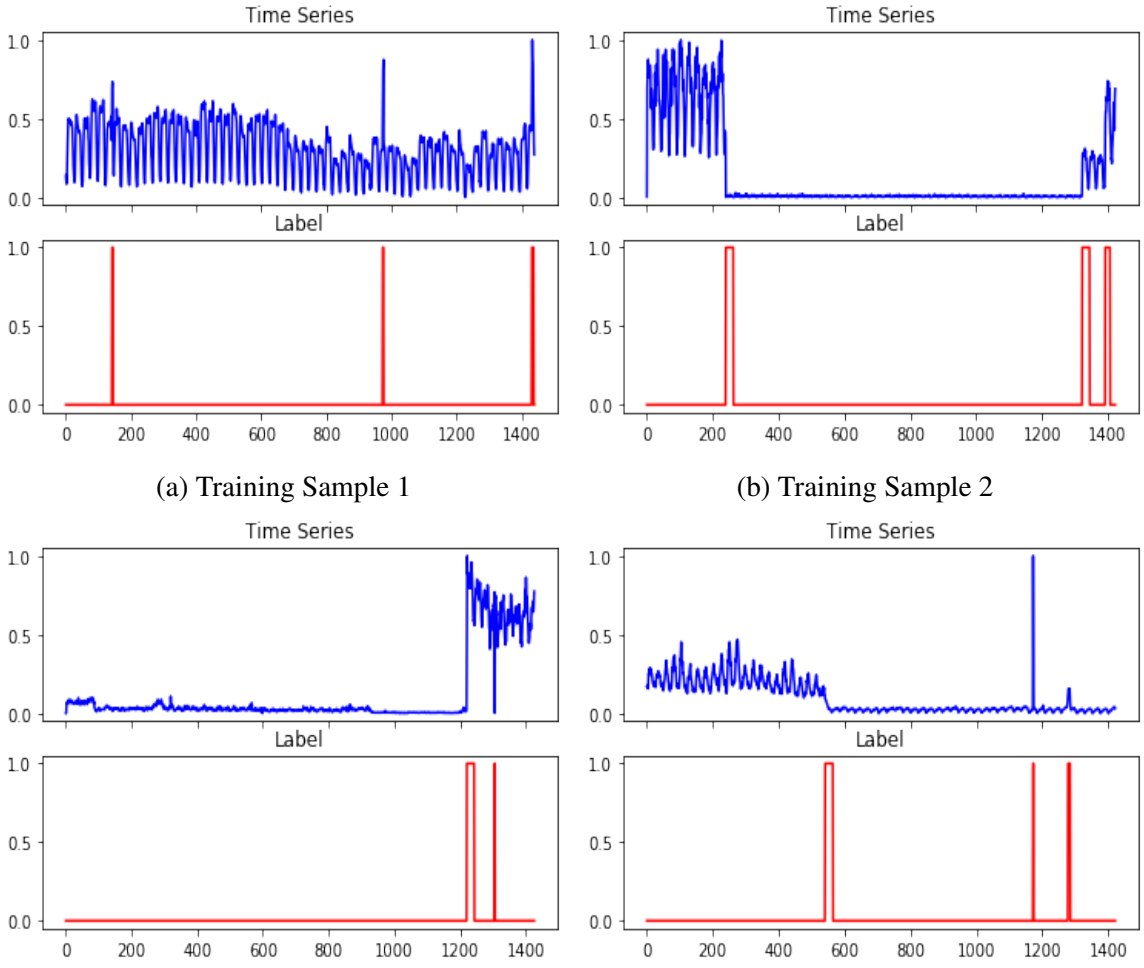


Fig. 6.5 The results (F1-score) of anomaly detection in Yahoo A2Benchmark-A4Benchmark

the data patterns are easy to analyse. While in A3Benchmark and A4Benchmark datasets, much more complex data patterns, change points and data noise are also included. Some sample time series and the corresponding labels are given in Fig.6.4. Note that for each benchmark dataset, 50 time series are utilised to train the anomaly detector and the remaining 50 time series are used for testing.

Results

Fig.6.5 shows the experiment results in Yahoo A2Benchmark, A3Benchmark and A4Benchmark datasets. It is shown that, for A2Benchmark dataset, all the anomaly detection results (F1-score) of the time series are 1, which shows that the anomaly detector has a perfect performance in this dataset. In A3Benchmark dataset, most of the F1-scores remain high and some of them are below 0.5, such as the 20th time series. Similarly, A4Benchmark dataset experiences a slight drop in the average F1-score. This is mainly due to the increasing difficulties of anomaly detection in the presence of various types of anomalies, e.g., change points, high noise and complex data patterns.



(a) Training Sample 1

(b) Training Sample 2

(c) Testing Sample 1

(d) Testing Sample 2

Fig. 6.6 Labeled time series datasets in A1Benchmark

6.6.3 Anomaly Detection with Different Types of Time Series

Generally speaking, after training, the anomaly detector shows promising performance in detecting anomalies in similar types of time series. To further analyse its potential capability, the anomaly detector is tested under the setting that the training dataset and testing dataset are very different in the types of time series patterns and anomalies. This is to examine whether the anomaly detector could generalise well to deal with unseen time series and anomalies. Firstly, the Yahoo A1Benchmark dataset is separated into a training part and a testing part to verify the capability of the anomaly detector. Secondly, all the Yahoo benchmark datasets, i.e., A1Benchmark-A4Benchmark, are utilised as the training data, while Numenta dataset is selected as the testing dataset. With the increment of training data, the anomaly detector is expected to perform much better in Numenta dataset.

Yahoo A1Benchmark First Half & Second Half

Some sample time series for training and testing are given in Fig.6.6. It is noted that time series in A1Benchmark are relatively complex in time series patterns and each time series has different patterns and anomalies that there does not exist a single anomaly detection method which could accurately identify all the anomalies in the dataset. Table 6.2 shows the F1-scores of four different methods in anomaly detection over Yahoo A1Benchmark dataset. The column of "Index" presents the indexes of the testing time series in the dataset. The columns of "Twitter", "Numenta" and "Skyline" are the anomaly detection performances of these three methods respectively. The detailed description of the methods could be found in Table 3.2. The method proposed in this chapter is shown as "Ours" in the given table. It is found that the proposed method achieves the best performances in 15 of the 33 time series anomaly detection problems. A detailed analysis shows that the proposed method transcends the other methods in situations where the anomalies are complex time series patterns and change points, i.e., time series 20, 35 and 46. However, due to the limited training resources, i.e., the 34 time series in A1Benchmark, the proposed method can not generalise well in anomaly detection tasks. That is, in situations where the time series patterns are unseen, the proposed method performs poorly. As a result, the proposed method has to leverage the RL framework to update its model for better performance. Note that, although the initial training process and the updating of the model both take a considerable amount of time, the proposed method does not require any manual parameter tuning process during the processes. Nevertheless, all the other methods compared here have to manually tune their parameters for best capable performances. The tuning process could be a complex procedure that asks for specific domain knowledge and intensive human interaction. Overall speaking, the comparison results in Table 6.2 is acceptable and it is expected that with more training resources, the proposed method could transcend other methods in anomaly detection performance without the complex tuning of parameters.

Yahoo Benchmark Datasets & Numenta Dataset

To demonstrate the performance of the proposed anomaly detector in dealing with unknown time series, the anomaly detector is trained with all the Yahoo benchmark datasets [249] and tested using Numenta [126] dataset. This is to ensure that the training and testing datasets are radically different so that the testing results reflect the essential capability of the anomaly detector, i.e., the grasp of the concept of anomalies. Specifically, the Yahoo benchmark datasets include 367 labeled time series with varying patterns and anomalies, while the Numenta dataset contains 58 labeled time series.

Table 6.2 The comparison among anomaly detection methods in Yahoo A1Benchmark

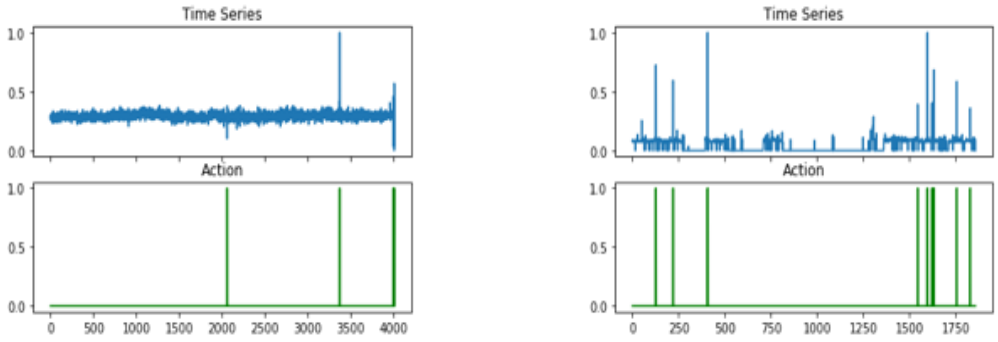
Index	**Twitter**	**Numenta**	**Skyline**	**Ours**
8	**1**	0.95	0.87	0.83
9	**1**	0.67	**1**	0.94
18	**1**	**1**	0	0.94
19	**0.99**	0.9	0.81	0.04
20	0.11	0.11	0	**0.89**
21	**1**	**1**	**1**	0.03
22	**1**	**1**	0.99	**1**
23	0.97	0.6	**1**	0.33
24	**1**	**1**	**1**	0.43
25	**1**	**1**	0.99	0.04
30	**1**	0.94	0.95	0.64
31	0.96	0.98	0.59	**1**
32	0.75	**0.76**	**0.76**	0.5
34	0.73	0.83	0.73	**1**
35	0	0	0	**0.53**
36	**1**	**1**	**1**	0.03
37	**0.82**	0.38	0.16	0.23
40	0	0	0	**0.5**
41	**1**	0	0.86	0.85
42	**1**	**1**	0.31	0.14
43	0.4	0.39	0.39	**1**
44	0.875	0.875	0.875	**1**
45	**1**	**1**	**1**	0.01
46	0	0	0	**0.62**
47	0	0.56	0	**0.69**
50	**1**	0.93	0.93	0.02
51	0	0	0	**1**
52	0.96	0.84	**1**	**1**
53	**1**	0.97	0.97	0.13
54	0.67	0.67	0.67	**1**
55	0	0	0.33	**0.83**
56	0.33	**0.5**	0	0.29
57	0	0	0.4	**0.95**

Due to the distinct labeling strategy in Numenta dataset, which is inconsistent with that in Yahoo benchmark datasets, one can not adopt the quantitative analysis of the anomaly detection performance over the testing results. Instead, qualitative results are presented. Fig. 6.7 shows the performances of the anomaly detector in some Numenta time series after training through the memory replay of the anomaly detection experience in Yahoo benchmark datasets. The original time series data are marked as blue lines and the green lines indicate the actions performed by the anomaly detector. It is worth noticing that the anomaly detector is capable of identifying shift of means, point anomalies and anomalous patterns of the target time series, and achieves very high-quality results in the testing time series.

Although the anomaly detector performs nicely in most of the testing time series, there are some of the time series that have unsatisfactory results, i.e., Fig.6.8a, 6.8b and 6.8c. One of the reasons is that the concepts of the anomalies are shaped by the training dataset, i.e., anomalies are reported whenever a known "anomalous" time series pattern occurs. Therefore, the change points in Fig.6.8a and 6.8c are all detected as anomalous regardless of the global/contextual information in the time series. In Fig.6.8b, the reason for the unexpected results is because the time series pattern is unseen and closer to anomalous patterns. To mitigate these issues, the involvement of the contextual information, e.g., periodicity, and the overall time series pattern in the anomaly detection process is required. This is left as a primary task in the future work.

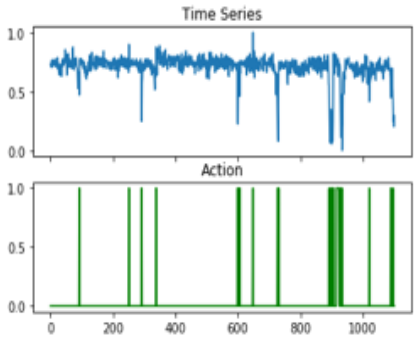
6.7 Conclusion

To summarise, this chapter proposes a design of time series anomaly detector which is fully determined by the experience of anomaly detection without explicit definitions or assumptions of anomalies. No threshold is required for the anomaly detector. And, with growing experience, it is expected that the anomaly detector can keep evolving and is able to perform nicely in general and new anomaly detection problems. The experiment results show that, in general, the anomaly detector is promising in achieving time series anomaly detection with high-quality results and desirable benefits. To extend the applicability of the method, the problem of generating accurately labeled time-series datasets of various types for anomaly detection training is considered as the next step. And, at the same time, how to integrate more information, e.g., contextual information, in the anomaly detection model is another critical task that awaits investigation. Besides, it is argued in this chapter that deep neural networks and active learning methods are also possible directions to improve the results and make the method more practical.

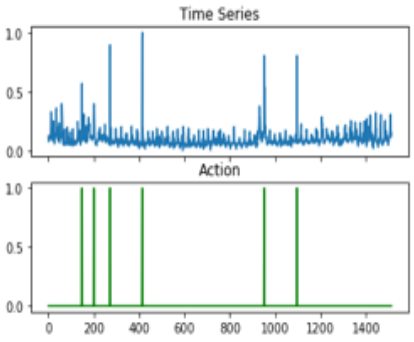


(a) Sample 1

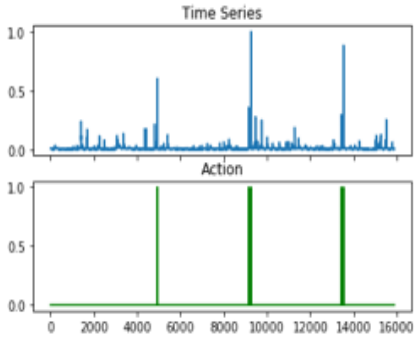
(b) Sample 2



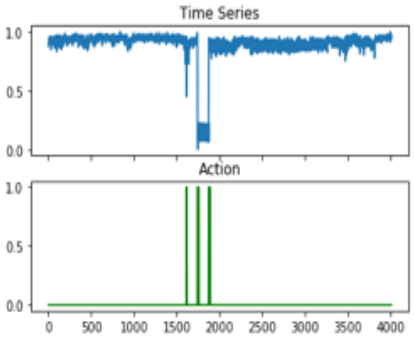
(c) Sample 3



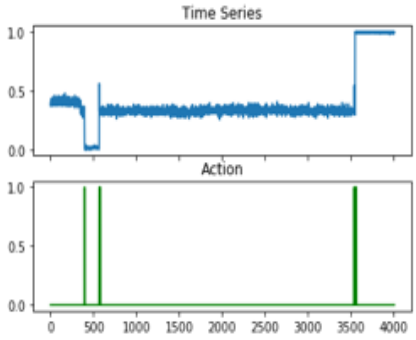
(d) Sample 4



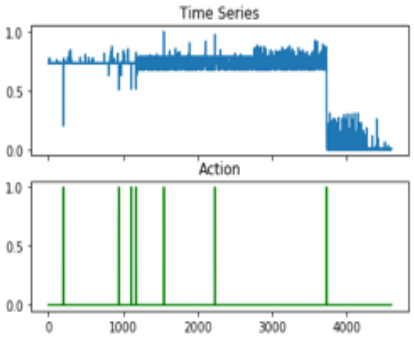
(e) Sample 5



(f) Sample 6

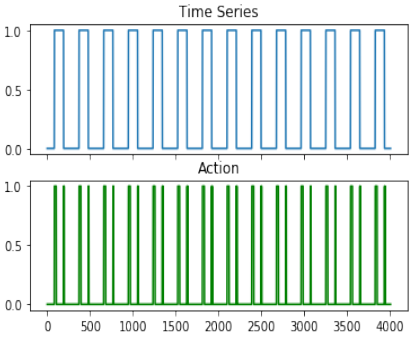


(g) Sample 7

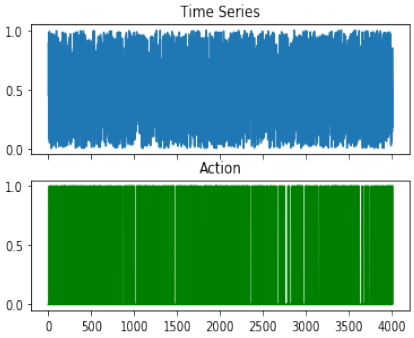


(h) Sample 8

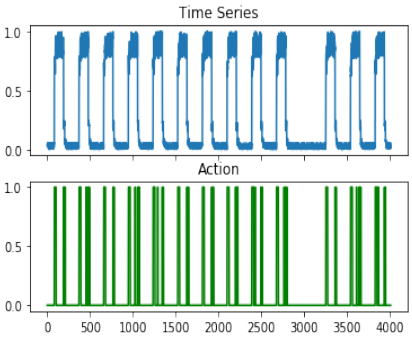
Fig. 6.7 The performance of the proposed method in Numenta dataset (Satisfactory)



(a) Sample 1



(b) Sample 2



(c) Sample 3

Fig. 6.8 The performance of the proposed method in Numenta dataset (Unsatisfactory)

Chapter 7

Conclusion and Future Work

Recent years have witnessed intense research efforts in developing novel anomaly detection strategies and methods that strive to meet various requirements of different applications. With the ever-growing demands of existing applications, e.g., high-speed network traffic anomaly detection, and the emerging research domains, e.g., anomalous event/behavior detection, the research on anomaly detection is expected to experience much speedy advancement in the upcoming years. In this thesis, several aspects of anomaly detection problems have been examined, especially time series anomaly detection, and four primary contributions have been made with the faith that they will give rise to more research ideas and benefit practical anomaly detection applications:

- Related concepts concerning anomaly detection are clarified and an extensive survey of existing anomaly detection strategies and methods is presented. More specifically, the fine-grained explanations of outlier and novelty, the detailed taxonomies of the anomaly types, inputs and solutions, and the classification of the anomaly detection applications are all presented. Based on the solid understanding of the fundamental concepts, the survey covers high-level anomaly detection strategies/tactics as well as specific anomaly detection methods. Generally speaking, there exist four basic strategies, i.e., rule-based, case-based, expectation-based and property-based strategies, while anomaly detection methods are classified into five categories, i.e., distance-based, density-based, boundary-based, partition-based and property-based methods. The concepts and taxonomies present a rounded research background for general anomaly detection. In addition, customary strategies and specific techniques for time series analysis are also reviewed in order to provide a complete research background for time series anomaly detection.

- To response to the high false alarm rate of anomaly detection over network time series and take contextual information into consideration, special attention has been focused on the utilisation and modification of Support Vector Data Description (SVDD), a famous tool for one-class classification, in order to support more accurate or interpretable anomaly detection in related applications. To be more specific, the high false positive rate of SVDD in anomaly detection over noisy datasets is carefully investigated and a countermeasure is taken to relax the decision boundary using additional density information. With the theoretical analysis of the range of the parameter, which constraints the relaxation, SVDD demonstrates high accuracy in anomaly detection. Besides, coordinating contextual information in SVDD is also studied aiming at providing a more clear explanation of the detected anomalies, i.e., anomaly interpretation. The contextual information supplements additional ingredients and helps to distinguish the root cause of the anomalies, i.e., original data or its contexts. Overall, featured anomaly detection methods based on SVDD exhibits enhanced capabilities in accuracy and interpretability.

- A novel featured anomaly detection method, i.e., Convex Hull Data Description (CHDD), is developed in an attempt to achieve data description with profitable features, the most important of which is the capability to realise one-class classification and clustering tasks at the same time. Based on convex hull analysis, the principle of CHDD is to describe a data with a linear combination of all the extreme points in the convex hull of the given dataset. The principle has revealed several advantageous attributes: 1) the identification of the convex hull helps with the description of the overall structure of the given dataset, which benefits applications that favors data representatives; 2) a data could be interpreted through the extreme points or, if it is an anomaly, the extreme point will also provide valuable information concerning the reason why the data is considered anomalous; 3) the handling of the convex hull emphasises the extreme points while the other points could be neglected, which mimics the benefit brought by Support Vectors (SV) and opens the possibility of incremental anomaly detection through incremental analysis of extreme points. Therefore, it is argued that CHDD is a featured method that has distinguished potentials in anomaly detection.

- For the purpose of processing the dynamic patterns in time series, based on the framework of Reinforcement Learning (RL), an anomaly detector is designed that can be consistently trained under the supervised setting so as to refrain from the cumbersome work of threshold setting and the uncertain definitions of anomalies in time series anomaly detection tasks. Specifically, the time series anomaly detection problem is

reformulated as a Markov Decision Process (MDP) and, therefore, the concepts in time series anomaly detection can be formulated under the general framework of RL. With a repository of labeled time series datasets, the RL is to make the time series anomaly detector learn the best possible strategy in reporting anomalies. In principle, the time series anomaly detector learns not only the concept of a time series anomaly but also the behavior of how to report anomalies. Through an initial training process, the time series anomaly detector is expected to grasp the general strategy of anomaly detection, while a specific online training could also be undertaken to fine-tune the behavior of the detector in particular applications. It is argued that the online training process could involve additional processes, e.g., active learning, to enhance the overall performance of the framework.

Generally speaking, the techniques developed in this thesis aim to supply featured anomaly detection methods in an attempt to meet distinct requirements and needs of various applications. They are initially driven from the engineering point of view and expected to reduce the labor work of human experts in anomaly detection processes. To make further progress towards this direction, the future work will primarily consider the two following aspects that may bring immediate consequences:

- Intrinsically, the fundamentals of CHDD point out that there should exist a dictionary which could express the given dataset well so that anomalous data points could be easily identified. However, recognising the best dictionary is a tough work that does not directly apply to anomaly detection. Consequently, one possible direction to circumvent this dictionary learning process could be utilising multiple sets of random samples from the given dataset as the set of dictionaries. Thus, the process of anomaly detection can focus on how well a data instance is expressed by the set of dictionaries. With this approach, the significance of a learned dictionary is reduced and more emphasis is placed on whether a data instance could be easily expressed in order to support anomaly detection. A possible way to practice this idea is to implement a Replicator Neural Network (RNN) with certain constraints on the outputs of the hidden layer. In other words, the compressed version of a data instance in RNN should be able to reconstruct the original data not only through the neural network but also with the help of a certain dictionary.

- When using RL to back anomaly detection learning, a critical difficulty is witnessed that the labeled time series repository is greatly limited according to certain applications and domains. Therefore, a way to automatically generate the labeled time series repository is in urgent demand. From an engineer's point of view, a labeled time series

could be composed by integrating a known normal time series with several known time series anomalies. As a result, the generation of a labeled time series is converted into the problems of producing normal time series and time series anomalies. Luckily, recent advances in Generative Adversarial Network (GAN) have paved the path to accurate data generation and shown great success in various domains. Therefore, one natural way of constructing a labeled time series repository is to utilised GAN and fuse generated normal and abnormal time series to fill the repository.

Although it still requires further efforts in improving the related methods and implementing them in practical products, it is believed that the developments of these methods will contribute positively to practical applications and drive the emergences of new ideas for more competent anomaly detection methods with favorable features.

References

[1] J. Aldrich, R. A. Fisher and the making of maximum likelihood 1912–1922, *Statistical Science*, vol. 12, no. 3, pp. 162-176, 1997.

[2] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician*, vol. 46, no. 3, pp. 175-185, 1992.

[3] S. Agrawal, J. Agrawal, Survey on anomaly detection using data mining techniques, *Procedia Computer Science*, vol. 60, pp. 708-713, 2015.

[4] N. K. Ahmed, A. F. Atiya, N. E. Gayar, H. El-Shishiny, An empirical comparison of machine learning models for time series forecasting, *Econometric Reviews*, vol. 29, no. 5-6, pp. 594-621, 2010.

[5] A. Arning, R. Agrawal, P. Raghavan, A linear method for deviation detection in large databases, *International Conference on Knowledge Discovery and Data Mining*, vol. 1141, no. 50, pp. 972-981, 1996.

[6] M. Agyemang, K. Barker, R. Alhajj, A comprehensive survey of numeric and symbolic outlier mining techniques, *Intelligent Data Analysis*, vol. 10, no. 6, pp. 521-538, 2006.

[7] E. G. Allan, M. R. Horvath, C. V. Kopek, B. T. Lamb, T. S. Whaples, M. W. Berry, Anomaly detection using nonnegative matrix factorization, *Survey of Text Mining II*, no. 11, pp. 203-217, 2008.

[8] A. Arnold, Y. Liu, N. Abe, Temporal causal modeling with graphical granger methods, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 66-75, 2007.

[9] M. Ahmed, A. N. Mahmood, J. Hu, A survey of network anomaly detection techniques, *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[10] M. Ahmed, A. N. Mahmood, M. R. Islam, A survey of anomaly detection techniques in financial domain, *Future Generation Computer System*, vol. 55, pp. 278–288, 2016.

[11] H. N. Akouemo, R. J. Povinelli, Probabilistic anomaly detection in natural gas time series data, *International Journal of Forecasting*, vol. 32, no. 3, pp. 948-956, 2016.

[12] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[13] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, *ACM-SIAM symposium on Discrete algorithms*, pp. 1027-1035, 2007.

[14] C. Bishop, Pattern recognition and machine learning, *Machine Learning*, 2006.

[15] J. P. Burg, A new analysis technique for time series data, *NATO Advanced Study Institute on Signal Processing*, 1968.

[16] K. P. Bennett, E. J. Bredensteiner, Duality and geometry in SVM classifiers, *International Conference on Machine Learning*, pp. 57-64, 2000.

[17] K. P. Bennett, C. Campbell, Support vector machines: hype or hallelujah?, *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 2, pp. 1–13, 2000.

[18] A. Banerjee, P. Burlina, C. Diehl, A support vector method for anomaly detection in hyperspectral imagery, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2282-2291, 2006.

[19] M. H. Bhuyan, D. K. Bhattacharyya, J. K. Kalita, Network anomaly detection: methods, systems and tools, *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303-336, 2014.

[20] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. Vapnik, Support vector clustering, *The Journal of Machine Learning Research*, vol. 2, pp. 125–137, 2002.

[21] M. M. Breunig, H. P. Kriegel, R. T Ng, J. Sander, LOF: identifying density-based local outliers, *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93-104, 2000.

[22] K. Bhaduri, B. L. Matthews, C. Giannella, Algorithms for speeding up distance-based outlier detection, *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 859-867, 2011.

[23] P. S. Bradley, O. L. Mangasarian, W. N. Street, Clustering via concave minimization, *Advances in Neural Information Processing Systems*, pp. 368-374, 1996.

[24] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, *VLDB Endowment*, vol. 5, no. 7, pp. 622-633, 2012.

[25] G. E. Box, D. A. Pierce, Distribution of residual autocorrelations in autoregressive-integrated moving average time series models, *Journal of the American Statistical Association*, vol. 65, no. 332, pp. 1509-1526, 1970.

[26] J. L. Bentley, F. P. Preparata, M. G. Faust, Approximation algorithms for convex hulls, *Communications of the ACM*, vol. 25, no. 1, pp. 64-68, 1982.

[27] Comprehensive R Archive Network (CRAN), https://cran.r-project.org/.

[28] J. Cannady, Next generation intrusion detection: Autonomous reinforcement learning of network attacks, *National Information Systems Security Conference*, pp. 1-12, 2000.

[29] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, *ACM Computing Surveys*, vol. 41, no. 3, pp. 15-58, 2009.

[30] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection for discrete sequences - a survey, *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, 2012.

[31] C. Campbell, K. P. Bennett, A linear programming approach to novelty detection, *Advances in Neural Information Processing Systems*, pp. 395-401, 2000.

[32] C. Campbell, Y. Ying, Learning with support vector machines, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 1, pp. 1-95, 2011.

[33] R. B. Cleveland, W. S. Cleveland, J. E. McRae, *STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics*, vol. 6, no. 1, pp. 3-73, 1990.

[34] G. A. Carpenter, S. Grossberg, Adaptive resonance theory, *Springer*, 2017.

[35] W. C. Chang, C. P. Lee, C. J. Lin, A revisit to support vector data description, *Technical Report, Department of Computer Science, National Taiwan University*, 2013.

[36] E. J. Candes, X. Li, Y. Ma, J. Wright, Robust principal component analysis? *Journal of the ACM*, vol. 58, no. 3, pp. 11-37, 2011.

[37] E. J. Candes, T. Tao, The power of convex relaxation: near-optimal matrix completion, *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053-2080, 2009.

[38] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[39] C. S. Chu, I. W. Tsang, J. T. Kwok, Scaling up support vector data description by using core-sets, *International Joint Conference on Neural Networks*, vol. 1, pp. 425-430, 2004.

[40] R. N. Calheiros, K. Ramamohanarao, R. Buyya, C. Leckie, S. Versteeg, On the effectiveness of isolation-based anomaly detection in cloud data centers, *Concurrency and Computation-Practice & Experience*, vol. 29, no. 18, 2017.

[41] L. Cao, D. Yang, Q. Wang, Y. Yu, J. Wang, E. A. Rundensteiner, Scalable distance-based outlier detection over high-volume data streams, *IEEE International Conference on Data Engineering (ICDE)*, pp. 76-87, 2014.

[42] Q. Cao, Y. Ying, P. Li, Similarity metric learning for face recognition, IEEE International Conference on Computer Vision, pp. 2408-2415, 2013.

[43] W. Cheng, K. Zhang, H. Chen, G. Jiang, Z. Chen, W. Wang, Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations, *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 805-814, 2016.

[44] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, Z. Wang, iBOAT - isolation-based online anomalous trajectory detection, *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806-818, 2013.

[45] G. Chen, X. Zhang, Z. J. Wang, F. Li, Robust support vector data description for outlier detection with noise or uncertain data, *Knowledge-Based System*, vol. 90, pp. 129-137, 2015.

[46] D. Defays, An efficient algorithm for a complete link method, *The Computer Journal*, vol. 20, no. 4, pp. 364-366, 1977.

[47] T. G. Dietterich, Machine learning for sequential data: A review, *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, pp. 15-30, 2002.

[48] J. K. Dutta, B. Banerjee, C. K. Reddy, RODS: Rarity based outlier detection in a sparse coding framework, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 483-495, 2016.

[49] M. M. Deza, E. Deza, Encyclopedia of distances, *Springer*, 2009.

[50] Z. G. Ding, D. J. Du, M. R. Fei, An isolation principle based distributed anomaly detection method in wireless sensor networks, *International Journal of Automation and Computing*, vol. 12, no. 4, pp. 402-412, 2014.

[51] E. De La Hoz, E. De La Hoz, A. Ortiz, J. Ortega, A. Martínez-Álvarez, Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps, *Knowledge-Based Systems*, vol. 71, pp.322-338, 2014.

[52] E. De La Hoz, E. De La Hoz, A. Ortiz, J. Ortega, B. Prieto, PCA filtering and probabilistic SOM for network intrusion detection, *Neurocomputing*, vol. 164, pp.71-81, 2015.

[53] Q. Ding, E. D. Kolaczyk, A compressed PCA subspace method for anomaly detection in high-dimensional data, *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7419-7433, 2013.

[54] C. Ding, T. Li, M. I. Jordan, Convex and semi-nonnegative matrix factorizations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45-55, 2010.

[55] A. P. Dempster, N. M Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the royal statistical society, Series B (methodological)*, pp. 1-38, 1977.

[56] D. J. Dean, H. Nguyen, X. Gu, UBL: unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems, *International Conference on Autonomic Computing*, pp. 191-200, 2012.

[57] R. Dunia, S. J. Qin, Multi-dimensional fault diagnosis using a subspace approach, *American Control Conference*, 1997.

[58] L. Duan, M. Xie, T. Bai, J. Wang, A new support vector data description method for machinery fault diagnosis with unbalanced datasets, *Expert Systems with Applications*, vol. 64, pp. 239-246, 2016.

[59] X. Ding, Y. Li, A. Belatreche, L. P. Maguire, Novelty detection using level set methods, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 576-588, 2015.

[60] B. Du, L. Zhang, A discriminative metric learning based anomaly detection method, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6844-6857, 2014.

[61] W. Enders, Stationary time-series models, *Applied Econometric Time Series, Wiley*, pp. 48-107, 2004.

[62] Etsy, Skyline, *https://github.com/etsy/skyline*, 2013.

[63] P. Esling, C. Agon, Time-series data mining, *ACM Computing Surveys (CSUR)*, vol. 45, no. 1, pp. 12-34, 2012.

[64] A. Erdinç, S. Aksoy, Anomaly detection with sparse unmixing and gaussian mixture modeling of hyperspectral images, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 5035-5038, 2015.

[65] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *SIGKDD Conference on Knowledge Discovery and Data Mining*, vol. 96, no. 34, pp. 226-231, 1996.

[66] E. Elhamifar, Sparse modeling for high-dimensional multi-manifold data analysis, *The Johns Hopkins University*, 2012.

[67] E. Elhamifar, G. Sapiro, R. Vidal, See all by looking at a few: Sparse modeling for finding representative objects, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600-1607, 2012.

[68] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765-2781, 2013.

[69] H. Fanaee-T, J. Gama, Tensor-based anomaly detection - an interdisciplinary survey, *Knowledge-Based System*, vol. 98, pp. 130–147, 2016.

[70] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, *Advances in Neural Information Processing Systems*, pp. 2962-2970, 2015.

[71] Q. Fan, Z. Wang, D. Li, D. Gao, H. Zha, Entropy-based fuzzy support vector machine for imbalanced datasets, *Knowledge-Based Systems*, vol 115, pp. 87-99, 2016.

[72] E. S. Gardner, Exponential smoothing: the state of the art, *International Journal of Forecasting*, vol. 4, no. 1, pp. 1-28, 1985.

[73] E. S. Gardner, Exponential smoothing: the state of the art - Part II, *International Journal of Forecasting*, vol. 22, pp. 637-666, 2006.

[74] A. Gosavi, Reinforcement learning: A tutorial survey and recent advances, *INFORMS Journal on Computing*, vol. 21, no. 2, pp. 178-192, 2009.

[75] F. E. Grubbs, Sample criteria for testing outlying observations, *The Annals of Mathematical Statistics*, pp. 27-58, 1950.

[76] P. Gogoi, D. K. Bhattacharyya, B. Borah, J. K. Kalita, A survey of outlier detection methods in network anomaly identification, *The Computer Journal*, vol. 54, no. 4, pp. 570–588, 2011.

[77] N. Görnitz, M. Braun, M. Kloft, Hidden markov anomaly detection, *International Conference on Machine Learning*, pp. 1833-1842, 2015.

[78] S. M. Guo, L. C. Chen, J. S. H. Tsai, A boundary method for outlier detection based on support vector domain description, *Pattern Recognition*, vol. 42, no. 1, pp. 77-83, 2009.

[79] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, Anomaly-based network intrusion detection - techniques, systems and challenges. *Computers & Security*, vol. 28, no. 1, pp. 18–28, 2009.

[80] M. Gupta, J. Gao, C. Aggarwal, J. Han, Outlier detection for temporal data: a survey, *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2250–2267, 2014.

[81] Y. Ge, G. Jiang, M. Ding, H. Xiong, Ranking metric anomaly in invariant networks, *ACM Transactions on Knowledge Discovery From Data*, vol. 8, no. 2, pp. 8-30, 2014.

[82] M. Gonen, A. A. Margolin, Localized data fusion for kernel k-means Clustering with application to cancer biology, *Advances in Neural Information Processing Systems*, pp. 1305-1313, 2014.

[83] S. Guha, N. Mishra, G. Roy, O. Schrijvers, Robust random cut forest based anomaly detection on streams, *International Conference on Machine Learning*, pp. 2712-2721, 2016.

[84] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, *Advances in Neural Information Processing Systems*, pp. 2672-2680, 2014.

[85] J. Gan, Y. Tao, DBSCAN revisited: mis-claim, un-fixability, and approximation, *ACM SIGMOD International Conference on Management of Data*, pp. 519-530, 2015.

[86] X. Guan, W. Wang, X. Zhang, Fast intrusion detection based on a non-negative matrix factorization model, *Journal of Network and Computer Applications*, vol. 32, no. 1, pp. 31-44, 2009.

[87] F. Hayes-Roth, Rule-based systems, *Communications of the ACM*, vol. 28, no. 9, pp. 921-932, 1985.

[88] N. D. Ho, Nonnegative matrix factorization algorithms and applications, *Ecole Polytechnique*, 2008.

[89] H. Hoffmann, Kernel PCA for novelty detection, *Pattern Recognition*, vol. 40, no. 3, pp. 863-874, 2007.

[90] V. Hodge, J. Austin, A survey of outlier detection methodologies, *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85-126, 2004.

[91] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, *OTexts*, 2014.

[92] Z. Han, H. Chen, T. Yan, G. Jiang, Time series segmentation to discover behavior switching in complex physical systems, *IEEE International Conference on Data Mining*, pp. 161-170, 2015.

[93] R. J. Hyndman, G. Athanasopoulos, 8.9 Seasonal ARIMA models, *Forecasting: principles and practice, oTexts*, 2012.

[94] S. Hawkins, H. He, G. Williams, R. Baxter, Outlier detection using replicator neural networks, *International Conference on Data Warehousing and Knowledge Discovery*, pp. 170-180, 2002.

[95] H. Hachiya, M. Matsugu, NSH: normality sensitive hashing for anomaly detection, *IEEE International Conference on Computer Vision Workshops*, pp. 795–802, 2013.

[96] J. L. Hintze, R. D. Nelson, Violin plots: a box plot-density trace synergism, *The American Statistician*, vol. 52, no. 2, pp. 181-184, 1998.

[97] Y. He, H. Tan, W. Luo, S. Feng, J. Fan, MR-DBSCAN - a scalable MapReduce-based DBSCAN algorithm for heavily skewed data, *Frontiers of Computer Science*, vol. 8, no. 1, pp. 83-99, 2014.

[98] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100-108, 1979.

[99] C. Huang, Y. Wu, Z. Yuan, G. Min, Towards practical anomaly detection in network big data, *Big Data and Computational Intelligence in Networking*, CRC Press, 2017.

[100] R. Hu, S. Wen, Z. Zeng, T. Huang, A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm, *Neurocomputing*, vol. 221, pp. 24-31, 2017.

[101] O. Ibidunmoye, F. Hernández-Rodriguez, E. Elmroth, Performance anomaly detection and bottleneck identification, *ACM Computing Surveys*, vol. 48, no. 1, p. 4, 2015.

[102] I. T. Jolliffe, Principal component analysis, *Springer New York*, 2002.

[103] G. Jiang, H. Chen, K. Yoshihira, Discovering likely invariants of distributed transaction systems for autonomic system management, *IEEE International Conference on Autonomic Computing*, pp. 199-208, 2006.

[104] N. A. James, A. Kejariwal, D. S. Matteson, Leveraging cloud data to mitigate user experience from "Breaking Bad", *IEEE International Conference on Big Data*, pp. 3499-3508, 2014.

[105] C. Ji, Y. Li, W. Qiu, U. Awada, K. Li, Big data processing in cloud computing environments, *International Symposium on Pervasive Systems, Algorithms and Networks*, pp. 17-23, 2013.

[106] Y. Ji, S. Sun, Y. Lu, Multitask multiclass privileged information support vector machines, *International Conference on Pattern Recognition*, pp. 2323-2326, 2012.

[107] T. Kohonen, The self-organizing map, *Neurocomputing*, vol. 21, no. 1-3, pp. 1-6, 1998.

[108] T. Kutsuna, A binary decision diagram-based one-class classifier, *IEEE International Conference on Data Mining*, pp. 284–293, 2010.

[109] T. Kutsuna, A. Yamamoto, A parameter-free approach for one-class classification using binary decision diagrams, *Intelligent Data Analysis*, vol. 18, no. 5, pp. 889-910, 2014.

[110] T. Kutsuna, A. Yamamoto, Outlier detection based on leave-one-out density using binary decision diagrams, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 486-497, 2014.

[111] T. Kutsuna, A. Yamamoto, Outlier detection using binary decision diagrams, *Data Mining and Knowledge Discovery*, vol. 31, no. 2, pp. 548-572, 2017.

[112] A. Karami, R. Johansson, Choosing DBSCAN parameters automatically using differential evolution, *International Journal of Computer Applications*, vol. 91, no. 7, pp. 1-11, 2014.

[113] Q. Ke, T. Kanade, Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming, *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 739-746, 2005.

[114] E. Keogh, S. Lonardi, C. A. Ratanamahatana, L. Wei, S. H. Lee, J. Handley, Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, vol. 14, no. 1, 99–129, 2007.

[115] S. S. Khan, M. G. Madden, One-class classification: taxonomy of study and review of techniques, *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[116] J. Kim, H. Naganathan, S. Y. Moon, W. K. Chong, S. T. Ariaratnam, Applications of clustering and isolation forest techniques in real-time building energy-consumption data: application to LEED certified buildings, *Journal of Energy Engineering*, vol. 143, no. 5, 2017.

[117] M. H. Kutner, C. Nachtsheim, J. Neter, Applied linear regression models, *McGraw-Hill/Irwin*, 2004.

[118] K. M. Kumar, A. R. M. Reddy, A fast DBSCAN clustering algorithm by accelerating neighbor searching using Groups method, *Pattern Recognition*, vol. 58, pp. 39-48, 2016.

[119] M. Kemmler, E. Rodner, E. S. Wacker, J. Denzler, One-class classification with Gaussian processes, *Pattern Recognition*, vol. 46, no. 12, pp. 3507-3518, 2013.

[120] H. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 444–452, 2008.

[121] B. Krawczyk, M. Wozniak, F. Herrera, On the usefulness of one-class classifier ensembles for decomposition of multi-class problems, *Pattern Recognition*, vol. 48, no. 12, pp. 3969-3982, 2015.

[122] S. B. Kotsiantis, I. D. Zaharakis, P. E. Pintelas, Machine learning: a review of classification and combining techniques, *Artificial Intelligence Review*, vol. 26, no. 3, pp. 159-190, 2006.

[123] M. Lichman, UCI machine learning repository, *http://archive.ics.uci.edu/ml*, University of California, Irvine, 2013.

[124] L. Lewis, A case-based reasoning approach to the management of faults in communication networks, *IEEE Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future*, pp. 1422-1429, 1993.

[125] N. Laptev, S. Amizadeh, I. Flint, Generic and scalable framework for automated time-series anomaly detection, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1939-1947, 2015.

[126] A. Lavin, S. Ahmad, Evaluating real-time anomaly detection algorithms – the Numenta anomaly benchmark, *International Conference on Machine Learning and Applications*, pp. 38-44, 2015.

[127] A. Lakhina, L. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 219-230, 2004.

[128] P. Laskov, C. Gehl, S. Krüger, K. R. Müller, Incremental support vector learning: analysis, implementation and applications, *Journal of Machine Learning Research*, vol. 7, pp. 1909-1936, 2006.

[129] L. Li, R. J. Hansman, R. Palacios, R. Welsch, Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring, *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 45-57, 2016.

[130] Q. Lin, C. Hammerschmidt, G. Pellegrino, S. Verwer, Short-term Time Series Forecasting with Regression Automata, *ACM SIGKDD 2016 Workshop on Mining and Learning from Time Series*, 2016.

[131] K. Lee, D. W. Kim, K. H. Lee, D. Lee, Density-induced support vector data description, *Neural Networks*, vol. 18, no. 1, pp. 284-289, 2007.

[132] L. J. Latecki, A. Lazarevic, D. Pokrajac, Outlier detection with kernel density functions, *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61-75, 2007.

[133] K. L. Lange, R. J. Little, J. M. Taylor, Robust statistical modeling using the t distribution, *Journal of the American Statistical Association*, vol. 84, no. 408, pp. 881-896, 1989.

[134] X. Liu, Q. Lin, S. Verwer, D. Jarnikov, Anomaly detection in a digital video broadcasting system using timed automata, *ACM/IEEE Symposium on Logic in Computer Science Workshop on Learning and Automata*, 2017.

[135] G. Liu, Z. Lin, Y. Yu, Robust subspace segmentation by low-rank representation, *International Conference on Machine Learning*, pp. 663-670, 2010.

[136] Y. H. Liu, Y. C. Liu, Y. J. Chen, Fast support vector data descriptions for novelty detection, *Neural Networks*, vol. 21, no. 8, pp. 1296-1313, 2010.

[137] U. Lindqvist, P.A. Porras, Detecting computer and network misuse with the production-based expert system toolset (P-BEST), *IEEE Symposium on Security and Privacy*, pp. 146-161, 1999.

[138] J. Li, W. Pedrycz, I. Jamal, Multivariate time series anomaly detection: A framework of hidden markov models, *Applied Soft Computing*, vol. 60, pp. 229-240, 2017.

[139] M. Lichman, P. Smyth, Modeling human location data with mixtures of kernel densities, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 35-44, 2014.

[140] D. D. Lee, S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, vol. 401, no. 6755, pp. 788-791, 1999.

[141] D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, *Advances in Neural Information Processing*, pp. 556-562, 2001.

[142] T. Le, D. Tran, W. Ma, D. Sharma, Fuzzy multi-sphere support vector data description, *IEEE International Conference on Fuzzy Systems*, pp. 1-5, 2012.

[143] T. Le, D. Tran, W. Ma, D. Sharma, A unified model for support vector machine and support vector data description, *International Joint Conference on Neural Networks*, pp. 1-8, 2012.

[144] F. T. Liu, K. M. Ting, Z. H. Zhou, Isolation-based anomaly detection, *ACM Transactions on Knowledge Discovery From Data*, vol. 6, no. 1, pp. 3-39, 2012.

[145] Y. Liao, V. R. Vemuri, Use of k-nearest neighbor classifier for intrusion detection, *Computers & security*, vol. 21, no. 5, pp. 439-448, 2002.

[146] B. Liu, Y. Xiao, L. Cao, Z. Hao, F. Deng, SVDD-based outlier detection on uncertain data, *Knowledge and Information Systems*, vol. 34, no. 3, pp. 597-618, 2013.

[147] MathWorks, https://uk.mathworks.com/products/matlab.html, 2018.

[148] P. C. Mahalanobis, On the generalised distance in statistics, *National Institute of Sciences of India*, vol. 2, no. 1, pp. 49-55, 1936.

[149] O. Mazhelis, One-class classifiers - a review and analysis of suitability in the context of mobile-masquerader detection, *Revue Africaine de la Recherche en Informatique et Mathematiques Appliquees INRIA*, vol. 6, pp. 29-48, 2007.

[150] A. J. McConnell, Applications of tensor analysis, *Courier Corporation*, 2014.

[151] R. Mazumder, T. Hastie, R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, *Journal of Machine Learning Research*, vol. 11, pp. 2287-2322, 2009.

[152] X. Miao, K. Liu, Y. He, D. Papadias, Q. Ma, Y. Liu, Agnostic diagnosis: discovering silent failures in wireless sensor networks, *IEEE Transactions on Wireless Communications*, vol. 12, no. 12, pp. 6067-6075, 2013.

[153] M. Marsden, K. McGuinness, S. Little, N. E. O'Connor, Holistic features for real-time crowd behaviour anomaly detection, *IEEE International Conference on Image Processing (ICIP)*, pp. 918-922, 2016.

[154] T. Mu, A. K. Nandi, Multiclass classification based on extended support vector data Description, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 5, pp. 1206-1216, 2009.

[155] M. X. Ma, H. Y. T. Ngan, W. Liu, Density-based outlier detection by local outlier factor on largescale traffic data, *Electronic Imaging*, vol. 14, pp. 1-4, 2016.

[156] J. Ma, S. Perkins, Online novelty detection on temporal sequences, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 613-618, 2003.

[157] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, *International Joint Conference on Neural Network*, pp. 1741-1745, 2003.

[158] C. Manikopoulos, S. Papavassiliou, Network intrusion and fault detection: a statistical anomaly approach, *Communications Magazine*, vol. 40, no. 10, pp. 76–82, 2002.

[159] S. Mahadevan, S. L. Shah, Fault detection and diagnosis in process data using one-class support vector machines, *Journal of Process Control*, vol. 19, no. 10, pp. 1627-1639, 2009.

[160] Y. Ma, H. Shi, H. Ma, M. Wang, Dynamic process monitoring using adaptive local outlier factor, *Chemometrics and Intelligent Laboratory Systems*, vol. 127, pp. 89-101, 2013.

[161] J. Ma, J. Theiler, S. Perkins, Accurate on-line support vector regression, *AAAI Fall Symposium Artificial Intelligence*, vol. 15, no. 11, pp. 2683-2703, 2003.

[162] X. Mu, K. M. Ting, Z. H. Zhou, Classification under streaming emerging new classes: a solution using completely-random trees, *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605-1618, 2017.

[163] Netflix, Surus, *https://github.com/Netflix/Surus*, 2015.

[164] K. Noto, C. Brodley, D. Slonim, Anomaly detection using an ensemble of feature models, *IEEE International Conference on Data Mining*, pp. 953-958, 2010.

[165] H. Nguyen, Z. Shen, Y. Tan, X. Gu, FChain: Toward black-box online fault localization for cloud systems, *International Conference on Distributed Computing Systems*, pp. 21-30, 2013.

[166] G. Orair, C. Teixeira, Y. Wang, W. Jr., S. Parthasarathy, Distance-based outlier detection - consolidation and renewed bearing, *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1469-1480, 2010.

[167] E. Parzen, On estimation of a probability density function and mode, *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065-1076, 1962.

[168] F. Pukelsheim, The three sigma rule, *American Statistician*, vol. 48, no. 2, pp. 88-91, 1994.

[169] V. Paxson, Bro: a system for detecting network intruders in real-time, *Computer networks*, vol. 31, no. 23-24, pp. 2435-2463, 1999.

[170] S. J. Peter, Local distance-based outlier removal for scattered data through minimum spanning tree. *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 16, no. 2, pp. 149-161, 2013.

[171] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, *Signal Processing*, vol. 99, pp. 215-249, 2014.

[172] N. H. Packard, I. P. Crutchfield, J. D. Farmer, R. S. Shaw, Geometry from a Time Series, *Physical Review Letters*, vol. 45, pp. 712-716, 1980.

[173] P. Papadimitriou, A. Dasdan, H. Garcia-Molina, Web graph similarity for anomaly detection, *Journal of Internet Services and Applications*, vol. 1, no.1, pp. 19-30, 2010.

[174] V. Petridis, A. Kehagias, L. Petrou, A. G. Bakirtzis, S. Kiartzis, H. Panagiotou, N. Maslaris, A bayesian multiple models combination method for time series prediction, *Journal of Intelligent and Robotic Systems*, vol. 31, no. 1-3, pp. 69-89, 2001.

[175] H. Paulheim, R. Meusel, A decomposition of the outlier detection problem into a set of supervised learning problems, *Machine Learning*, vol. 100, no. 2-3, pp. 509-531, 2015.

[176] A. Patcha, J. Park, An overview of anomaly detection techniques - existing solutions and latest technological trends, *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[177] N. Pham, R. Pagh, A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data, *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 877–885, 2012.

[178] M. R. Pillutla, N. Raval, P. Bansal, K. Srinathan, C. V. Jawahar, LSH based outlier detection and its application in distributed setting, *ACM International Conference on Information and Knowledge Management*, pp. 2289-2292, 2011.

[179] D. Pechyony, V. Vapnik, On the theory of learnining with privileged information, *Advances in Neural Information Processing Systems*, pp. 1894-1902, 2010.

[180] A. A. Qahtan, B. Alharbi, S. Wang, X. Zhang, A PCA-based change detection framework for multidimensional data streams: change detection in multidimensional data streams, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 935–944, 2015.

[181] H. Qiu, Y. Liu, N. A. Subrahmanya, W. Li, Granger causality for time-series anomaly detection, *IEEE International Conference on Data Mining*, pp. 1074-1079, 2012.

[182] J. A. Quinn, M. Sugiyama, A least-squares approach to anomaly detection in static and sequential data, *Pattern Recognition Letters*, vol. 40, pp. 36-40, 2014.

[183] C. E. Rasmussen, The infinite Gaussian mixture model, *Advances in Neural Information Processing Systems*, pp. 554-560, 2000.

[184] S. Reece, R. Garnett, M. Osborne, S. Roberts, Anomaly detection and removal using non-stationary Gaussian processes, *arXiv preprint*, arXiv:1507.00566, 2015.

[185] P. Rousseeuw, A. Leroy, Robust regression and outlier detection, *John Willey & Sons*, 2005.

[186] K. Riesen, M. Neuhaus, H. Bunke, Graph embedding in vector spaces by means of prototype selection, *International Workshop on Graph-Based Representations in Pattern Recognition*, pp. 383-393, 2007.

[187] M. Radovanovic, A. Nanopoulos, M. Ivanovic, Reverse nearest neighbors in unsupervised distance-based outlier detection, *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1369–1382, 2015.

[188] S. Ramaswamy, R. Rastogi, K. Shim, Efficient algorithms for mining outliers from large data sets, *ACM SIGMOD Record*, vol. 29, no. 2, pp. 427-438, 2000.

[189] I. Ruts, P. J. Rousseeuw, Computing depth contours of bivariate point clouds, *Computational Statistics & Data Analysis*, vol. 23, no. 1, pp. 153-168, 1996.

[190] S. Rayana, W. Zhong, L. Akoglu, Sequential ensemble learning for outlier detection: A bias-variance perspective, *IEEE International Conference on Data Mining*, pp. 1167-1172, 2016.

[191] S. Sathe, C. C. Aggarwal, Subspace outlier detection in linear time with randomized hashing, *IEEE International Conference on Data Mining*, pp. 459-468, 2016.

[192] M. Sugiyama, K. M. Borgwardt, Rapid distance-based outlier detection via sampling, *Advances in Neural Information Processing Systems*, pp. 467-475, 2013.

[193] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, *Cambridge: MIT press*, vol. 1, no. 1, 1998.

[194] M. Slaney, M. Casey, Locality-sensitive hashing for finding nearest neighbors, *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 128-131, 2008.

[195] A. Servin, D. Kudenko, Multi-agent reinforcement learning for intrusion detection, *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pp. 211-223, 2008.

[196] N. Shahid, I. H. Naqvi, S. Bin Qaisar, One-class support vector machines: analysis of outlier detection for wireless sensor networks in harsh environments, *Artificial Intelligence Review*, vol. 43, no. 4, pp. 515-563, 2015.

[197] V. Sharmanska, N. Quadrianto, C. H. Lampert, Learning to rank using privileged information, *IEEE International Conference on Computer Vision*, pp. 825-832, 2013.

[198] V. A. Sindagi, S. Srivastava, Domain adaptation for automatic OLED panel defect detection using adaptive support vector data description, *International Journal of Computer Vision*, vol. 122, no. 2, pp. 193-211, 2017.

[199] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, X. Xu, DBSCAN revisited, revisited - why and how you should (still) use DBSCAN, *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1-21, 2017.

[200] B. Song, H. Shi, Y. Ma, J. Wang, Multisubspace principal component analysis with local outlier factor for multimode process monitoring, *Industrial and Engineering Chemistry Research*, vol. 53, no. 42, pp. 16453-16464, 2014.

[201] B. Schölkopf, R. C. Williamson, A. J. Smola, Support vector method for novelty detection, *Advances in Neural Information Processing Systems*, pp. 582-588, 1999.

[202] B. Schölkopf, A. Smola, K. R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.

[203] H. Sartipizadeh, T. L. Vincent, Computing the approximate convex hull in high dimensions, *arXiv preprint*, arXiv:1603.04422, 2016.

[204] E. Schubert, A. Zimek, H. P. Kriegel, Generalized outlier detection with flexible kernel density estimates, *SIAM International Conference on Data Mining*, pp. 542-550, 2014.

[205] Twitter, AnomalyDetection, *https://github.com/twitter/AnomalyDetection*, 2015.

[206] J. Tang, Z. Chen, A. W. C. Fu, D. W. Cheung, Enhancing effectiveness of outlier detections for low density patterns, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535-548), 2002.

[207] D. M. J. Tax, DDtools - the data description toolbox for Matlab, *http://prlab.tudelft.nl/david-tax/dd_ tools.html*, version 2.1.2, 2015.

[208] D. M. J. Tax, R. P. W. Duin, Support vector data description, *Machine Learning*, vol. 54, no. 1, pp. 45-66, 2004.

[209] D. M. J. Tax, R. P. W. Duin, Support vector domain description, *Pattern Recognition Letters*, vol. 20, no. 11, pp. 1191-1199, 1999.

[210] D. M. J. Tax, P. Laskov, Online SVM learning: from classification to data description and back, *IEEE Workshop on Neural Networks for Signal Processing*, pp. 499-508, 2003.

[211] C. Tao, Y. Ge, Q. Song, Y. Ge, O. A. Omitaomu, Metric ranking of invariant networks with belief propagation, *IEEE International Conference on Data Mining*, pp. 1001-1006, 2014.

[212] T. N. Tran, K. Drab, M. Daszykowski, Revised DBSCAN algorithm to cluster data with dense adjacent clusters, *Chemometrics and Intelligent Laboratory Systems*, vol. 120, pp. 92-96, 2013.

[213] H. M Tran, J. Schönwälder, DisCaRia - Distributed case-based reasoning system for fault management, *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, pp. 540-553, 2015.

[214] H. M Tran, J. Schönwälder, Fault representation in case-based reasoning, *International Workshop on Distributed Systems: Operations and Management*, pp. 50-61, 2007.

[215] K. Teeyapan, N. Theera-Umpon, S. Auephanwiriyakul, Ellipsoidal support vector data description, *Neural Computing and Applications*, vol. 28, no. 1, pp. 337-347, 2017.

[216] S. C. Tan, K. M. Ting, F. T. Liu, Fast anomaly detection for streaming data, *International Joint Conference on Artificial Intelligence*, vol. 22, no. 1, pp. 1511, 2011.

[217] K. M. Ting, Y. Zhu, M. Carman, Y. Zhu, Z. H. Zhou, Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1205-1214, 2016.

[218] K. M. Ting, T Washio, J. R. Wells, F. T. Liu, Density estimation based on mass, *IEEE International Conference on Data Mining*, pp. 715-724, 2011.

[219] J. Tian, L. Zhu, S. Zhang, L. Liu, Improvement and parallelism of k-means clustering algorithm, *Tsinghua Science & Technology*, vol. 10, no. 3, pp.277-281, 2005.

[220] P. E. Utgoff, Incremental induction of decision trees, *Machine learning*, vol. 4, no. 2, pp. 161-186, 1989.

[221] J. Vanderplas, mst_clustering: Clustering via Euclidean minimum spanning trees, *The Journal of Open Source Software*, vol. 1, no. 1, 2016.

[222] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, A. Mislove, Towards detecting anomalous user behavior in online social networks, *USENIX Security Symposium*, pp. 223-238, 2014.

[223] N. X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: is a correction for chance necessary?, *International Conference on Machine Learning*, pp. 1073-1080, 2009.

[224] O. Vallis, J. Hochenbaum, A. Kejariwal, A novel technique for long-term anomaly detection in the cloud, *USENIX Workshop on Hot Topics in Cloud Computing*, 2014.

[225] A. R. Vasudevan, S. Selvakumar, Local outlier factor and stronger one class classifier based hierarchical model for detection of attacks in network intrusion detection dataset, *Frontiers of Computer Science*, vol. 10, no. 4, pp. 755-766, 2016.

[226] R. A. Vandermeulen, C. Scott, Robust kernel density estimation by scaling and projection in Hilbert space, *Advances in Neural Information Processing Systems*, pp. 433-441, 2014.

[227] V. Vapnik, R. Izmailov, Learning using privileged information-similarity control and knowledge transfer, *Journal of Machine Learning Research*, vol. 16, no. 55, pp. 2023-2049, 2015.

[228] V. Vapnik, A. Vashist, A new learning paradigm: Learning using privileged information, *Neural Networks*, vol. 22, no. 5, pp. 544-557, 2009.

[229] J. S. Walker, Fast fourier transforms, *CRC press*, 2017.

[230] S. X. Wu, W. Banzhaf, The use of computational intelligence in intrusion detection systems: A review, *Applied Soft Computing*, vol. 10, no. 1, pp. 1-35, 2010.

[231] D. J. Weller-Fahy, B. J. Borghetti, A. A. Sodemann, A survey of distance and similarity measures used within network intrusion anomaly detection, *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 70-91, 2015.

[232] D. Wang, R. Hao, D. Lee, Fault detection in rule-based software systems, *Information & Software Technology*, vol. 45, no. 12, pp. 865-871, 2003.

[233] M. Wu, C. Jermaine, Outlier detection by sampling with accuracy guarantees, *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 767-772, 2006.

[234] Y. Wang, S. Parthasarathy, S. Tatikonda, Locality sensitive outlier detection: a ranking driven approach, *IEEE International Conference on Data Engineering*, pp. 410-421, 2011.

[235] D. Wang, H. Qiao, B. Zhang, M. Wang, Online support vector machine based on convex hull vertices selection, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 593-609, 2013.

[236] C. Williams, C. E. Rasmussen, Gaussian processes for machine learning, *MIT Press*, 2006.

[237] M. Wytock, S. Salapaka, M. Salapaka, Preventing cascading failures in microgrids with one-sided support vector machines, *IEEE Annual Conference on Decision and Control*, pp. 3252-3258, 2014.

[238] K. Wang, H. Xiao, Y. Fu, Ellipsoidal support vector data description in kernel PCA subspace, *International Conference on Digital Information Processing, Data Mining, and Wireless Communications*, pp. 13-18, 2016.

[239] M. Wu, J. Ye, A small sphere and large margin approach for novelty detection using training data with outliers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 2088-2092, 2009.

[240] Y. X. Wang, Y. J. Zhang, Nonnegative matrix factorization - a comprehensive review, *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336-1353, 2013.

[241] K. Wu, K. Zhang, W. Fan, A. Edwards, P. S. Yu, RS-forest - a rapid density estimator for streaming anomaly detection, *IEEE International Conference on Data Mining*, pp. 600-609, 2014.

[242] X. Xu, Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies, *Applied Soft Computing*, vol. 10, no. 3, pp. 859-867, 2010.

[243] X. Xu, L. Zuo, Z. Huang, Reinforcement learning algorithms with function approximation: recent advances and applications, *Information Sciences*, vol. 261, pp. 1-31, 2014.

[244] L. Xiong, X. Chen, J. G. Schneider, Direct robust matrix factorizatoin for anomaly detection, *IEEE International Conference on Data Mining*, pp. 844-853, 2011.

[245] H. Xu, C. Caramanis, S. Sanghavi, Robust PCA via outlier pursuit, *Advances in Neural Information Processing Systems*, pp. 2496-2504, 2010.

[246] E. P. Xing, A. Y. Ng, M. I. Jordan, S. J. Russell, Distance metric learning with application to clustering with side-information, *Advances in Neural Information Processing Systems*, pp. 521-528, 2003.

[247] M. Xie, S. Han, B. Tian, S. Parvin, Anomaly detection in wireless sensor networks: a survey, *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1302–1325, 2011.

[248] Y. Xiao, B. Liu, L. Cao, X. Wu, C. Zhang, Z. Hao, F. Yang, J. Cao, Multi-sphere support vector data description for outliers detection on multi-distribution Data, *IEEE International Conference on Data Mining Workshops*, pp. 82-87, 2009.

[249] Yahoo, S5-A labeled anomaly detection dataset, version 1.0, *http://webscope. sandbox.yahoo.com/catalog.php?datatype=s&did=70*, 2015.

[250] N. Ye, A markov chain model of temporal behavior for anomaly detection, *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, pp. 166-169, 2000.

[251] R. Yu, H. Qiu, Z. Wen, C. Lin, Y. Liu, A survey on social media anomaly detection, *SIGKDD Explorations*, vol. 18, no. 1, pp. 1–14, 2016.

[252] L. Zhao, Z. Chen, Y. Hu, G. Min, Z. Jiang, Distributed feature selection for efficient economic big data analysis, *IEEE Transactions on Big Data*, vol. 13, no. 9, pp. 2332-7790, 2016.

[253] A. Zimek, R. J. G. B. Campello, J. Sander, Ensembles for unsupervised outlier detection - challenges and research questions a position paper, *SIGKDD Explorations*, vol. 15, no. 1, pp. 11-22, 2013.

[254] X. Zhang, W. Dou, Q. He, R. Zhou, C. Leckie, R. Kotagiri, Z. Salcic, LSHiForest: a generic framework for fast tree isolation based ensemble anomaly analysis, *IEEE International Conference on Data Engineering*, pp. 983-994, 2017.

[255] Y. Zhang, B. Du, L. Zhang, S. Wang, A low-rank and sparse matrix decomposition-based Mahalanobis distance method for hyperspectral anomaly detection, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1376-1389, 2016.

[256] H. Zhao, Y. Fu, Dual-regularized multi-view outlier detection, *International Joint Conference on Artificial Intelligence*, pp. 4077-4083, 2015.

[257] L. Zhang, J. Lin, R. Karim, Adaptive kernel density-based anomaly detection for nonlinear systems, *Knowledge-Based Systems*, vol. 139, pp. 50-63, 2018.

[258] Z. Zhou, X. Li, J. Wright, E. Candes, Y. Ma, Stable principal component pursuit, *IEEE International Symposium on Information Theory*, pp. 1518-1522, 2010.

[259] Y. L. Zhang, L. Li, J. Zhou, X. Li, Z. H. Zhou, Anomaly detection with partially observed anomalies, *International World Wide Web Conference*, pp. 639-646, 2018.

[260] Y. Zhang, H. Lu, L. Zhang, X. Ruan, S. Sakai, Video anomaly detection based on locality sensitive hashing filters, *Pattern Recognition*, vol. 59, pp. 302-311, 2016.

[261] J. Zhang, H. Wang, Detecting outlying subspaces for high-dimensional data - the new task, algorithms, and performance, *Knowledge and Information Systems*, vol. 10, no. 3, pp. 333-355, 2006.

[262] W. Zhu, P. Zhong, A new one-class SVM based on hidden information, *Knowledge-Based Systems*, vol. 60, pp. 35-43, 2014.