Universal Anomaly Detection and Applications

by

Sehong Oh

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Electrical and Computer Engineering)
in The University of Michigan
2023

Doctoral Committee:

Professor Alfred O. Hero, Chair
Professor Clayton Scott
Assistant Professor Qing Qu

To my family

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Alfred Hero. Without his support, insight, and patience, it would have been impossible to complete this thesis. I am confident that the time spent under the professor's mentorship will be the most valuable period in my life. I would like to thank the other committee members, Prof. Scott and Prof. Qu. I deeply appreciate their teaching in class. Those classes served as the foundation for me to begin my research.

I would like to thank our research group members, especially Dr. Sabeti and Zeyu. Chapter 2 was a joint work with Dr. Sabeti and provided me with the foundation to successfully develop Chapter 3 and Chapter 4. Zeyu always took an interest in my work and helped me with his extensive knowledge.

I would also like to express my gratitude to my friends: Jun, Dongmyeong, Chenyu, and Alan. Without their support, I would not have been able to finish my study at the University of Michigan. I am grateful to my military friends who have supported me unconditionally.

Finally, I want to express my gratitude to my parents, sister, and Elim. Their unwavering support allowed me to study happily and more than anyone else. Without their help, I would not have been able to complete this journey.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES vi

LIST OF TABLES viii

ABSTRACT ix

CHAPTER

I. Introduction 1

1.1 What is anomaly detection? 1
1.2 Why is anomaly detection difficult? 2
1.3 Atypicality 3
1.4 Source coding 4
1.4.1 Huffman Coding 5
1.4.2 Lempel-Ziv78 (LZ78) 6
1.5 Summary of Contributions 7

II. Pattern Dictionary Method for Anomaly Detection 8

2.1 Introduction 8
2.2 Related Works 10
2.3 Framework and Notation 14
2.4 Pattern Dictionary: Design and Properties 15
2.4.1 A Special Case 17
2.4.2 The General Case 18
2.4.3 Pattern Dictionary for Detection (PDD) 21
2.5 Pattern Dictionary-Based Atypicality (PDA) 23
2.5.1 Lempel-Ziv Parser 25
2.5.2 Pattern Dictionary Parser versus $LZ78$ Parser . . . 36

2.5.3 Atypicality Criterion for Detection of Anomalous Subsequences 37
2.6 Experiments 38
2.6.1 Anomaly Detection in Mackey-Glass Time Series . . 38
2.6.2 Infection Detection Using Physiological Signals . . . 43
2.7 Conclusions 45

III. Pattern Dictionary Method with Clustering 48

3.1 Introduction 48
3.2 Multiple Sample Anomaly Detection 49
3.2.1 Mackey Glass Experiments 49
3.2.2 Results of Clustering 2-dimensional Scores 49
3.2.3 Results of Clustering High-dimensional Scores . . . 53
3.3 Leave-one-out (L1O) kNNG 55
3.3.1 Introduction 55
3.3.2 Methodology 55
3.3.3 Experiments and Results 56
3.4 Conclusion 57

IV. Application: Pattern Dictionary Method for Image Data . . 60

4.1 Introduction 60
4.2 Related Works 61
4.3 Feature Extraction 61
4.4 Experiments and Results 62
4.4.1 MNIST [1] and CIFAR10 [2] 62
4.4.2 Cats and No cats 64
4.5 Conclusion 64

V. Conclusion 67

BIBLIOGRAPHY 68

LIST OF FIGURES

Figure

1.1 Binary classification vs. Anomaly detection 2

1.2 A simple example of Huffman coding 5

1.3 A simple example of Lempel-Ziv78 algorithm 7

2.1 Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed alphabet size $|\mathcal{X}| = 100$). 18

2.2 Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed $D_{\max} = 20$) 19

2.3 Plot of the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of an $|\mathcal{X}|$-ary sequence of length l. 28

2.4 Simulation results compared to the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of binary sequences of length l generated by sources with three different source probabilities $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases. . 29

2.5 Similar to Figure 2.4 , the number of distinct phrases resulting from LZ78-parsing of binary sequences of fixed length $l = 1000$ varies over the source probability parameter $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases. 32

2.6 Simulation of the probability-dependent upper bound $c(l, p)$ for binary sequences of fixed length $l = 100$ with various probability parameters $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases. 35

2.7 Mackey-Glass time series: the training data (top) and an example of the test data (bottom) in which samples in $[501, 1000]$ are anomalous (shown in red). 39

2.8 The effect of maximum dictionary depth $D_{\max}$ on parsing and detection sensitivity (true positive rate) of the Mackey-Glass time series presented in Figure 2.7. 40
2.9 Anomaly detection using the proposed PDA method for a subject based on heart rate and temperature data collected from a wearable wrist sensor. Anomalies are shown in red in (a,b). (c) shows the subject's infection level. 45
2.10 Anomaly detection using the proposed PDA method for a subject who had a mild infection with low level of viral shedding based on heart rate and temperature data collected from a wearable wrist sensor. Note that no anomaly has been detected: (a) heart rate, (b) temperature, and (c) infection level. 46
3.1 Train data is generated by Mackey Glass equation 50
3.2 Complex anomaly cases (Anomalies are embedded from 501 to 1000) 50
3.3 Simple anomaly cases (Anomalies are embedded from 501 to 1000) . 51
3.4 Accuracy and F1 score of the case in Figure 3.2 (a) based on time window size 51
3.5 Accuracy and F1 score of the case in Figure 3.2 (b) based on time window size 52
3.6 Accuracy and F1 score of the case in Figure 3.3 (a) based on time window size 52
3.7 Accuracy and F1 score of the case in Figure 3.3 (b) based on time window size 52
3.8 Accuracy and F1 score of the case in Figure 3.2 (a) based on the number of dimensions 53
3.9 Accuracy and F1 score of the case in Figure 3.2 (b) based on the number of dimensions 54
3.10 Accuracy and F1 score of the case in Figure 3.3 (a) based on the number of dimensions 54
3.11 Accuracy and F1 score of the case in Figure 3.3 (b) based on the number of dimensions 54
3.12 Accuracy of four cases based on the number of dimensions for different values of the kNN parameter k in the L1O-kNNG. The number of samples = 300. 57
3.13 F1-score of four cases based on the number of dimensions for different values of the kNN parameter k in the L1O-kNNG. The number of samples = 300. 57
4.1 Resnet50, 101, 152 Architecture and feature extraction 62
4.2 ROC AUCs based on the number of training samples. The number of samples is from 5 to 50 (left) and from 500 to 2000 (right). One label is set as normal data and the others are considered as anomalies. 64
4.3 Cats: Detected as normal images (left) and anomalous images (right) 65
4.4 Indoor Scene: Detected as normal images (left) and anomalous images (right) 65

LIST OF TABLES

Table

2.1 Filling (training) the dictionary (of maximum depth $D_{\max} = 3$) with the patterns in the training sequence ABACADABBACCADDABABACADAB. 22

2.2 Comparison of anomaly detection methods ($\mu \pm \sigma$ representation is used where μ is the mean and σ is the standard deviation). The proposed PDA method attains overall best performance (bold entries of table). 42

2.3 Comparison of anomaly detection methods for different cases of data resolutions: high resolution corresponds to an alphabet size of 90 , medium resolution corresponds to an alphabet size of 45 , and low resolution corresponds to an alphabet size of 10 . In this table, $\mu \pm \sigma$ representation is used where μ is the mean and σ is the standard deviation. The proposed PDA method achieves overall best performance (bold entries of table). 43

2.4 Table 4. Comparison of running time (in second) of PD-based method and two implementations of the ZM method for different cases of data resolutions: high resolution corresponds to an alphabet size of 90 , medium resolution corresponds to an alphabet size of 45 , and low resolution corresponds to an alphabet size of 10. This experiment is performed on a Hansung laptop with 2.60GHzCPU, 500 GB of SSD, and 16 GB of RAM using MATLAB R2021a. The proposed PD-based method has fastest run time overall (bold entries in table). 44

3.1 Comparison by accuracy 58

3.2 Comparison by F1-score 58

3.3 ROC AUC of PDA (Chapter 2) and L1O-kNNG (this Chapter) for the case shown in Figure 3.2.(a). The resolution is set the same as the previous experiment in Table 2.3 58

4.1 ROC AUC(% and Std) with MNIST (over 10 seeds) per method . . 62

4.2 ROC AUC(% and Std) with CIFAR10 (over 10 seeds) per method. . 63

ABSTRACT

Universal anomaly detection and applications

by

Sehong Oh

Chair: Alfred O. Hero

Anomaly detection is important in many research areas including fraud detection and biological change detection. However, anomaly detection is a difficult task due to the lack of anomalies available for training. In this thesis, we propose a compression-based nonparametric anomaly detection method for time series and image data using a pattern dictionary. This method constructs two features (typicality and atypicality) to distinguish anomalies based on normal training data captured in a tree-structured data structure. The typicality of a test sequence is a measure of how well the data can be compressed by the pattern dictionary. The typicality can be used as an anomaly score to detect anomalous data at a certain threshold. The atypicality of a sequence is a measure of compressibility of the test data by a universal source coder, determined independently of training data. The typicality and the atypicality of each sub-sequence in the test sequence are complementary and anomalous deviations can be determined by combining them. Several methods are evaluated for aggregating these measures. These include a scalarized of the typicality and atypicality score, a 2-dimensional (typicality and atypicality) score, and a high-dimensional score.

CHAPTER I

Introduction

It is hard to define what an anomaly is. However, it is obvious that anomalies exist in various field across the world, especially in engineering and science. Detection of anomalies is thus important in many fields and various anomaly detection methods have been developed. An anomaly is generally defined as patterns that deviate significantly from a normal distribution [3]. The motivation for this thesis is to identify patterns in the data that are indicative of an anomaly. A pattern dictionary method is presented that uses a measure of typicality and atypicality, two information theoretic notions used in compression algorithms, specifically the Huffman coder and the Lempel-Ziv encoder. We apply our method not only to time series data (Ch.2, 3) but also to image data (Ch.4) by using image features. Moreover, by considering multiple patterns of different lengths simultaneously, we show the performance of our model is improved.

1.1 What is anomaly detection?

There is a difference between anomaly detection and binary classification. Anomaly detection seeks to classify anomalous data that significantly deviates from a normal training sample. There is no label for the anomalous sample, so this is often called on-class classification. Binary classification is a supervised learning method that uses

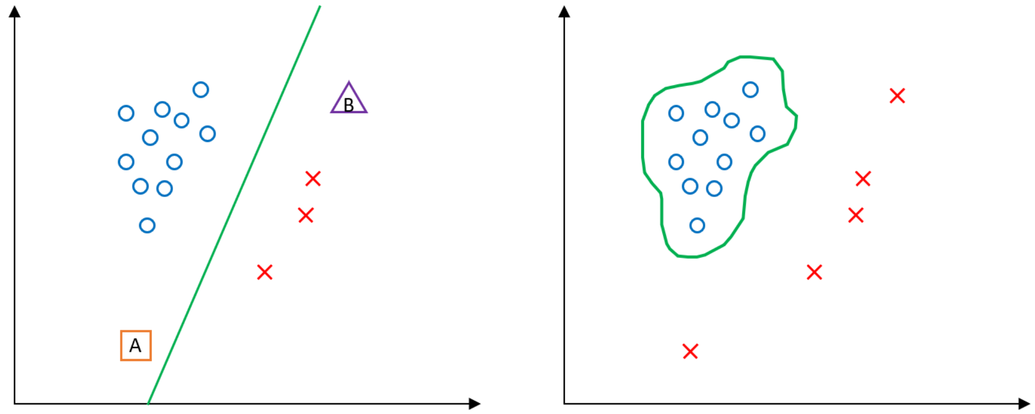


Figure 1.1: Binary classification vs. Anomaly detection

labeled training data for both classes while anomaly detection is an unsupervised learning problem. For illustration see Fig 1.1. There are blue Os and red Xs in two dimensions. In the case of binary classification, the labels ”O” and ”X” are available in the training data. The labels and coordinates of the training data are used to train a classifier with a decision boundary (in green). The classifier is then applied to classify new data ”A’ and ”B”. On the other hand, anomaly detection must classify one class (the ”X” class) without seeing any instances in the training data (the ”O” class). An anomaly detection establishes a region (in green) that represents typical or expected behavior and considers any observation in the dataset that falls outside this region as an anomaly [3].

1.2 Why is anomaly detection difficult?

Anomaly detection might not seem like a difficult task at first, as humans can often distinguish anomalies in a given dataset. However, there are several causes that make anomaly detection challenging [3]:

1. Different domains might have their definition of what constitutes an anomaly, so it can be hard to use methods developed for one domain to another domain.
2. The length and occurrence frequency of potential anomalous data is unknown

in time series.

3. There is extreme data imbalance, with no or few instances of one of the classes of data during training.

1.3 Atypicality

One of the most important concepts exploited in this thesis is atypicality. According to A. Høst-Madsen et al. (2019) in [4], we should start with the theory of randomness developed by Kolmogorov and Martin-Löf to understand this concept [5],[6],[7]. According to Kolmogorov, infinite sequences can be divided as "typical" or "special" and we consider the typical sequences random. Namely, the typical sequences satisfy all laws of probability, whereas the special sequences do not. Kolmogorov complexity is a way of measuring the amount of information contained in data and the more complex data, the higher Kolmogorov complexity. A sequence that consists of bits $\{x_n, n = 1, \ldots, \infty\}$ is random (i.e, i.i.d. uniform) if the Kolmogorov complexity satisfies $K(x_1, \ldots, x_n) \geq n - c$ for all n with some constant c [6][4]. Furthermore, if $K(x_1, \ldots, x_n \mid n) \geq n$ for all n is satisfied, the sequence is incompressible and random [7],[4]. Suppose we generate sequences x^n from an i.i.d. uniform distribution. The identity function is the optimum coder of the sequences with the code length n. Let's assume that we have a (universal) coder that makes the code length less than n. This can be stated as $K(x_1, \ldots, x_n \mid n) < n$. In this case that sequences can be compressed to a length shorter than n, we would not call the sequence "typical" but a "special" sequence. In this thesis, we consider such sequences "atypical". The definition of "atypical" is as follows:

Definition 1. *A sequence is atypical if it can be described (coded) with fewer bits in itself rather than using the (optimum) code for typical sequences* [4].

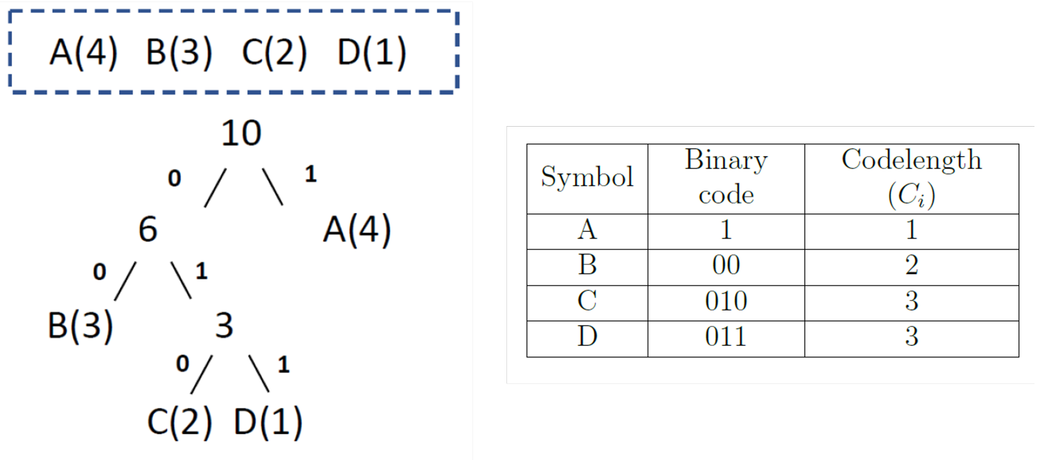
This definition is the key to approaching the atypicality problem. We assume prefix-free codes to understand ”the (optimum) code for typical sequences” based on the principle in [7] The possible methods are Huffman codes, Shannon codes, Shannon-Fano-Elias codes, arithmetic coding, etc. We use Huffman codes in other chapters, focusing only on the code length. The code length for typical encoding is calculated accurately because the variation of the length is within a few bits. On the other hand, “described (coded) with fewer bits in itself” in definition 1 is less clear. We can use Kolmogorov complexity, but it cannot be calculable. Thus, we should use some types of universal source coder instead of the concept of Kolmogorov complexity for a more accurate comparison. A detailed description of atypicality is in [4].

1.4 Source coding

In information theory, source coding is the process of converting a sequence of symbols from information into a sequence of alphabet symbols, which are typically bits (binary digits). The goal of source coding is to represent the original data efficiently while maintaining the desired level of fidelity. Source coding is generally divided into two types as follows [7]:

1. Lossless source coding: Lossless source coding can exactly recover the original symbols from the encoded bits. Namely, we do not lose any information during the data compression and decompression processes. We focus on this type of data compression method in this research and Huffman coding and Lempel-Ziv78 (LZ78) [8] algorithm are mainly used.

2. Lossy source coding: Lossy source coding allows for better data compression compared to lossless source coding. It implies that the original symbols might be distorted when they are decompressed so this type of data compression method



Symbol	Binary code	Codelength (C_i)
A	1	1
B	00	2
C	010	3
D	011	3

Figure 1.2: A simple example of Huffman coding

is used when information loss is acceptable.

1.4.1 Huffman Coding

Huffman coding is an algorithm that compresses data without losing information. Huffman code is an optimal prefix code and compresses data by assigning variable-length codes to each character in the data. This algorithm works by building a binary tree by combining the least frequent symbols into a new node repeatedly. This repeated process assigns shorter code lengths to frequent symbols, whereas the less frequent symbols have longer code lengths. A simple example is shown in figure 1 with a sequence x = ABCDABCABA. The steps are as follows:

1. Create a leaf node for each symbol with the frequency.

2. Arrange all nodes in descending order based on their frequency.

3. Repeat the below sub-steps until there is one node left.

   (a) Remove the last two nodes and create a new internal node with those two nodes

(b) Assign the removed nodes as the left and right children of the new internal node.

(c) Rearrange all nodes in descending order based on the updated frequency.

4. Assign a '0' for every left branch and a '1' for every right branch. The Huffman code for a symbol is the sequence of '0's and '1's encountered along the path from the root to the symbol's leaf node.

1.4.2 Lempel-Ziv78 (LZ78)

The Lempel-Ziv 78 algorithm is a lossless data compression algorithm designed by Abraham Lempel and Jacob Ziv in 1978 [8]. The basic idea of Lempel-Ziv algorithms is subsequences that have already appeared are more likely to be repeated than subsequences that we haven't seen. The algorithm works by creating a dictionary of patterns in the input data and replacing repeated subsequences with an index in a dictionary. The compressed data can be also decompressed to the original data without losing information like Huffman coding.

1. Initialize: Start with an empty dictionary. As you process the input, you will fill the dictionary with new entries.

2. Read input: Look at the input data one symbol at a time.

3. Build the phrase: Begin with an empty phrase. Keep adding symbols from the input to the phrase until you get a combination that isn't in the dictionary yet.

4. Add to the dictionary: Once you have a new phrase, add it to the dictionary and assign it a unique index number.

5. Output the code: For the new phrase, find the longest prefix (the part before the last symbol) that already exists in the dictionary. Output the index number of that prefix along with the last symbol of the new phrase.

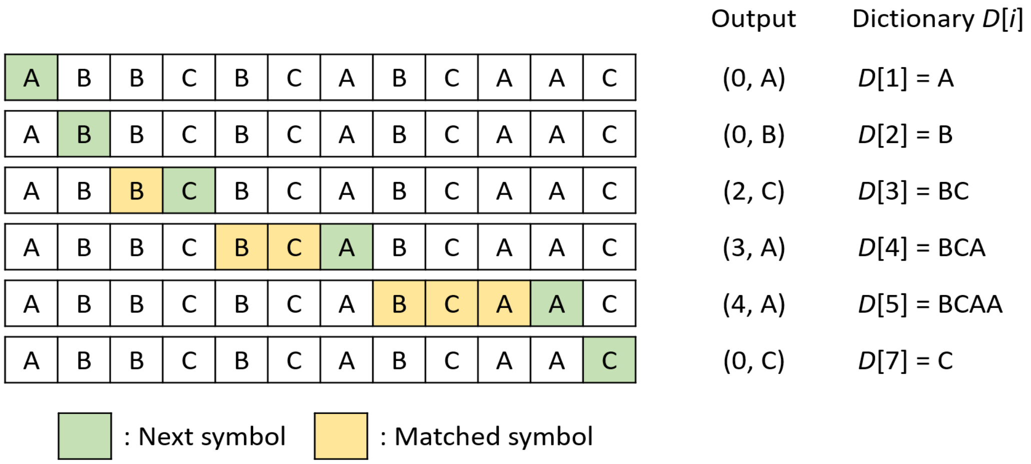


Figure 1.3: A simple example of Lempel-Ziv78 algorithm

6. Repeat: Continue steps 3-5 until you reach the end of the input data.

7. Termination: If there is a remaining phrase at the end of the input that is not in the dictionary, output its prefix's index number and the last symbol, as you did in step 5.

1.5 Summary of Contributions

My contributions are divided into three principal chapters.

1. Chapter 2 was published as the paper [9]. I contributed to Section 2.6 of this paper by doing the simulations that led to the results presented in Tab2.2 - 2.4

2. Chapter 3 describes my extension of the paper [9] that uses the vector-valued pair of typicality and atypicality scores.

3. Chapter 4 describes my application of the methods of chapter 2 to detect anomalous images.

CHAPTER II

Pattern Dictionary Method for Anomaly Detection

2.1 Introduction

Anomaly detection and outlier detection are used for detecting data samples that are inconsistent with normal data samples. Early methods did not take the sequential structure of the data into consideration [3]. However, many real world applications involve data collected as a sequence or time series. In such data, anomalous samples are better characterized as subsequences of time series. Anomaly detection is a challenging task due to the uncertain nature of anomalies. Anomaly detection in time series and sequence data is particularly difficult since both length and occurrence frequency of potentially anomalous subsequences are unknown. Additionally, algorithmic computational complexity can be a challenge, especially for streaming data with large alphabet sizes.

In this paper, we propose a universal nonparametric model-free anomaly detection method for time series and sequence data based on a pattern dictionary (PD). Given training and test data sequences, a pattern dictionary is created from the sets of all the patterns in the training data. This dictionary is then used to sequentially parse and compress (in a lossless manner) the test data sequence. Subsequently, we interpret the number of parsed phrases or the codelength of the test data as anomaly scores. The smaller the number of parsed phrases or the shorter the compressed codelength of the test data, the more similarity between training and test data patterns. This sequential parsing and lossless compression procedure lead to detection of anomalous test sequences and their potential anomalous patterns (subsequences).

The proposed pattern dictionary method has the following properties: (i) it is nonparametric since it does not rely on a family of parametric distributions; (ii) it is universal in the sense that the detection criterion does not require any prior modeling of the anomalies or nominal data; (iii) it is non-Bayesian as the detection criterion is model-free; and (iv) as it depends on data compression, data discretization is required prior to building the dictionary. While the proposed pattern dictionary can be used as a stand-alone anomaly detection method (Pattern Dictionary for Detection (PDD)), we show how it can be utilized in the atypicality framework [10], [11] for more general data discovery problems. This results in a method we call PDA (Pattern Dictionary based Atypicality), in which the proposed pattern dictionary is contrasted against a universal source coder which is the Tree-Structured Lempel–Ziv (LZ78) [7], [8]. We use the LZ78 as the universal encoder since its compression procedure is similar to our proposed pattern dictionary, and it is (asymptotically) optimal [7], [8].

The main contributions of this paper are as follows. First, we propose the pattern dictionary method for anomaly detection and characterize its properties. We show in Theorem 1 that using a multi-level dictionary that separates the patterns by their depth results in a shorter average indexing codelength in comparison to a uni-level dictionary that uses a uniform indexing approach. Second, we develop novel non-asymptotic lower and upper bounds of the LZ78 parser in Theorem 2 and further analyze its non-asymptotic properties. We demonstrate that the non-asymptotic upper bound on the number of distinct phrases resulting from the LZ78 parsing of an $|\mathcal{X}|$-ary sequence of length l can be explicitly expressed by the Lambert W function [12]. To the best of our knowledge, such characterization has not previously appeared in the literature. Then, we show in Lemma 1 that the achieved non-asymptotic upper bound on the number of distinct phrases resulting from the LZ78 parsing converges to the optimal upper bound $\frac{l}{\log l}$ of the LZ78 parser as $l \to \infty$. Third, we show how the pattern dictionary and LZ78 can be used together in an atypicality detection framework. We demonstrate that the achieved non-asymptotic lower and upper bounds on both LZ78 and pattern dictionary determine the range of the anomaly score. Consequently, we show how these bounds can be used to analyze the effect of dictionary depth on the anomaly score. Furthermore, the bounds are used to set the anomaly detection threshold. Finally, we compare our proposed methods with the competing methods, including nearest neighbors-based similarity [13], threshold sequence time-delay embedding [14], [15], [16], [17], and compression-based dissimilarity measure [18], [19], [20], [21], [22], that are designed for anomaly detection in sequence data and time series. We conclude our paper with an experiment that details how the proposed framework can be used to construct a baseline of health against which anomalous deviations are detected.

The paper is organized as follows. In Section 2.2, we briefly review the relevant literature in anomaly detection (readers who are familiar with anomaly detection can skip this section). Section 2.3 introduces the detection framework and the notation used in this paper. Section 2.4 presents our proposed pattern dictionary method and its properties. In Section 2.5, we show how the proposed pattern dictionary can be used in an atypicality framework alongside LZ78, and we analyze the non-asymptotic properties of the LZ78 parser. Section 2.6 presents experiments that illustrate the proposed pattern dictionary anomaly detection procedure. Finally, Section 2.7 concludes our paper.

2.2 Related Works

Anomaly detection has a vast literature. Anomaly detection procedures can be categorized into parametric and nonparametric methods. Parametric methods rely on a family of parametric distributions to model the normal data. The slippage problem [23], change detection [24], [25], [26], [27], concept drift detection [28], minimax quickest change detection (MQCD) [29], [30], [31], and transient detection [32], [33], [34] are examples of parametric anomaly detection problems. The main difference between our proposed pattern dictionary method and the aforementioned techniques is that our method is a model-free nonparametric method. The main drawback of the parametric anomaly detection procedure is that it is difficult to accurately specify the parametric distribution for the data under investigation. Nonparametric anomaly detection approaches do not assume any explicit parameterized model for the data distributions. An example is an adaptive nonparametric anomaly detection approach called geometric entropy minimization (GEM) [35], [36] that is based on the minimal covering properties of K-point entropic graphs constructed on N training samples from a nominal probability distribution. The main difference between GEM-based methods and our proposed pattern dictionary is that former techniques are designed to detect outliers and cannot easily incorporate the temporal information regarding anomaly in a data stream. Another nonparametric detection method is sequential nonparametric testing that considers data as online stream and addresses the growing data storage problem by sequentially testing every new data samples [37], [38]. A key difference between sequential nonparametric testing and our proposed pattern dictionary method is that our method is based on coding theory instead of statistical decision theory.

Information theory and universal source coding have been used previously in anomaly detection [39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. The detection criteria in these approaches are based on comparing metrics such as complexity or similarity distances that depend on entropy rate. An issue with these approaches is that there are many completely dissimilar sources with the same entropy rate, reducing outlier sensitivity. Another related problem is universal outlier detection [51],

[52]. In these works, different levels of knowledge about nominal and outlier distributions and number of outliers are incorporated. Unlike these methods, our proposed pattern dictionary approach does not require any prior knowledge about outliers and anomalies. In [53], a measure of empirical informational divergence between two individual sequences generated from two finite-order, finite-alphabet, stationary Markov processes is introduced and used for a simple universal classification. While the parsing procedure used in [53] is similar to the pattern dictionary used in this paper, there are important differences. The empirical measure proposed in [53] is a stand alone score function that is designed for two-class classification, while our measure is a direct byproduct of the LZ78 encoding algorithm designed for single-class classification, i.e., anomaly detection. In addition, the theoretical convergence of the empirical measure to the relative entropy between the class conditioned distributions, shown in [53], is only guaranteed when the sequences satisfy the finite-order Markov property, a condition that may be difficult to satisfy in practice. In [10], [11], an information theoretic data discovery framework called atypicality has been introduced in which the detection criterion is based on a descriptive codelength comparison of an optimum encoder or a training-based fixed source coder, namely a data-dependent source coder introduced in [10]) with a universal source coder. In this paper, we show how our proposed pattern dictionary method can be used as a training-based fixed source coder in an atypicality framework. Anomaly and outlier detection for time series has also been extensively studied [54]. Various time series modeling techniques such as regression [55], auto regression [56], auto regression moving average [57], auto regressive integrated moving average [58], support vector regression [59], and Kalman filters [60] have been used to detect anomalous observations by comparing the estimated residuals to a threshold. Many of these methods depend on a statistical assumption on the residuals, e.g., an assumption of Gaussian distribution, while the pattern dictionary method is model-free.

The proposed pattern dictionary method is closely related to the anomaly detection methods that are designed for sequence data. Many of these methods are focused on specific applications. For instance, detection of mutations in DNA sequences [13], [61], detection of cyberattacks in computer network [62], and detection of irregular behaviors in online banking [63] are all application-specific examples of anomaly detection for discrete sequences. In the recent years, multiple sequence data anomaly detection methods have been developed specifically for graphs [64], dynamic networks [65], and social networks [66]. Chandola et al. [39] summarized many anomaly detection methods for discrete sequences.

and identified three general approaches to this problem. These anomaly detection formulations are unique in the way that anomalies are defined, but similar in their reliance on comparison between a test (sub)sequence and normal sequences in the training data. For example, kernel-based techniques such as nearest neighbor-based similarity (NNS) [13] are designed to detect anomalous sequences that are dissimilar to the training data. As another example, threshold sequence time-delay embedding (t-STIDE) [14, 15, 16, 17] is established to detect anomalous sequences that contain subsequences with anomalous occurrence frequencies. The compression-based dissimilarity measure (CDM) is proposed for discord detection [18, 19, 20, 21, 22] to detect anomalous subsequences within a long sequence. Chandola et al. [39] also showed how various techniques developed for one problem formulation can be changed and applied to other problem formulations. While our pattern dictionary method shares similarity with NNS, CDM, and t-STIDE, our proposed method is generally applicable to any of the categories of anomaly detection identified in [39]. Furthermore, our detection criterion does not depend on the specific type of anomaly. Note that while CDM is also a compression-based method, its anomaly score is based on a dissimilarity measure that might fail to detect atypical subsequences [10]. For instance, using CDM method, a binary i.i.d. uniform training sequence is equally dissimilar to another binary i.i.d. uniform test sequence or to a test sequence drawn from some other distribution. In Section 2.6, the detection performance of our proposed pattern dictionary method is compared with NNS, CDM, t-STIDE, and the Ziv-Merhav method of [53].

It is worth mentioning that since the proposed pattern dictionary method is based on lossless source coding, it requires discretization of time series prior to deployment. In fact, many anomaly detection approaches require discretization of continuous data prior to applying inference techniques [67, 68, 3, 69, 70]. Note that discretization is also a requirement in other problem settings such as continuous optimization in genetic algorithms [71], image pattern recognition [72], and nonparametric histogram matching over codebooks in computer vision [73].

Anomaly and outlier detection for time series has also been extensively studied [54]. Various time series modeling techniques such as regression [55], auto regression [56], auto regression moving average [57], auto regressive integrated moving average [58], support vector regression [59], and Kalman filters [60] have been used to detect anomalous observations by comparing the estimated residuals to a threshold. Many of these methods depend on a statistical assumption on the residuals, e.g., an assumption of Gaussian distribution, while the pattern dictionary method is model-free.

2.3 Framework and Notation

In the anomaly detection literature for sequence data and time series, the following three general formulations are considered [39]: (i) an entire test sequence is anomalous if it is notably different from normal training sequences; (ii) a subsequence within a long test sequence is anomalous if it is notably different from other subsequences in the same test sequence or the subsequences in a given training sequence; and (iii) a given test subsequence or pattern is anomalous if its occurrence frequency in a test sequence is notably different from its occurrence frequency in a normal training sequence. In this paper, we consider a unified formulation in which we determine if a (sub)sequence is anomalous with respect to a training sequence (or training sequence database) if any of the aforementioned three conditions are met. In other words, given a training sequence or a training sequence database, a test sequence is anomalous if it is significantly different from training sequences, or it contains a subsequence that is significantly different from subsequences in the training sequence, or it contains a subsequence whose occurrence frequency is significantly different from its occurrence frequency in the training data.

Notation We use x to denote a sequence and x_n^m to denote a subsequence of $x: \quad x_n^m = \{x_i, i = n, n+1, \ldots, m\}$, and x^l represents a sequence of length l, i.e., $\{x_n, n = 1, \ldots, l\}$.$\mathcal{X}$ denotes a finite set, and $\mathcal{D}$ represents a dictionary of subsequences. Throughout this paper:

- All logarithms are base 2 unless otherwise is indicated.
- In the encoding process, we always adhere to lossless compression and strict decodability at the decoder.
- While adhering to strict decodability, we only care about the codelength, not the codes themselves

2.4 Pattern Dictionary: Design and Properties

Consider a long sequence, called the training data, $\{x_n, n = 1, \ldots, L\}$ of length L drawn from a finite alphabet $\mathcal{X}$. The goal is to learn the patterns (subsequences) of this sequence by creating a dictionary that contains all distinct patterns of maximum length (depth) $D_{\max} \ll L$ that are embedded in the sequence. We call this dictionary a pattern dictionary $\mathcal{D}$ with the maximum depth $\mathcal{S}_{\mathcal{D}}(x) = \{A, B, C, D, AB, BA, AC, CA,$

$AD, DA, BB, CC, DD\}$.

Example 1. Suppose $D_{\max} = 2$, the alphabet is $\mathcal{X} = \{A, B, C, D\}$ and the training sequence is $x = ABACADABBACCADDABABACADAB$. The set of patterns with depth $d \leq D_{\max}$ in this sequence is $\mathcal{S}_{\mathcal{D}}(x) = \{A, B, C, D, AB, BA, AC, CA, AD, DA, BB, CC, DD\}$.

Since the pattern dictionary is going to be used as a training-based fixed source coder (a data-dependent source coder as defined in [10]), an efficient structure for the pattern representation that minimizes the indexing codelength is of interest. The simplest approach is to consider all the patterns of length $1 \leq d \leq D_{\max}$ in one set $\mathcal{S}_{\mathcal{D}}$ and use a uniform indexing approach. This approach is called a uni-level dictionary. Another approach is to separate all the patterns by their depth (pattern length) and arrange them in $D_{\max}$ sets $\mathcal{S}_{\mathcal{D}}^{(1)}, \mathcal{S}_{\mathcal{D}}^{(2)}, \ldots, \mathcal{S}_{\mathcal{D}}^{(\mathrm{D_{max}})}$, and define $\mathcal{S}_{\mathcal{D}} = \bigcup_{d=1}^{D_{\max}} \mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$, which we call a multi-level dictionary. In the following sections, we show that the latter results in a shorter average indexing codelength. It is worth mentioning that since a multi-level dictionary results in a depthdependent indexing codelength, the average over the depth is considered. A relevant question is if the average of indexing codelength over all the patterns independent of depth should be used as an alternative. Since such pattern dictionaries are used to sequentially parse test data, patterns at smaller depth are more likely to be matched, even if they are anomalous. Thus, the average of indexing codelength over depth can better differentiate depth-dependent anomalies.

2.4.1 A Special Case

Suppose all the possible patterns of depth $d \leq D_{\max}$ exist in the training sequence $\{x_n, n = 1, \ldots, L\}$. That is, the cardinality of $\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$ is $\left|\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}\right| = |\mathcal{X}|^d$ for $1 \leq d \leq D_{\max}$. Then, the total number of patterns is

$$\begin{aligned}\left|\mathcal{S}_{\mathcal{D}}\left(x_1^L\right)\right| &= \sum_{d=1}^{D_{\max}} \left|\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}\left(x_1^L\right)\right| \\ &= \sum_{d=1}^{D_{\max}} |\mathcal{X}|^d \\ &= \frac{|\mathcal{X}|\left(|\mathcal{X}|^{D_{\max}} - 1\right)}{|\mathcal{X}| - 1}\end{aligned}$$

Hence, a uni-level dictionary results in a uniform indexing codelength of

$$\begin{aligned}L^{uni} &= \log\left(\frac{|\mathcal{X}|\left(|\mathcal{X}|^{D_{\max}} - 1\right)}{|\mathcal{X}| - 1}\right) \\ &\approx D_{\max}\log(|\mathcal{X}|).\end{aligned}$$

On the other hand, a multi-level dictionary requires a two-stage description of index. The first stage is the index of the depth d (using $\log D_{\max}$ bits), and the second stage is the index of the pattern among all the patterns with the same depth (using $d\log(|\mathcal{X}|)$ bits). This two-stage description of the index leads to a non-uniform indexing of codelength: the minimum indexing codelength occurring for the patterns of depth $d = 1$ equals to $L_{\min}^{multi} = \log D_{\max} + \log(|\mathcal{X}|)$ bits, while the maximum indexing codelength occurring for the patterns of depth $d = D_{\max}$ equals to $L_{\max}^{\mathrm{multi}} = \log D_{\max} + D_{\max}\log(|\mathcal{X}|)$ bits. Thus, the average indexing codelength of a multi-level dictionary

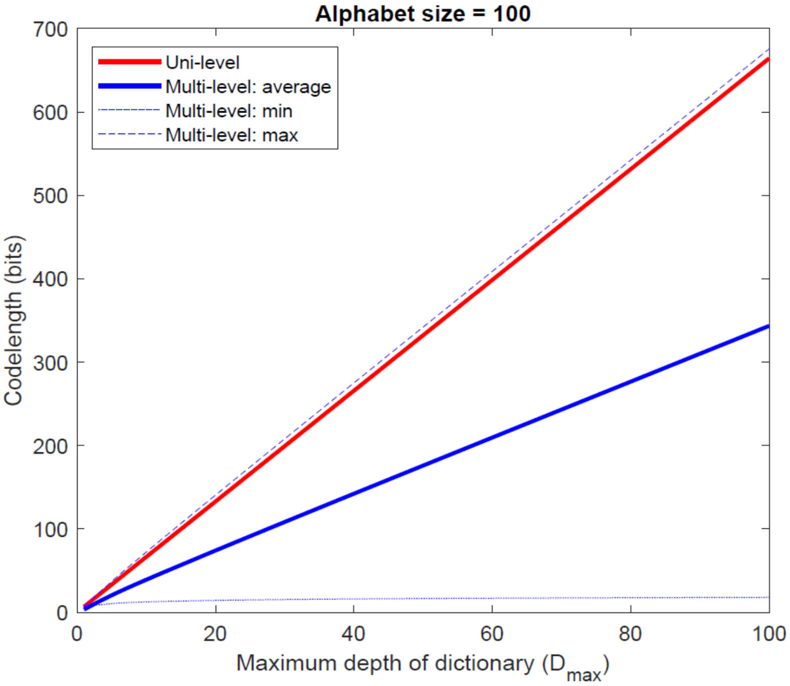


Figure 2.1: Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed alphabet size $|\mathcal{X}| = 100$).

is given by

$$\begin{aligned} L^{\text{multi}} &= \frac{1}{D_{\max}} \sum_{d=1}^{D_{\max}} (\log D_{\max} + d \log(|\mathcal{X}|)) \\ &= \log D_{\max} + \frac{\log(|\mathcal{X}|)}{D_{\max}} \sum_{d=1}^{D_{\max}} d \\ &\approx \log D_{\max} + \frac{1}{2} D_{\max} \log(|\mathcal{X}|) \end{aligned}$$

Figures 2.1 and 2.2 graphically compare the indexing codelength between a uni-level dictionary and a multi-level dictionary for a fixed alphabet size and a fixed $D_{\max}$, respectively. As seen, the average indexing codelength of a multi-level dictionary results in a shorter indexing codelength.

2.4.2 The General Case

Given the training sequence $\{x_n, n = 1, \ldots, L\}$, suppose there are $a_d = \left|\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}\right| \leq |\mathcal{X}|^d$ patterns of depth $d \leq D_{\max}$ (a_1 patterns of depth one, a_2 patterns of depth two,

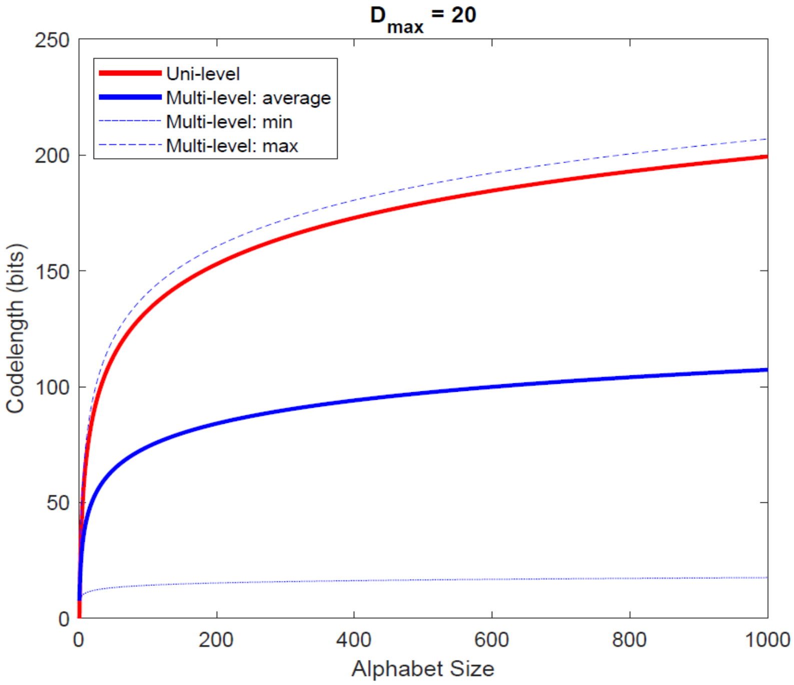


Figure 2.2: Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed $D_{\max} = 20$)

etc.). The following Theorem 1 shows that the average indexing codelength using a multi-level dictionary is always less than the indexing codelength of a uni-level dictionary.

Theorem 1. Assume there are embedded $a_d = \left|\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}\right| \leq |\mathcal{X}|^d$ patterns of depth $1 \leq d \leq D_{\max}$ in a training sequence of length $L \gg D_{\max}$. Let L^{uni} and L^{multi} be the indexing codelength of a uni-level dictionary and the average indexing codelength of a multi-level dictionary, respectively. Then,

(1) $L^{\mathrm{multi}} \leq L^{\mathrm{uni}}$; and

(2) $\log\left(1 + \frac{\left(\sqrt{a_{D_{\max}}} - \sqrt{a_1}\right)^2}{D_{\max}}\right) \leq L^{uni} - L^{\mathrm{multi}} \leq \log\left(1 + w + (1-w)\frac{a_{D_{\max}}}{a_1} - a_1^{w-1} a_{D_{\max}}^{1-w}\right)$,

where

$$w = \frac{\ln\left[\left(\frac{a_{D\max}}{a_{D_{\max}} - a_1}\right)\ln\frac{a_{\mathrm{Dmax}}}{a_1}\right]}{\ln\frac{a_{D_{\max}}}{a_1}}.$$

Proof. Since $L \gg D_{\max}$, clearly $0 < a_1 \leq a_2 \leq \cdots \leq a_{D_{\max}}$. Using a uni-level dictionary, the indexing codelength is

$$\begin{aligned} L^{\text{uni}} &= \log\left(\sum_{d=1}^{D_{\max}} a_d\right) \\ &= \log D_{\max} + \log A_{D_{\max}} \end{aligned}$$

where $A_{D_{\max}} \triangleq \left(a_1 + a_2 + \cdots + a_{D_{\max}}\right)/D_{\max}$ is the arithmetic mean of $a_1, a_2, \ldots, a_{D_{\max}}$. Using a multi-level dictionary the average indexing codelength is

$$\begin{aligned} L^{\text{multi}} &= \frac{1}{D_{\max}} \sum_{d=1}^{D_{\max}} \left(\log D_{\max} + \log a_d\right) \\ &= \log D_{\max} + \log G_{D_{\max}} \end{aligned}$$

where $G_{D_{\max}} \triangleq \left(\prod_{d=1}^{D_{\max}} a_d\right)^{1/D_{\max}}$ is the geometric mean of $a_1, a_2, \ldots, a_{D_{\max}}$. Hence, the comparison between L^{uni} and L^{multi} comes down to comparing the arithmetic mean and the geometric mean of $a_1, a_2, \ldots, a_{D_{\max}}$. Thus, $A_{D_{\max}} \geq G_{D_{\max}}$, which established the first part of the theorem. For the second part of the theorem, we use lower and upper bounds on $A_{D_{\max}} - G_{D_{\max}}$ derived in [74].

$$\begin{aligned} &\frac{\left(\sqrt{a_{D_{\max}}} - \sqrt{a_1}\right)^2}{D_{\max}} \leq A_{D_{\max}} - G_{D_{\max}} \leq \\ &\left[w a_1 + (1-w) a_{D_{\max}} - a_1^w a_{D_{\max}^{1-w}}\right], \end{aligned}$$

where $w = \frac{\ln[(a_{D_{\max}}/(a_{D_{\max}} - a_1)) \ln(a_{D_{\max}}/a_1)]}{\ln(a_{D_{\max}}/a_1)}$. Since $a_1 \leq G_{D_{\max}} \leq a_{D_{\max}}$ and $L^{\text{uni}} - L^{\text{multi}} = \log \frac{A_{D_{\max}}}{G_{D_{\max}}}$, the proof is complete. □

Theorem 1 shows that a multi-level dictionary gives a shorter average indexing codelength than a uni-level dictionary. $\log D_{\max} + \log a_d$ is the indexing codelength for patterns of depth d, where a_d is the total number of observed patterns of the depth d. In order to reduce the indexing codelength even further, the patterns of the same length in each set $\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$ can be ordered according to their relative frequency (empirical probability) in the training sequence. This allows Huffman or Shannon-Fano-Elias source coding [7] to be used to assign prefix codes to patterns in each set $\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$ separately. In this case, for any pattern $x_1^d \in \mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$, the indexing codelength becomes

$$L^{\text{multi}}\left(x_1^d\right) = \log D_{\max} + L_{\mathcal{D}}^{(d)}\left(x_1^d\right), \tag{2.1}$$

where $L_{\mathcal{D}}^{(d)}\left(x_1^d\right)$ is the codelength assigned to the pattern x_1^d based on its empirical probability using a Huffman or Shannon-Fano-Elias encoder. If such encoders are used, the codelength (1) is optimal ([7] Theorem 5.8.1). Since the whole purpose of creating a pattern dictionary is to learn the patterns in the training data, assigning the shorter codelength to the more frequent patterns and assigning longer codelength to the less frequent patterns in any pattern set $\mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$ will improve the efficiency of the coded representation.

Example 2. Suppose the alphabet is $\mathcal{X} = \{A, B, C, D\}$ and the training sequence is $x = ABACADABBACCADDABABACADAB$. Table 2.1 shows the dictionary with $D_{\max} = 3$ created by the patterns inside the training sequence, and the codelength assigned for each pattern using Huffman coding.

2.4.3 Pattern Dictionary for Detection (PDD)

Suppose we want to sequentially compress a test sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$ using a trained pattern dictionary $\mathcal{D}$ with maximum depth $D_{\max} < l$. The encoder parses the test sequence x_1^l into c phrases, $x_{v_1}^{v_2-1}, x_{v_2}^{v_3-1}, \ldots, x_{v_c}^l$ where v_i is the index of the start of the i th phrase, and each phrase $x_{v_i}^{v_{i+1}-1}$ is a pattern in the pattern

Table 2.1: Filling (training) the dictionary (of maximum depth $D_{\max} = 3$) with the patterns in the training sequence ABACADABBACCADDABABACADAB.

Depth 1			**Depth 2**			**Depth 3**		
x_1^d	$\Pr\left(x_1^d\right)$	$L_{\mathcal{D}}^{(1)}\left(x_1^d\right)$	x_1^d	$\Pr\left(x_1^d\right)$	$L_{\mathcal{D}}^{(2)}\left(x_1^d\right)$	x_1^d	$\Pr\left(x_1^d\right)$	$L_{\mathcal{D}}^{(3)}\left(x_1^d\right)$
A	0.44	1	AB	0.2083	2	ABA	0.1304	3
B	0.24	2	BA	0.1667	3	BAC	0.1304	3
C	0.16	3	AC	0.1250	3	CAD	0.1304	3
D	0.16	3	CA	0.1250	3	DAB	0.1304	3
			AD	0.1250	3	ACA	0.0870	4
			DA	0.1250	3	ADA	0.0870	4
			BB	0.0417	4	ABB	0.0435	4
			CC	0.0417	5	BBA	0.0435	4
			DD	0.0417	5	ACC	0.0435	4
						CCA	0.0435	4
						ADD	0.0435	4
						DDA	0.0435	5
						BAB	0.0435	5

dictionary $\mathcal{D}$. Let $\mathcal{S}_{\mathcal{D}}\left(x_1^l\right) = \left\{x_{v_1}^{v_2-1}, x_{v_2}^{v_3-1}, \ldots, x_{v_c}^l\right\}$ denote the set of the parsed phrases using pattern dictionary $\mathcal{D}$. The parsing process begins with setting $v_1 = 1$ and finding the largest $v_2 \leq D_{\max}$ and $v_2 \leq l$ such that $x_{v_1}^{v_2-1} \in \mathcal{D}$ but $x_{v_1}^{v_2} \notin \mathcal{D}$. This results in the first phrase $x_1^{v_2-1}$. Similarly, the same procedure is performed in order to find the largest $v_3 \leq D_{\max}$ and $v_3 \leq l$ such that $x_{v_2}^{v_3-1} \in \mathcal{D}$ but $x_{v_2}^{v_3} \notin \mathcal{D}$. This type of cross-parsing was first introduced in [53] in order to estimate an empirical relative entropy between two individual sequences that are independent realizations of two finite-order, finite-alphabet and stationary Markov processes. Here, we do not impose such an assumption on the sources generating the sequences. Algorithm 1 summarizes the procedure of the proposed pattern dictionary (PD) parser. After parsing the whole test sequence x_1^l into c phrases, $x_{v_1}^{v_2-1}, x_{v_2}^{v_3-1}, \ldots, x_{v_c}^l$, the codelength will be

$$L\left(x_1^l\right) = \sum_{i=1}^{c} L_{\mathcal{D}}\left(x_{v_i}^{v_{i+1}-1}\right) + c \log D_{\max}. \tag{2.2}$$

For detection purposes, on a test sequence x_1^l, either the number of parsed phrases or the codelength can be used as anomaly scores with respect to the trained pattern

Algorithm 1 Pattern Dictionary (PD) Parser

```
Require: Pattern Dictionary $\mathcal{D}$, Test Sequence $x_1^l$
Set $c = 1, v_c = 1, d = 1$
while $v_c + d - 1 < 1$ do
    if $x_{v_c}^{v_c+d-1} \in \mathcal{S}_{\mathcal{D}}^{(\mathrm{d})}$ then
        if $d + 1 \leq D_{\max}$ then
            $d = d + 1$
        else
            $v_{c+1} = v_c + d$
            $c = c + 1$
            $d = 1$
    else
        $v_{c+1} = v_c + d - 1$
        $c = c + 1$
        $d = 1$
return $x_{v_1}^{v_2} - 1, x_{v_2}^{v_3-1}, \dots, x_{v_c}^{l}$
```

dictionary $\mathcal{D}$. In other words, for any test sequence x_1^l and given a pattern dictionary, if the number of parsed phrases $\left|\mathcal{S}_{\mathcal{D}}\left(x_1^l\right)\right|$ or the codelength $L\left(x_1^l\right)$ in Equation (2) are greater than a certain threshold, then x_1^l is declared to be anomalous. While the proposed pattern dictionary technique can be used as a stand-alone anomaly detection technique, below we show how it can be used for atypicality detection [10], [11] as a training-based fixed source coder (data-dependent encoder).

2.5 Pattern Dictionary-Based Atypicality (PDA)

In [10], [11], an atypicality framework was introduced as a data discovery and anomaly detection framework that is based on a central definition: "a sequence (or subsequence) is atypical if it can be described (coded) with fewer bits in itself rather than using the (optimum) code for typical sequences". In this framework, detection is based on the comparison of a lossless descriptive codelength between an optimum encoder (if the typical model is known) or a training-based fixed source coder (if the typical model is unknown, but training data are available) and a universal source coder in order to detect atypical subsequences in the data [10], [11]. In this section, we apply our proposed pattern dictionary as a training-based fixed source coder (typical encoder) in an atypicality framework. We call it pattern dictionary-based aty picality (PDA) method.

The pattern dictionary-based source coder can be considered as a generalization of the Context Tree [75], [76], [77] based fixed source coder that was used in [10] for discrete data. The universal source coder (atypical encoder) used here is the Tree-Structured LempelZiv (LZ78) [7], [8]. The primary reason for choosing LZ78 as the universal encoder is that its sequential parsing procedure is similar to the proposed pattern dictionary described in Section 2.4, and it is (asymptotically) optimal [7], [8]. One might ask why do we even need to compare descriptive codelengths of a training-based (or optimum) encoder with a universal encoder for data discovery purposes when, as alluded to in the end of last section, a training-based fixed source coder can be a stand-alone anomaly detector. The necessity of such concurrent comparison is articulated in [10]. In fact, such a codelength comparison enables the atypicality framework to go beyond the detection of anomalies and outliers, extending to the detection of rare parts of data that might have a data structure of interest to the practitioner.

We give an example to provide further intuition for why anomaly detection can benefit from our framework that compares the outputs of a typical encoder and an atypical encoder. Consider an i.i.d. binary sequence of length L with $P(X = 1) = p$ in which there is embedded an anomalous subsequence of length $l \ll L$ with $P(X = 1) = \widehat{p} \neq p$ that we would like to detect. If $p = \frac{1}{2}$ and $\hat{p} = 1$, the typical encoder cannot catch the anomaly while the atypical encoder can. On the other hand, if $p = \frac{1}{3}$ and $\hat{p} = \frac{2}{3}$, the typical encoder identifies the anomaly while an atypical encoder fails to do so (since the entropy for $p = \frac{1}{3}$ and $\hat{p} = \frac{2}{3}$ is the same). Note that in both cases, our framework would catch the anomaly since it uses the difference between the descriptive codelengths of these two encoders.

Recall that in Section 2.4, we supposed that a test sequence x_1^l has been parsed using a trained pattern dictionary $\mathcal{D}$ with maximum depth $D_{\max} < l$. This parsing results in $\left|\mathcal{S}_{\mathcal{D}}\left(x_1^l\right)\right|$ parsed phrases. Using Equation (2.2), the typical codelength of the sequence x_1^l is given by

$$L_T\left(x_1^l\right) = \sum_{y \in \mathcal{S}_{\mathcal{D}}\left(x_1^l\right)} L_{\mathcal{D}}(y) + \left|\mathcal{S}_{\mathcal{D}}\left(x_1^l\right)\right| \log D_{\max}.$$

For the atypical encoder, the LZ78 algorithm results in a distinct parsing of the test sequence x_1^l. Let $\mathcal{S}_{LZ}\left(x_1^l\right)$ denote the set of parsed phrases in the LZ78 parsing of x_1^l. As such, the resulting atypical codelength is [7], [8]

$$L_A\left(x_1^l\right) = \left|\mathcal{S}_{LZ}\left(x_1^l\right)\right| \left[\log\left|\mathcal{S}_{LZ}\left(x_1^l\right)\right| + 1\right].$$

Since $L\left(x_1^l\right)$ using both LZ78 and the pattern dictionary depends on the number of parsed phrases, we investigate the possible range and properties of $\left|\mathcal{S}_{\mathcal{D}}\left(x_1^l\right)\right| - \left|\mathcal{S}_{LZ}\left(x_1^l\right)\right|$. While the LZ78 encoder is a well-known compression method which is asymptotically optimal [7], [8], its non-asymptotic behavior is not well understood. In the next section, we establish a novel non-asymptotic property of an LZ78 parser, and then compare it with the pattern dictionary parser.

2.5.1 Lempel-Ziv Parser

We start this section with a theorem that establishes the non-asymptotic lower and upper bounds on the number of distinct phrases in a sequence parsed by LZ78.

Theorem 2. The number of distinct phrases $c(l)$ resulting from LZ78 parsing of an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$ satisfies

$$\frac{1}{2}(\sqrt{8l+1} - 1) \leq c(l) \leq \frac{l \ln |\mathcal{X}|}{W\left(\frac{\beta}{\alpha}|\mathcal{X}|^{\frac{\alpha+1}{-\alpha}} \ln |\mathcal{X}|\right)}$$

where $\alpha = |\mathcal{X}| - 1, \beta = (|\mathcal{X}| - 1)^2 l - |\mathcal{X}|$, and W(.) is the Lambert W function [12].

Proof. First, we establish the upper bound. Note that the number of parsed distinct phrases $c(l)$ is maximized when all the phrases are as short as possible. Define $M \triangleq |\mathcal{X}|$ and let l_k be the sum of the lengths of all distinct strings of length less than or equal to k. Then,

$$l_k = \sum_{j=1}^{k} j M^j = \frac{1}{(M-1)^2}\left[\{(M-1)k - 1\}M^{k+1} + M\right]$$

Since $l = l_k$ occurs when all the phrases are of length $\leq k$,

$$c\left(l_k\right) \leq \sum_{j=1}^{k} M^j = \frac{M\left(M^k - 1\right)}{M-1} < \frac{M^{k+1}}{M-1} \leq \frac{l_k}{k - \frac{1}{M-1}}.$$

If $l_k \leq l < l_{k+1}$, we write $l = l_k + \triangle$ where

$$\begin{aligned} \Delta < l_{k+1} - l_k &= (Mk + M - 1 - k)\frac{M^{k+1}}{M-1} \\ &= (k+1)\frac{M^{k+1}}{M-1}. \end{aligned}$$

We conclude that the parsing ends up with $c\left(l_k\right)$ phrases of length $\leq k$ and $\frac{l-l_k}{k+1}$ phrases of length $k+1$. Therefore,

$$\begin{aligned} c(l) &\leq c\left(l_k\right) + \frac{l - l_k}{k+1} \leq \frac{l_k}{k - \frac{1}{M-1}} + \frac{\Delta}{k+1} \\ &\leq \frac{l_k + \Delta}{k - \frac{1}{M-1}} = \frac{l}{k - \frac{1}{M-1}}. \end{aligned} \tag{2.3}$$

We now bound the size of k for a given sequence of length l by setting $l = l_k$. Define $\alpha \triangleq M - 1$ and $\beta \triangleq (M-1)^2 l - M$. Then,

$$\begin{aligned}
&\frac{1}{(M-1)^2}\left[((M-1)k-1)M^{k+1}+M\right] = l \\
\Longleftrightarrow &((M-1)k-1)M^{k+1} = (M-1)^2 l - M \\
\Longleftrightarrow &(\alpha k - 1)M^{k+1} = \beta \\
\Longleftrightarrow &\widehat{k} M^{(\hat{k}+1)/\alpha+1} = \beta \\
\Longleftrightarrow &\hat{k}\frac{\ln M}{\alpha}\exp\left(\widehat{k}\frac{\ln M}{\alpha}\right) = \frac{\beta}{\alpha} M^{-1-1/\alpha}\ln M.
\end{aligned}$$

where $\hat{k} = \alpha k - 1$. The last equation can be solved using the Lambert W function [12]. Since all the involved numbers are real and for $M > 1$ and $l \geq 2$, we have $\frac{\beta}{\alpha}M^{-1-1/\alpha}\ln M \geq 0 > -\frac{1}{e}$, it follows that

$$\begin{aligned}
\widehat{k}\frac{\ln M}{\alpha} &= \mathrm{W}\left(\frac{\beta}{\alpha}M^{-1-1/\alpha}\ln M\right) \\
\Longleftrightarrow k &= \frac{\alpha \mathrm{W}\left(\frac{\beta}{a}M^{-1-1/\alpha}\ln M\right) + \ln M}{\alpha \ln M},
\end{aligned}$$

where W(.) is the Lambert W function. Using (2.3), we write

$$c(l) \leq \frac{l}{k - \frac{1}{\alpha}} = \frac{l \ln M}{W\left(\frac{\beta}{\alpha}M^{-1-1/\alpha}\ln M\right)}.$$

To prove the lower bound, note that the number of parsed distinct phrases $c(l)$ is minimized when the sequence of length l consists of only one symbol that repeats. Let $\widetilde{l_k}$ be the sum of the lengths of all such distinct strings of length less than or equal to k. Then,

$$\widetilde{l_k} = \sum_{j=1}^{k} j = \frac{k(k+1)}{2}.$$

Thus, given a sequence of length l by enforcing $l = \frac{k(k+1)}{2}$, we obtain the lower bound. □

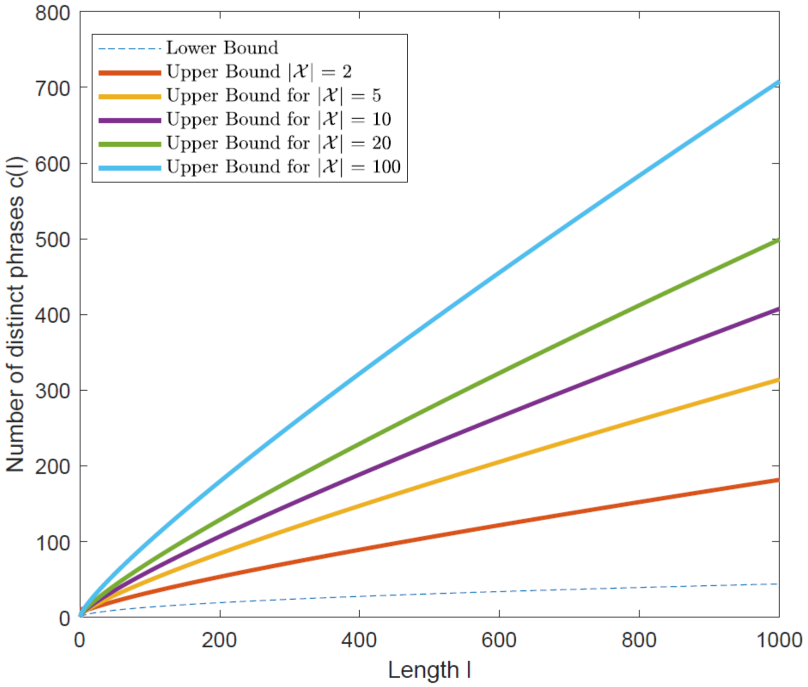


Figure 2.3: Plot of the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of an $|\mathcal{X}|$-ary sequence of length l.

Figure 2.3 illustrates the lower and upper bounds established in Theorem 2 against the sequence length for various alphabet sizes. Note that the lower bound on the number of distinct phrases is independent of the alphabet size.

While numerical experiments are not a substitute for the mathematical proof of Theorem 2 provided above, the reader may find it useful to understand the theorem in terms of a simple example. In Figures 4-6, we compare the theoretical bound with numerical results of simulation for binary i.i.d. sequences. In these experiments, for each value of $P(X = 1)$, a thousand binary sequences are generated; then, the number of distinct phrases resulting from LZ78 parsing of each sequence is calculated, and hence, the average, minimum, and maximum of these counts are found and represented by error bars.

Next, we verify the convergence of the non-asymptotic upper bound achieved in Theorem 2 to the asymptotic upper bound of the LZ78 parser. Using a lower bound

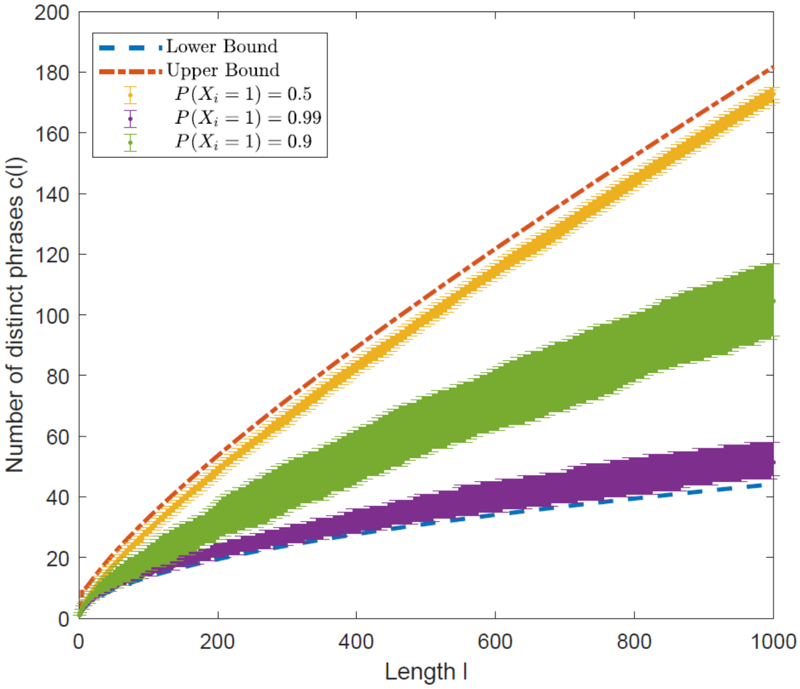


Figure 2.4: Simulation results compared to the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of binary sequences of length l generated by sources with three different source probabilities $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

on Lambert W function $\ln x - \ln(\ln x) \leq \mathrm{W}(x)$ [78], we write

$$\begin{aligned}
\mathrm{W}\left(\frac{\beta}{\alpha}\frac{\ln M}{M^{1+1/\alpha}}\right) &= \mathrm{W}\left(\left((M-1)l - \frac{M}{M-1}\right)\frac{\ln M}{M^{\frac{M}{M-1}}}\right) \\
&\approx \mathrm{W}\left(c_M l \ln M\right) \\
&\geq \ln \frac{c_M l \ln M}{\ln\left(c_M l \ln M\right)} \\
&= \ln \frac{c_M l}{\log\left(c_M l \ln M\right)}
\end{aligned}$$

where the logarithm is base $M = |\mathcal{X}|$ and $c_M = \frac{M-1}{M^{M/(M-1)}}$. Hence, we can further simplify the asymptotic upper bound of $c(l)$ as follows

$$\begin{aligned}
c(l) &\leq \frac{l \ln M}{W\left(\frac{\beta}{a}M^{-1-1/a}\ln M\right)} \\
&\leq \frac{l \ln M}{\ln \frac{c_M l}{\log(c_M l \ln M)}} \\
&= \frac{l}{\log \frac{c_M l}{\log(c_M l \ln M)}} \\
&= \frac{l}{\log l + \log c_M - \log\log\left(c_M l \ln M\right)} \\
&= \frac{l}{\left(1 - \frac{\log\log l + \widehat{c_M}}{\log l}\right)\log l},
\end{aligned}$$

where $\widehat{c_M} = \log c_M - \log\log\left(c_M \ln M\right)$. Therefore, as $l \to \infty$, we have $c(l) \leq \frac{1}{\log l}$. This is consistent with the binary case $M = 2$ proved in ([7] Lemma 13.5.3) or [8]. The following Lemma extends the result of ([7] Lemma 13.5.3) to $|\mathcal{X}|$-ary case.

Lemma 1. The number of distinct phrases $c(l)$ resulting from $LZ78$-parsing of an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$ satisfies

$$c(l) \leq \frac{l}{(1-\epsilon_l)\log l'}$$

where the logarithm is base $|\mathcal{X}|$ and $\epsilon_l = \min\left\{1, \frac{\log\log l - \log(|\mathcal{X}|-1) + \frac{3|\mathcal{X}|-2}{|\mathcal{X}|-1}}{\log l}\right\} \to 0$ as $l \to \infty$.

Proof. The proof is similar to the proof in ([7] Lemma 13.5.3) or ([79] Theorem 2). Let $M \triangleq |\mathcal{X}|$. In Theorem 2, we defined l_k as the sum of the lengths of all distinct strings of length less than or equal to k, and we showed that for any given l such that $l_k \leq l < l_{k+1}$, we have $c(l) \leq c\left(l_k\right) + \frac{l-l_k}{k+1} \leq \frac{l}{k-\frac{1}{M-1}}$. Next, we bound the size of k. As such, we have $l \geq l_k \geq M^k$ or, equivalently, $k \leq \log l$ where the logarithm is base M. Additionally,

$$\begin{aligned} l \leq l_{k+1} &= \left(k+1-\frac{1}{M-1}\right)\frac{M^{k+2}}{M-1} + \frac{M}{(M-1)^2} \\ &= \left(\frac{k}{M-1} + \frac{M-2}{(M-1)^2}\right)M^{k+2} + \frac{M}{(M-1)^2} \\ &\leq \frac{k+2}{M-1}M^{k+2} \leq \frac{\log l + 2}{M-1}M^{k+2} \end{aligned}$$

therefore, $k+2 \geq \log\frac{(M-1)l}{\log l+2}$. Equivalently, for $l \geq M^2$,

$$\begin{aligned} k - \frac{1}{M-1} &\geq \log l - \log(\log l + 2) + \log(M-1) - 2 - \frac{1}{M-1} \\ &= \left(1 - \frac{\log(\log l + 2) - \log(M-1) + \frac{2M-1}{M-1}}{\log l}\right)\log l \\ &\geq \left(1 - \frac{\log(2\log l) - \log(M-1) + \frac{2M-1}{M-1}}{\log l}\right)\log l \\ &= \left(1 - \frac{\log\log l - \log(M-1) + \frac{3M-2}{M-1}}{\log l}\right)\log l \\ &= (1-\epsilon_l)\log l, \end{aligned}$$

where $\epsilon_l = \min\left\{1, \frac{\log\log l - \log(M-1) + \frac{3M-2}{M-1}}{\log I}\right\}$. □

Next, we analyze the properties of the number of distinct phrases $c(l)$ resulting

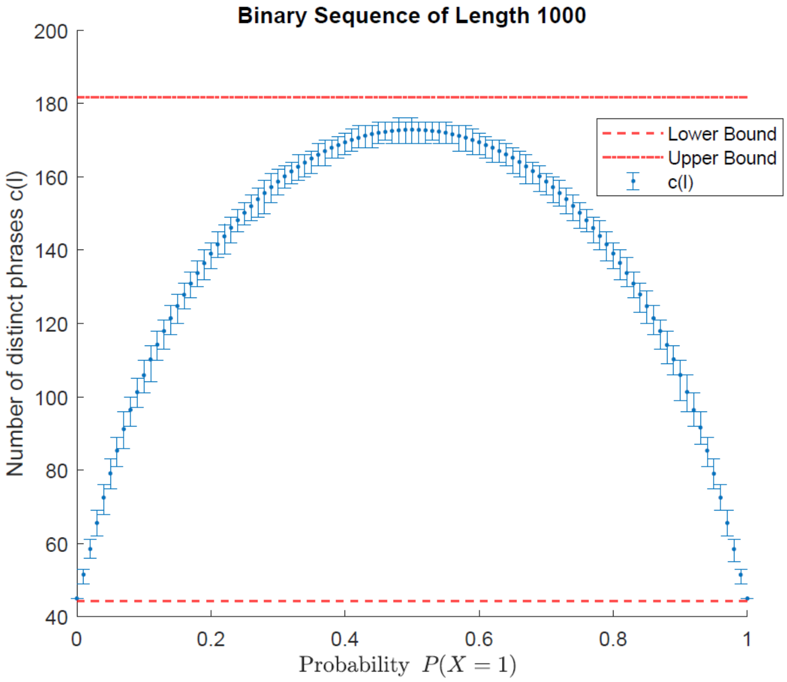


Figure 2.5: Similar to Figure 2.4 , the number of distinct phrases resulting from LZ78-parsing of binary sequences of fixed length $l = 1000$ varies over the source probability parameter $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

from LZ78-parsing of an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$ when l is fixed. The error bar representation in Figure 2.4 shows the variation of $c(l)$ when l is fixed. A possible explanation for such variations is that the statistical distribution of the pseudorandomly generated data are different from the theoretical distribution of the generating source. To elucidate this possibility, we enforce the exact matching of the source probability mass function and the empirical probability mass function of the generated data. Figure 2.5 represents the number of distinct phrases $c(l)$ resulting from LZ78-parsing of a binary sequence of fixed length where the characteristic of the generating source and the generated data matches. As seen, there is still some variation around the average value of $c(l)$. We can specify a distribution-dependent bound on $c(l)$ when both l and the distribution of the source are fixed.

In ([80] Theorem 1), for sequences generated from a memoryless source, $c(l)$ is assumed to be a random variable with the following mean and variance:

$$\mathrm{E}(c(l)) \sim \frac{hl}{\log l'}$$

$$\mathrm{Var}(c(l)) \sim \frac{(h_2 - h^2)\, l}{\log^2 l}, \tag{2.4}$$

where $h = -\sum_{a\in\mathcal{X}} p_a \log p_a$ is the entropy rate, and $h_2 = \sum_{a\in\mathcal{X}} p_a \log^2 p_a$ with p_a being the probability of symbol $a \in \mathcal{X}$. Note that the approximations (2.4) are asymptotic as $l \to \infty$. Below, we obtain a finite sample characterization of $c(l)$.

Consider an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$ with fixed length l generated from a source with the probability mass function $p(x)$. Here, the notations x_1^l and x^l are used interchangeably. Let $c(l, p)$ denote the number of distinct phrases resulting from LZ78-parsing of the sequence x_1^l of length l and the generating probability mass function is defined by $p(x)$. In order to find a distribution-dependent bound on the number of distinct phrases in LZ78-based parsing of x_1^l, we note that since the generating distribution is not necessarily uniform, all the strings x^n for $n < l \ll \infty$ do not necessarily appear as parsed phrases. For instance, consider the binary case with $P(X = 1) = 0.9$. Then, it is very unlikely to have a string with multiple consecutive zeros in any parsing of a realization of the finite sequence x^l. As such, using the Asymptotic Equipartition Properties (AEP) ([7] Chapter 3) or Non-asymptotic Equipartition Properties (NEP) [81], we define the typical set $\mathcal{A}_\epsilon^{(n)}$ with respect to $p(x)$ as the set of subsequences $x^n \in \mathcal{X}^n$ of x_1^l with the property

$$2^{-n(h+\epsilon)} \le p\left(x^n\right) \le 2^{-n(h-\epsilon)},$$

where h is the entropy. Then, we have

$$1 = \sum_{x^n \in \mathcal{X}^n} p\left(x^n\right) \geq \sum_{x^n \in \mathcal{A}_\varepsilon^{(n)}} p\left(x^n\right) \geq \left|\mathcal{A}_\epsilon^{(n)}\right| 2^{-n(h+\epsilon)},$$

therefore, $\left|\mathcal{A}_\varepsilon^{(n)}\right| \leq 2^{n(h+\epsilon)}$. Let l_k be the sum of the lengths of all the distinct strings x^n in the set $\left|\mathcal{A}_\epsilon^{(n)}\right|$ of length less than or equal to k. We write,

$$\begin{aligned} l_k &= \sum_{n=1}^{k} n \left|\mathcal{A}_\epsilon^{(n)}\right| \\ &\leq \sum_{n=1}^{k} n 2^{n(h+\epsilon)} \\ &= \frac{1}{(m-1)^2}\left[((m-1)k-1)m^{k+1}+m\right] \end{aligned}$$

where $m \triangleq 2^{h+\epsilon}$. Therefore, $l = \frac{1}{(m-1)^2}\left[((m-1)k-1)m^{k+1}+m\right]$ can be solved for k which leads into an upper bound for $c(l,p)$ as follows

$$\begin{aligned} k &= \frac{\alpha W\left(\frac{\beta}{a} m^{-1-1/\alpha} \ln m\right) + \ln m}{\alpha \ln m} \\ c(l,p) &\leq \sum_{n=1}^{k} \left|\mathcal{A}_\epsilon^{(n)}\right| = \frac{m\left(m^k-1\right)}{m-1} \\ &= \frac{2^{k(h+\epsilon)}-1}{1-2^{-h-\epsilon}}, \end{aligned}$$

where $\alpha = m-1$ and $\beta = (m-1)^2 l - m$. Therefore, the dependency of the $c(l,p)$ upper bound on the distribution is only through the entropy. Figure 6 depicts the upper bound on $c(l,p)$ for $\epsilon = 0.1$

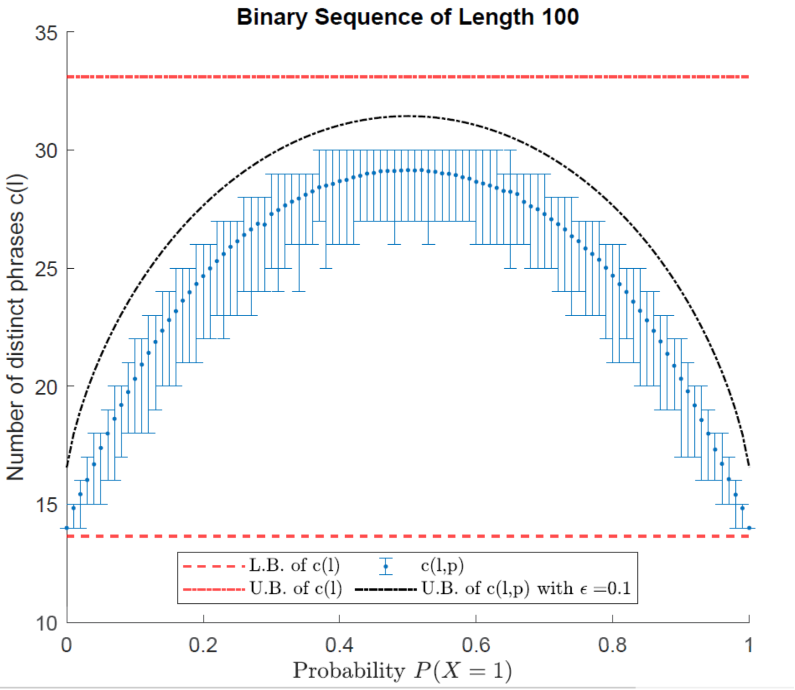


Figure 2.6: Simulation of the probability-dependent upper bound $c(l, p)$ for binary sequences of fixed length $l = 100$ with various probability parameters $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

2.5.2 Pattern Dictionary Parser versus $LZ78$ Parser

Given an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n = 1, \ldots, l\}$, let $c_T(l)$ be the number of parsed phrases of x_1^l when the ty pical encoder (pattern dictionary with $D_{\max}$) is used, and $c_A(l)$ be the number of parsed phrases of x_1^l when the atypical encoder (LZ78) is used. Clearly, $\frac{l}{D_{\max}} \leq c_T(l) \leq l$ where the lower bound is achieved when $\mathcal{S}_{\mathcal{D}}\left(x_1^l\right) = \left\{x_{v_1}^{v_2-1}, x_{v_2}^{v_3} - 1, \ldots, x_{v_c}^{l}\right\}$, and each $x_{v_i}^{v_i-1} \in \mathcal{S}_{\mathcal{D}}^{(\mathrm{D_{max}})}$, namely $x_{v_i}^{v_i-1}$ is of length $D_{\max}$ and exists in the dictionary. The upper bound is achieved when $\mathcal{S}_{\mathcal{D}}\left(x_1^l\right) = \{x_1, x_2, \ldots, x_l\}$ where each $x_n \in \mathcal{S}_{\mathcal{D}}^{(1)}$. Using the result of Theorem 2 and a lower bound on the Lambert W function, $\ln x - \ln(\ln x) \leq \mathrm{W}(x)$ [78], we have

$$\begin{aligned} \frac{l}{D_{\max}}\left(1 - \frac{D_{\max}}{\log \frac{l}{\log(l \ln |\mathcal{X}|)}}\right) &\leq c_T(l) - c_A(l) \\ &\leq l\left(1 - \frac{\sqrt{8l+1} - 1}{2l}\right) \end{aligned} \tag{2.5}$$

The above bounds have asymptotic and non-asymptotic implications. The asymptotic analysis of the bounds in (2.5) suggests that as $l \to \infty$, for a dictionary with fixed $D_{\max}$, we have $\frac{l}{D_{\max}} \leq c_T(l) - c_A(l) \leq l$. This inequality implies the asymptotic dominance of the parser using a typical encoder. This is to be expected due to the asymptotic optimality of LZ78. However, the above inequality also implies a more interesting result: if $D_{\max} > \log \frac{l}{\log(l \ln |X|)}$ as $l \to \infty$, then $c_T(l)$ can be smaller than $c_A(l)$. The non-asymptotic behavior of the bounds in (2.5) is more relevant to the anomaly detection problem. These bounds suggest that for a fixed l and $|\mathcal{X}|$, increasing $D_{\max}$ has a vanishing effect on the possible range of the anomaly score. Additionally, the achieved bounds on $c_T(l) - c_A(l)$ provide the range of values of the anomaly score. This facilitates the search for a data-dependent threshold for anomaly detection, as the search can be restricted to this range.

2.5.3 Atypicality Criterion for Detection of Anomalous Subsequences

Consider the problem of finding the atypical (anomalous) subsequences of a long sequence with respect to a trained pattern dictionary $\mathcal{D}$. Suppose we are looking for an infrequent anomalous subsequence $x_n^{n+l-1} = \{x_n, n = n, \ldots, n+l-1\}$ embedded in a test sequence $\{x_n, n = 1, \ldots, L\}$ from the finite alphabet $\mathcal{X}$. Using Equation (2.2), the typical codelength of the subsequence x_n^{n+l-1} is

$$L_T\left(x_n^{n+l-1}\right) = \sum_{y \in \mathcal{S}_{\mathcal{D}}\left(x_n^{n+l-1}\right)} L_{\mathcal{D}}(y) + \left|\mathcal{S}_{\mathcal{D}}\left(x_n^{n+l-1}\right)\right| \log D_{\max},$$

while using LZ78, the atypical codelength of the subsequence x_n^{n+1-1} is

$$\begin{aligned} L_A\left(x_n^{n+l-1}\right) = & \left|\mathcal{S}_{LZ}\left(x_n^{n+l-1}\right)\right| \left[\log\left|\mathcal{S}_{LZ}\left(x_n^{n+l-1}\right)\right| + 1\right] \\ & + \log^*(l) + \tau, \end{aligned}$$

where $\log^*(l) + \tau$ is an additive penalty for not knowing in advance the start and end points of the anomalous sequence [10], [11], and $\log^*(l) = \log l + \log\log l + \ldots$ where the sum continues as long as the argument to the outer log is positive. Let $L'_A = L_A - \tau$. We propose the following atypicality criterion for detection of an anomalous subsequence:

$$\Delta L(n) = \max_l \left\{L_T\left(x_n^{n+l-1}\right) - L'_A\left(x_n^{n+l-1}\right)\right\} > \tau, \tag{2.6}$$

where τ can be treated as an anomaly detection threshold. In practice, τ can be set to ensure a false positive constraint, e.g., using bootstrap estimation of the quantiles in the training data.

2.6 Experiments

In this section, we illustrate the proposed pattern dictionary anomaly detection on a synthetic time series, known as Mackey-Glass [82], as well as on a real-world time series of physiological signals. In both experiments, first, the real-valued samples are discretized using a uniform quantizer [83], and then, anomaly detection methods are applied.

2.6.1 Anomaly Detection in Mackey-Glass Time Series

In this section, we illustrate the proposed anomaly detection method for the case of a chaotic Mackey-Glass (MG) time series that has an anomalous segment grafted into the middle of the sequence. MG time series are generated from a nonlinear time delay differential equation. The MG model was originally introduced to represent the appearance of complex dynamic in physiological control systems [82]. The nonlinear differential equation is of the form $\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-\delta)}{1+x^{10}(t-\delta)}, t \geq 0$, where a, b and δ are constants. For the training data, we generated 3000 samples of the MG time series with $a = 0.2$, $b = 0.1$, and $\delta = 17$. For the test data, we normalized and embedded 500 samples of the MG time series with $a = 0.4, b = 0.2$, and $\delta = 17$ inside 1000 samples of a MG time series generated from the same source as the training data, resulting in a test sequence of length 1500. Figure 2.7 shows a realization of the training data and the test data.

The anomaly detection performance of our proposed pattern dictionary is evaluated. To illustrate the effect of the model parameter, i.e., the maximum depth $D_{\max}$, on the detection and compression performance of the pattern dictionary, we run two experiments. First, we use a 30-fold cross-validation on the training data (resulting in 30 sequences of length 100) and calculate the number of distinct parsed phrases against $D_{\max}$. Second, we train a pattern dictionary with various $D_{\max}$ using the training data and then evaluate the sensitivity of detector of the anomalous subse-

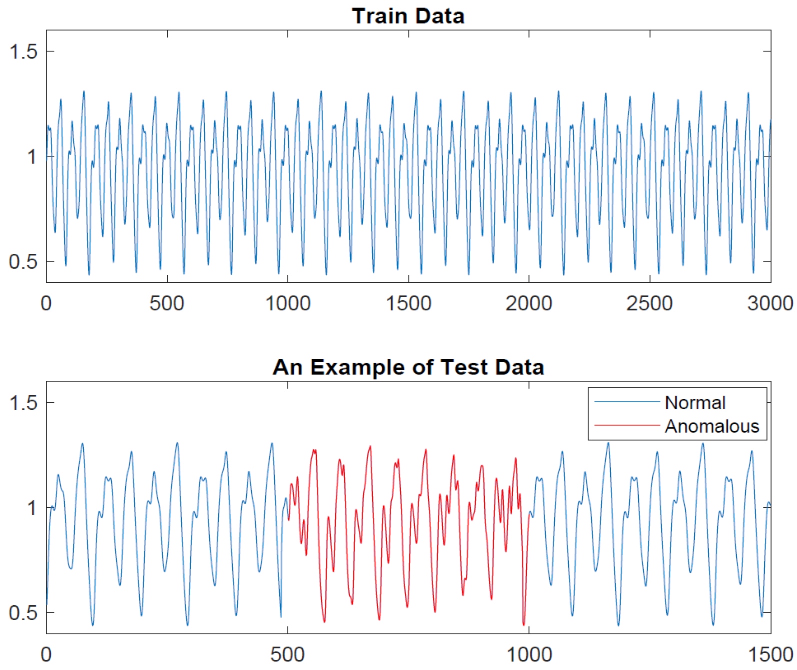


Figure 2.7: Mackey-Glass time series: the training data (top) and an example of the test data (bottom) in which samples in $[501, 1000]$ are anomalous (shown in red).

quences in the test data using Equation (2.6) with $\tau = 0$. In this experiment, the detection sensitivity (true positive rate) is defined as the ratio of number of samples correctly identified as anomalous over the total number of anomalous samples. Figure 2.8 illustrates the result of both experiments. As seen, after some point, increasing $D_{\max}$ has diminishing effect on both detection sensitivity and the number of distinct parsed phrases. Note that this behavior is to be expected as it was suggested by the bounds in (2.5).

Next, we compare anomaly detection performance of our proposed pattern dictionary methods, PDD and PDA, with the nearest neighbors-based similarity (NNS) technique [13], the compression-based dissimilarity measure (CDM) method [18], [19], [20], Ziv-Merhav method (ZM) [53], and the threshold Sequence Time-Delay Embedding (t-STIDE) technique [14], [15], [16], [17]. In this experiment, a window of length 100 is slid over the test data and each method measures the anomaly score (as described below) of the current subsequence with respect to the training data.

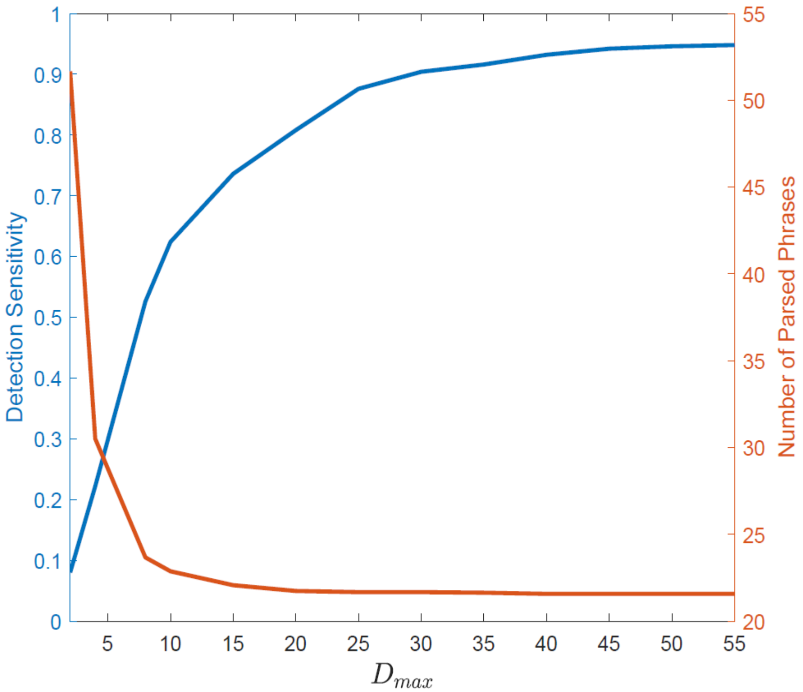


Figure 2.8: The effect of maximum dictionary depth $D_{\max}$ on parsing and detection sensitivity (true positive rate) of the Mackey-Glass time series presented in Figure 2.7.

The anomaly is detected when the score exceeds a threshold, determined to ensure a specified false positive rate. In the following, we compute AUC (area under the curve) of the ROC (receiver operating characteristic) and Precision-Recall curves as performance measures. In the following, we provide details of the implementation.

Pattern Dictionary for Detection (PDD): First, the training data are used to create a pattern dictionary with $D_{\max} = 40$, as described in Section 2.4. Then, for each subsequence x^{100} (the sliding window of length 100) of the test data, the anomaly score is computed as the codelength $L\left(x^{100}\right)$ of Equation (2.2) described in Section 2.4.3.

Pattern Dictionary Based Atypicality (PDA): Similar to PDD, first the training data are used to create a pattern dictionary with $D_{\max} = 40$, as described in Section 2.4. Then, for each subsequence x^{100} of the test data, the anomaly score is the atypicality measure described in Section 2.5, i.e., $L_T\left(x^{100}\right) - L_A\left(x^{100}\right)$, the difference between the compression codelength of the test subsequence using typical encoder (pattern dictionary) and atypical encoder (LZ78).

Ziv-Merhav Method (ZM) [53]: In this method, a cross-parsing procedure is used in which for each subsequence x^{100} of the test data, the anomaly score is computed as the number of the distinct phrases of x^{100} with respect to the training data.

Nearest Neighbors-Based Similarity (NNS) [13]: In this method, a list $\mathcal{S}$ of all the subsequence of length 100 (the length of the sliding window) of the training data is created. Then, for each subsequence x^{100} of the test data, the distance between x^{100} and all the subsequences in the list $\mathcal{S}$ is calculated. Finally, the anomaly score of x^{100} is its distance to the nearest neighbor in the list $\mathcal{S}$.

Compression-Based Dissimilarity Measure (CDM) [18], [19], [20]: In this method, given the training data x_{train} , for each subsequence x^{100} of the test data the anomaly score is

$$\operatorname{CDM}\left(x_{\text{train}}, x^{100}\right) = \frac{\mathcal{L}\left(\mathcal{C}\left(x_{\text{train}}, x^{100}\right)\right)}{\mathcal{L}\left(x_{\text{train}}\right) + \mathcal{L}\left(x^{100}\right)}$$

where $\mathcal{C}(y, x)$ represents concatenation of sequences y and z, and $\mathcal{L}(x)$ is the size of the compressed version of the sequence x using any standard compression algorithm. The CDM anomaly score is close to 1 if the two sequence are not related, and smaller than one if the sequences are related.

Threshold Sequence Time-Delay Embedding (t-STIDE) [14], [15], [16], [17]: In this method, given $l < 100$, for each sub-subsequence x^l of the subsequence x^{100} of the test data, the likelihood score of x^l is the normalized frequency of its occurrence in the training data, and the anomaly score of x^{100} is one minus the average likelihood score of all its sub-subsequences of length l. In this experiment, various values of l are tested and the best performance is reported.

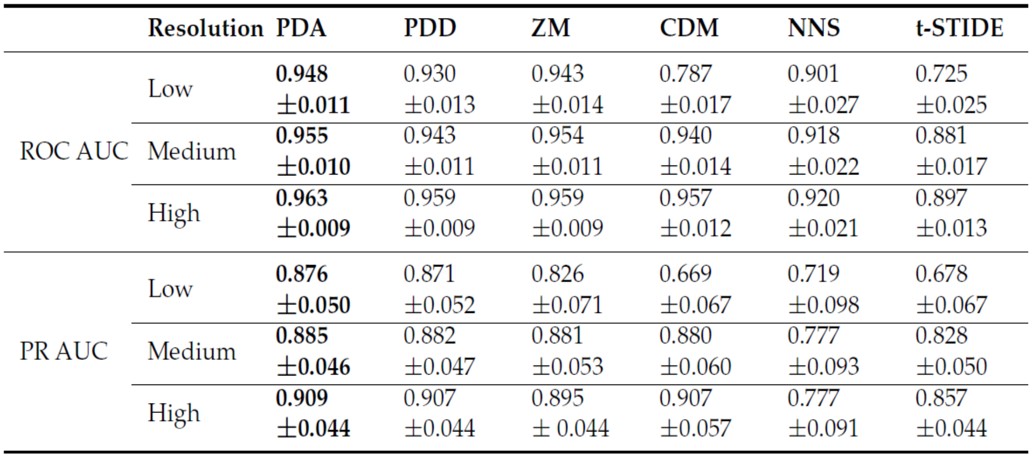
We compare the detection performance of the aforementioned methods by generating 200 test data sequences with different anomaly segments (the anomalous MG segments have different initializations in each test dataset). The detection results of comparisons are reported in Table 2.2. As seen, our proposed PDD and PDA methods outperform the rest, with ZM and CDM coming in third place. The effect of alphabet size of the quantized data (the resolution parameter of the uniform quantizer [83]) on anomaly detection performance is summarized in Table 2.3. Table 2.3 shows that our proposed PDD and PDA methods outperform in all three cases of data resolution.

Table 2.2: Comparison of anomaly detection methods ($\mu \pm \sigma$ representation is used where μ is the mean and σ is the standard deviation). The proposed PDA method attains overall best performance (bold entries of table).

	ROC AUC	PR AUC
PDA	$\mathbf{0.963 \pm 0.009}$	$\mathbf{0.909 \pm 0.044}$
PDD	0.959 ± 0.009	0.907 ± 0.044
ZM	0.959 ± 0.009	0.895 ± 0.049
CDM	0.957 ± 0.012	0.907 ± 0.057
NNS	0.920 ± 0.021	0.777 ± 0.091
t-STIDE	0.897 ± 0.013	0.857 ± 0.044

Since the parsing procedure of our proposed PD-based methods and the ZM method [53] are similar, it is of interest to compare the running time of these two methods. While the cross-parsing procedure of the ZM method was introduced as an on the fly process [53], we can also consider another implementation similar to our proposed PD by creating a codebook of all the subsequences of the training data prior to the parsing procedure. As such, in order to compare the running time of the dictionary/codebook creation and parsing procedure of our PD-based methods with the aforementioned two implementations of the ZM method, we use the same MG training data of length 3000 , one test dataset of length 1500 while a sliding window of length 100 is slid over it for anomaly score calculation, and the PD-based method with $D_{\max} = 40$. Note that since a sliding window of length 100 over the test data is considered, for the codebook-based implementation of ZM, all the subsequences of the training data up to length 100 are extracted which make its codebook creation process significantly faster. Table 2.4 summarizes the running time comparison. As it can be seen, our PD-based method is faster in both dictionary/codebook creation and parsing process.

Table 2.3: Comparison of anomaly detection methods for different cases of data resolutions: high resolution corresponds to an alphabet size of 90 , medium resolution corresponds to an alphabet size of 45 , and low resolution corresponds to an alphabet size of 10 . In this table, $\mu \pm \sigma$ representation is used where μ is the mean and σ is the standard deviation. The proposed PDA method achieves overall best performance (bold entries of table).

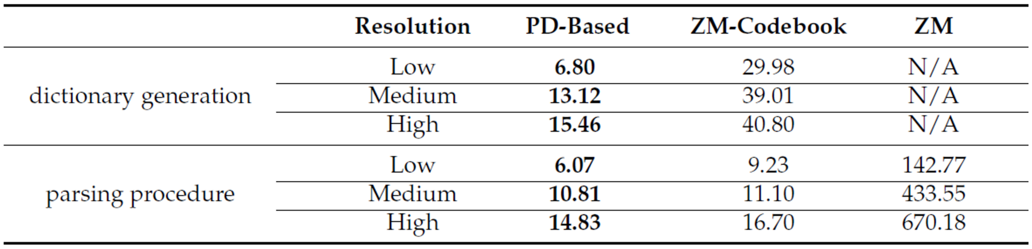


	Resolution	PDA	PDD	ZM	CDM	NNS	t-STIDE
ROC AUC	Low	**0.948** **±0.011**	0.930 ±0.013	0.943 ±0.014	0.787 ±0.017	0.901 ±0.027	0.725 ±0.025
	Medium	**0.955** **±0.010**	0.943 ±0.011	0.954 ±0.011	0.940 ±0.014	0.918 ±0.022	0.881 ±0.017
	High	**0.963** **±0.009**	0.959 ±0.009	0.959 ±0.009	0.957 ±0.012	0.920 ±0.021	0.897 ±0.013
PR AUC	Low	**0.876** **±0.050**	0.871 ±0.052	0.826 ±0.071	0.669 ±0.067	0.719 ±0.098	0.678 ±0.067
	Medium	**0.885** **±0.046**	0.882 ±0.047	0.881 ±0.053	0.880 ±0.060	0.777 ±0.093	0.828 ±0.050
	High	**0.909** **±0.044**	0.907 ±0.044	0.895 ± 0.044	0.907 ±0.057	0.777 ±0.091	0.857 ±0.044

2.6.2 Infection Detection Using Physiological Signals

Finally, we apply the proposed pattern dictionary method to detect unusual patterns in physiological signals of two human subjects after exposure to a pathogen while only one of these subjects became symptomatically ill. The time series data were collected in a human viral challenge study that was performed in 2018 at the University of Virginia under a DARPA grant. Consented volunteers were recruited into this study following an IRB-approved protocol and the data was processed and analyzed at Duke University and the University of Michigan. The challenge study design and data collection protocols are described in [84]. Volunteers' skin temper-

Table 2.4: Table 4. Comparison of running time (in second) of PD-based method and two implementations of the ZM method for different cases of data resolutions: high resolution corresponds to an alphabet size of 90 , medium resolution corresponds to an alphabet size of 45 , and low resolution corresponds to an alphabet size of 10. This experiment is performed on a Hansung laptop with 2.60GHzCPU, 500 GB of SSD, and 16 GB of RAM using MATLAB R2021a. The proposed PD-based method has fastest run time overall (bold entries in table).



	Resolution	PD-Based	ZM-Codebook	ZM
dictionary generation	Low	**6.80**	29.98	N/A
	Medium	**13.12**	39.01	N/A
	High	**15.46**	40.80	N/A
parsing procedure	Low	**6.07**	9.23	142.77
	Medium	**10.81**	11.10	433.55
	High	**14.83**	16.70	670.18

ature and heart rate were recorded by a wearable device (Empatica E4) over three consecutive days before and five consecutive days after exposure to a strain of human Rhinovirus (RV) pathogen. During this period, the wearable time series were continuously recorded while biospecimens (viral load) were collected daily. The infection status can be clinically detected by biospecimen samples, but in practice, the collection process of these types of biosamples can be invasive and costly. As such, here, we apply the proposed anomaly detection framework to the measured two-dimensional heart rate and temperature time series to detect unusual patterns after exposure with respect to the normal (healthy) baseline patterns. In the preprocessing phase, we followed the wearable data preprocessing procedure described in [85]. Specifically, we first downsample the time series to one sample per minute by averaging. Then, we apply an outlier detection procedure to remove technical noise, e.g., sensor contact loss. After preprocessing, the two-dimensional space of temperature and heart rate time series is discretized using a two-dimensional uniform quantizer [83] with step size of 5 for heart rate and 0.5 for temperature, resulting in one-dimensional discrete sequence data. The first three days of data are used as the training data, and the PDA methods with maximum depth $D_{\max} = 30$ are used to learn the patterns in the training data. In order to detect anomalous patterns of the test data (the last five

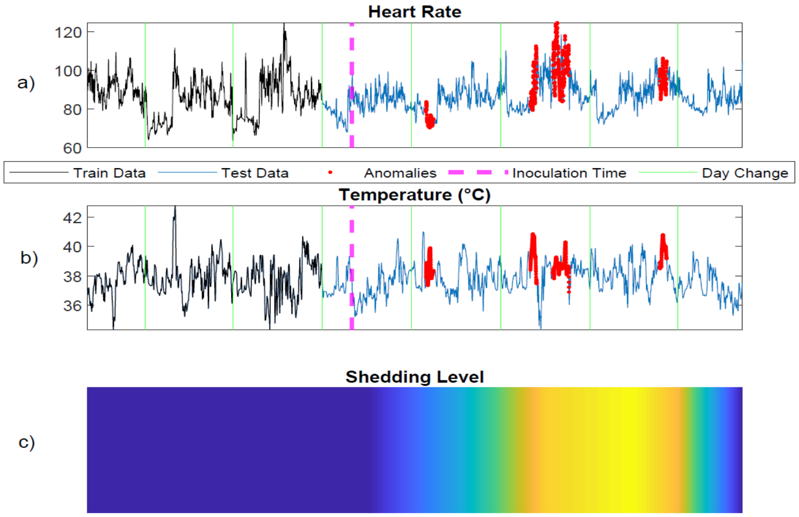


Figure 2.9: Anomaly detection using the proposed PDA method for a subject based on heart rate and temperature data collected from a wearable wrist sensor. Anomalies are shown in red in (a,b). (c) shows the subject's infection level.

days), we used the result of Section 2.5.3 and the atypicality criterion of Equation (2.6), which requires choosing the threshold τ. While this threshold can be chosen freely, we selected it using cross-validation on the training data. Leave-one-out cross-validation over the training data generates an empirical null distribution of the PDA anomaly score function $L_T - L_A$. The threshold τ was chosen as the upper 99% quantile of this distribution. Figure 2.9 illustrates the result of anomaly detection on one subject who became infected as measured by viral shedding as shown in Figure 2.9.c. All the anomalous patterns occur when the subject was shedding the virus. Figure 2.10 also depicts the result of anomaly detection on one subject who had a mild infection with a low level of viral shedding, as shown in Figure 2.10.c. Note that in this case, no anomalous patterns were detected.

2.7 Conclusions

In this paper, we have developed a universal nonparametric model-free anomaly detection method for time series and sequence data using a pattern dictionary. We

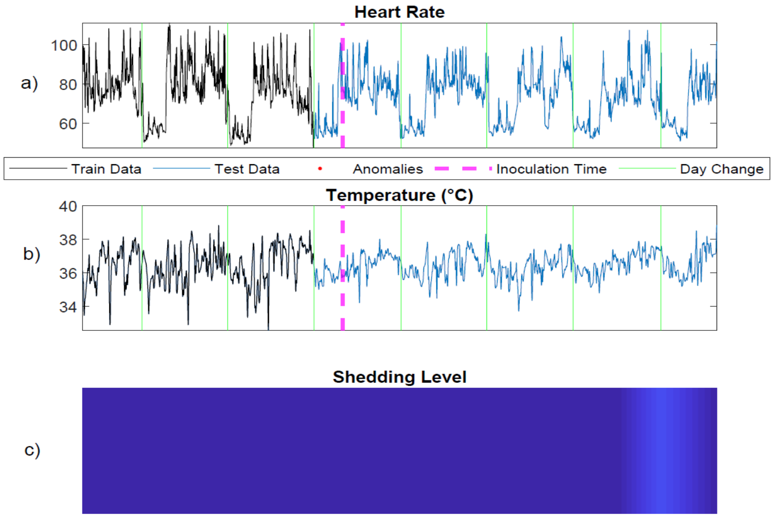


Figure 2.10: Anomaly detection using the proposed PDA method for a subject who had a mild infection with low level of viral shedding based on heart rate and temperature data collected from a wearable wrist sensor. Note that no anomaly has been detected: (a) heart rate, (b) temperature, and (c) infection level.

proved that using a multi-level dictionary that separates the patterns by their depth results in a shorter average indexing codelength in comparison to a uni-level dictionary that uses a uniform indexing approach. We illustrated that the proposed pattern dictionary method can be used as a stand-alone anomaly detector, or integrated with Tree-Structured LempelZiv (LZ78) and incorporated into an atypicality framework. We developed novel nonasymptotic lower and upper bounds of the LZ78 parser and demonstrated that the nonasymptotic upper bound on the number of distinct phrases resulting from LZ78-parsing of an $|\mathcal{X}|$-ary sequence can be explicitly derived in terms of the Lambert W function, an important theoretical result that is not trivial. We showed that the achieved non-asymptotic bounds on LZ78 and pattern dictionary determine the range of the anomaly score and the anomaly detection threshold. We also presented an empirical study in which the pattern dictionary approach is used to detect anomalies in physiological time series. In the future work, we will investigate the generalization of the context tree weighting methods to the general discrete case, using the pattern dictionary since the pattern dictionary handles sparsity well and is computationally less expensive when the alphabet size is large.

CHAPTER III

Pattern Dictionary Method with Clustering

3.1 Introduction

The previous chapter described an anomaly detection procedure that combined typicality and atypicality scores by subtracting atypicality from typicality, and thresholding the result (Equation 2.6). This can be integrated as scalarizing the 2-dimensional vector scores (typicality and atypicality). In this chapter, we develop the pattern dictionary method without scalarization, looking instead at the new 2-dimensional score vector for deviations from a normal 2D distribution. We first describe a clustering approach for the case where several samples from anomalous distribution are available. Then we apply a level-set type of anomaly detection technique, Leave-one-out kNN graph (L1O-kNNG) to detect single sample anomalies from the 2-dimensional score vector. The 2-dimensional score (typicality, atypicality) can achieve better anomaly detection performance than the scalarized score. We apply the method to detecting anomalous sequences of the Mackey glass signal explained in the next section. In Section 3.2, we will demonstrate that using the (typicality, atypicality) score instead of the scalarized score used in Chapter 2 improves the performance of anomaly detection. Here, we apply k-means, agglomerative, and normalized spectral clustering, in addition to a consensus-based DRPT (Dual Rooted Prim Tree) method [86] for the multiple sample anomaly detector. These are based on centroid-based [87], hierarchical [88], [89], and spectral clustering [90] methods, respectively.

3.2 Multiple Sample Anomaly Detection

In this section, we demonstrate how clustering can be applied to the 2D plane of typicality and atypicality to accomplish anomaly detection for multiple samples. For this multiple-sample anomaly detection problem, several clustering methods can be applied including agglomerative, k-means, spectral clustering, and DRPT methods. In Section 3.2.1, we describe the different experiments we performed. In Section 3.2.2, we show that the 2-dimensional typicality and atypicality plane results in improved performance. Furthermore, in Section 3.2.3, we show that using high-dimensional typicality and atypicality results in more stable performance compared to using a 2-dimensional score.

3.2.1 Mackey Glass Experiments

Several experiments are conducted for the case of a chaotic Mackey-Glass time series that has an anomalous segment embedded into the middle of the sequence. We generated the same MG samples with $a = 0.2$, $b = 0.1$, and $\delta = 17$ for the training data in 2.6. For the test data, we embedded the same anomalous sample with $a = 0.4$, $b = 0.2$, and $\delta = 17$ in Section 2.6 into the middle of the training data. We generated and embedded different types of anomalous data into the middle of the training data so, the experiments are conducted with four cases to prove the robustness of the proposed method. Refer to Figure 3.2 and 3.3.

3.2.2 Results of Clustering 2-dimensional Scores

By using a 2-dimensional score (typicality, atypicality), we will demonstrate that the clustering methods are applied as shown through accuracy and F1-score in Figure 3.4 - 3.7. The accuracy and the F1-score are calculated by $\frac{TP+TN}{TP+FP+TN+FN}$ and

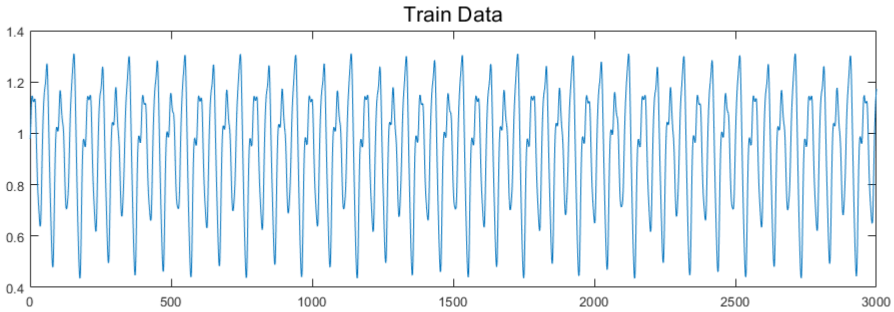
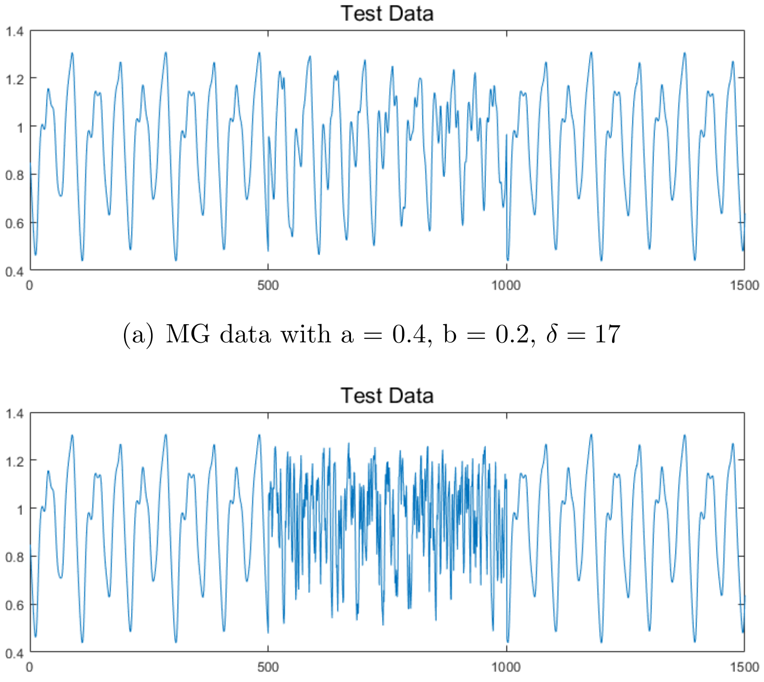


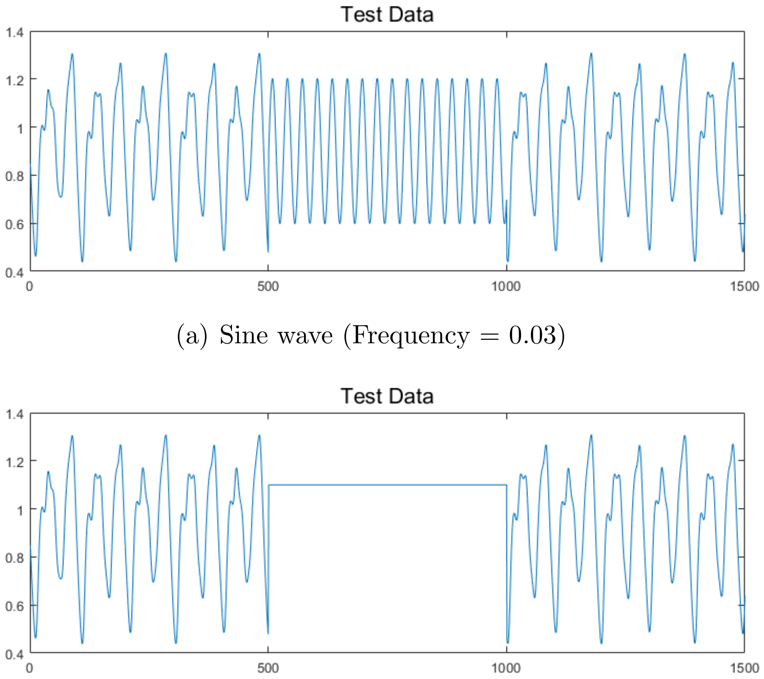
Figure 3.1: Train data is generated by Mackey Glass equation



(a) MG data with a = 0.4, b = 0.2, $\delta = 17$

(b) MG data with a = 1.2, b = 0.6, $\delta = 17$

Figure 3.2: Complex anomaly cases (Anomalies are embedded from 501 to 1000)



(a) Sine wave (Frequency = 0.03)

(b) Flat signal (y = 1.1)

Figure 3.3: Simple anomaly cases (Anomalies are embedded from 501 to 1000)

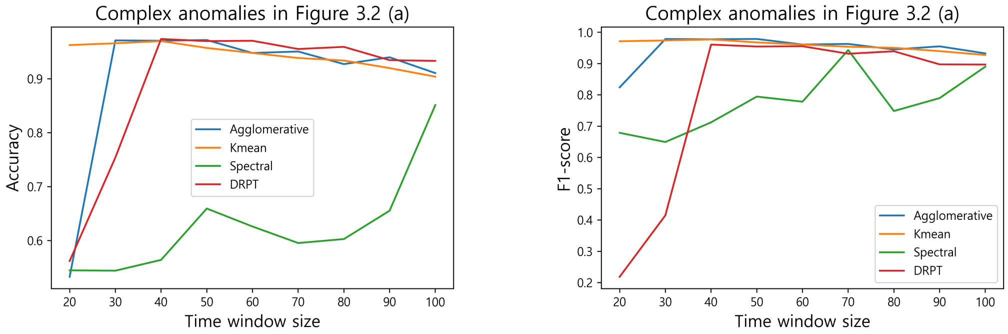


Figure 3.4: Accuracy and F1 score of the case in Figure 3.2 (a) based on time window size

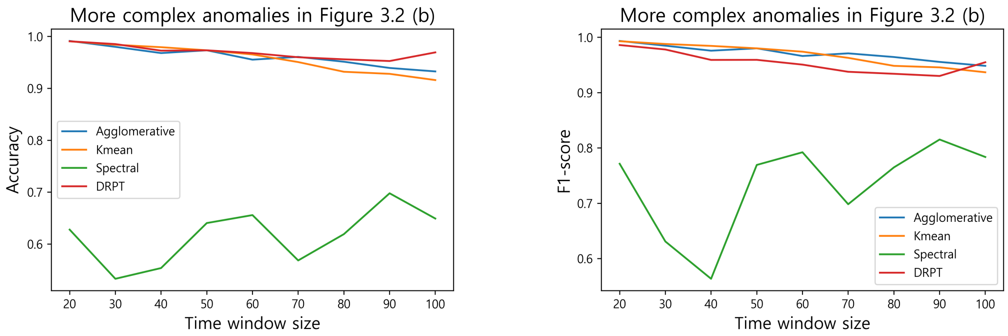


Figure 3.5: Accuracy and F1 score of the case in Figure 3.2 (b) based on time window size

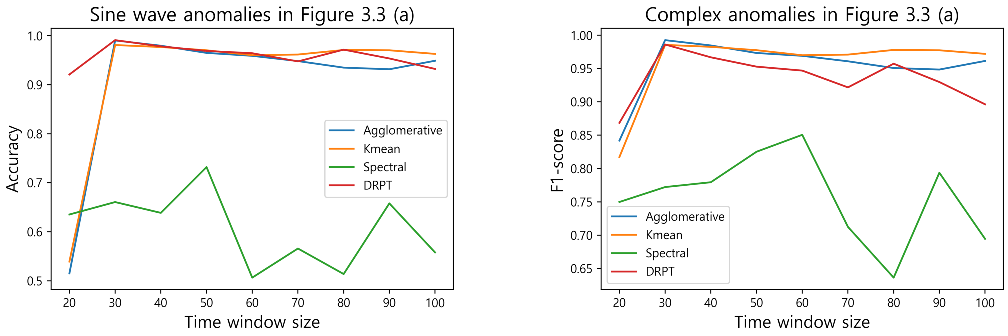


Figure 3.6: Accuracy and F1 score of the case in Figure 3.3 (a) based on time window size

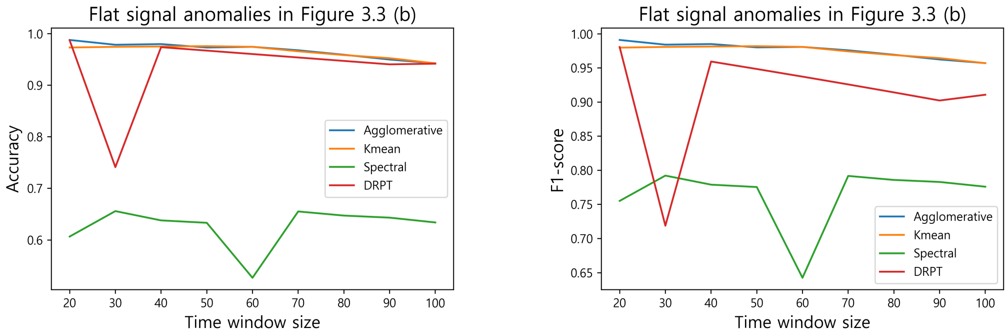


Figure 3.7: Accuracy and F1 score of the case in Figure 3.3 (b) based on time window size

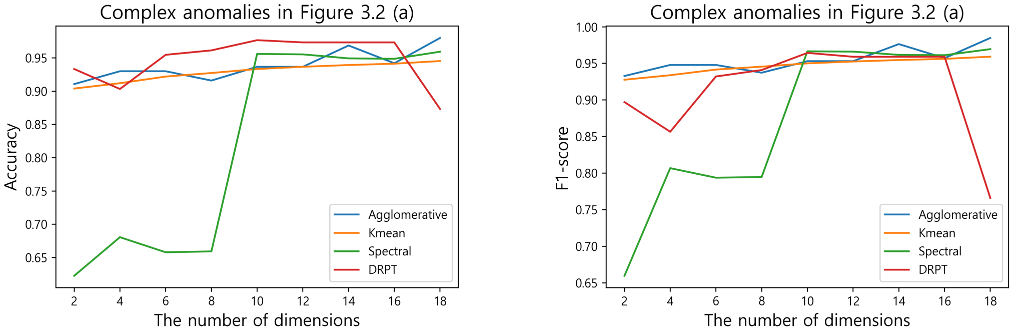


Figure 3.8: Accuracy and F1 score of the case in Figure 3.2 (a) based on the number of dimensions

$\frac{2 \times precision \times recall}{precision + recall}$, respectively. In the cases, the clustering methods perform well with a certain time window size except for the spectral clustering method. There are also specific time window sizes where clustering methods do not perform effectively, such as the window size of 20 in Figure 3.6 and the window size of 30 in Figure 3.7. Thus, we propose a method that considers multiple typicality and atypicality at the same time in the next section.

3.2.3 Results of Clustering High-dimensional Scores

In Figure 3.8 - 3.11, the x-axis represents the number of dimensions. These figures start with two dimensions (One typicality $C_T(100)$ and one atypicality $C_A(100)$). Then, the number of dimensions is increasing by 18 (Nine typicality from $C_T(100)$ to $C_T(20)$ and nine atypicality from $C_A(100)$ to $C_A(20)$) while the window size decreases by 10 each time. The accuracy for the four cases in Figure 3.8 and 3.9 tends to improve as the number of samples increases. Similar to the previous accuracy, the F1-scores in Figure 3.10 and 3.11 tend to similarly improve. When we use multi-dimensional typicality and atypicality, we obtain relatively stable results.

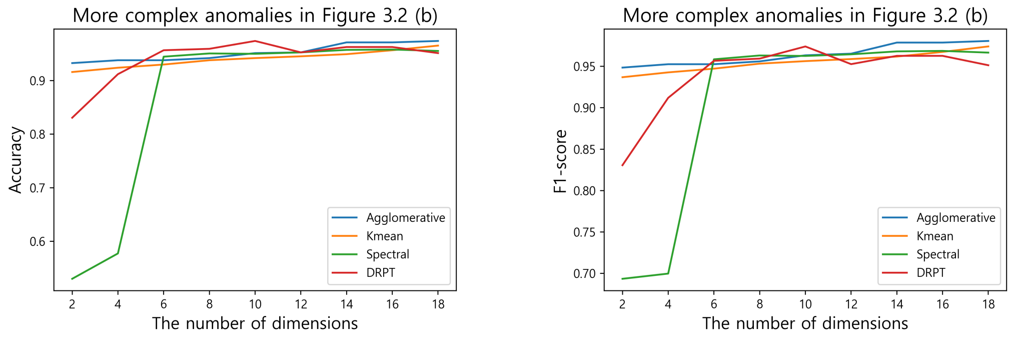


Figure 3.9: Accuracy and F1 score of the case in Figure 3.2 (b) based on the number of dimensions

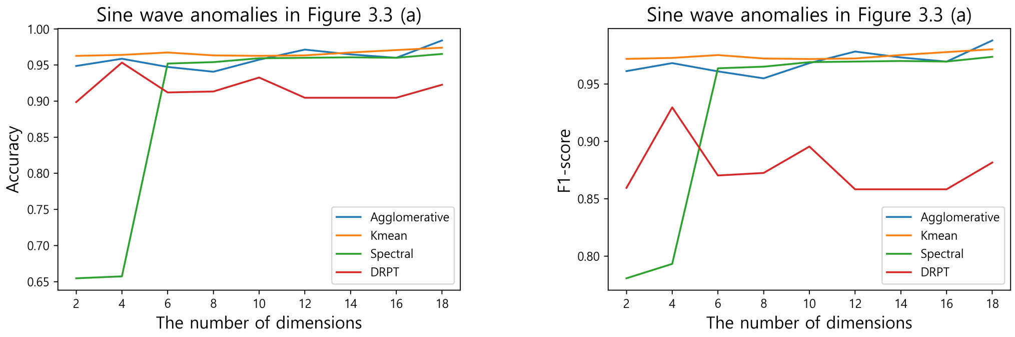


Figure 3.10: Accuracy and F1 score of the case in Figure 3.3 (a) based on the number of dimensions

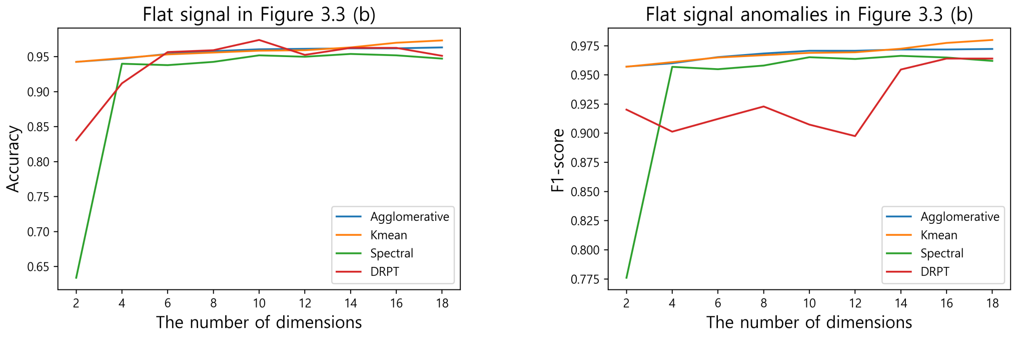


Figure 3.11: Accuracy and F1 score of the case in Figure 3.3 (b) based on the number of dimensions

3.3 Leave-one-out (L1O) kNNG

3.3.1 Introduction

The Leave-one-out (L1O) kNNG was introduced in [35] as a level-set anomaly detection method for multidimensional distributions. First, we must consider why this approach is necessary. In the previous chapter, we demonstrated clustering methods for detecting anomalies when multiple samples are available. Multiple samples from the post anomalous distributions may not be available. However, the L1O-kNNG is capable of performing anomaly detection for a single sample data and is particularly well-suited for handling high-dimensional features. The kNN anomaly detector thresholds the distance between the test point and its k-th nearest neighbor. On the other hand, the L1O-kNNG detector determines the change in the entire kNN graph's topology when a test sample is added, and approximates a level set test of anomalous at a false positive level of 1 - $\frac{k}{n}$, where n is the total number of samples. The specific algorithm will be explained in the next section.

3.3.2 Methodology

Suppose we have a set of n points, denoted as $\mathcal{X}_n = \{X_1, \ldots, X_n\}$. For any point $X_i \in \mathcal{X}_n$, its k nearest neighbors, represented as the k nearest neighbors (kNN) $\left\{X_{i(1)}, \ldots X_{i(k)}\right\}$, are the k points closest to X_i within the set $\mathcal{X}_n - \{X_i\}$. The closeness is determined by the Euclidean distance. The set of edges connecting X_i to its k nearest neighbors is denoted as $\left\{e_{i(1)}, \ldots, e_{i(k)}\right\}_{i=1}^{n}$. The kNN graph (kNNG) over the set $\mathcal{X}$ is formed by the union of all kNN edges $\left\{e_{i(1)}, \ldots, e_{i(k)}\right\}_{i=1}^{n}$. The total power-weighted edge length of the kNN graph is given by:

$$L_{kNN}\left(\mathcal{X}_n\right) = \sum_{i=1}^{n} \sum_{l=1}^{k} \left|e_{i(l)}\right|^{\gamma}$$

The definition of K-point kNNG is as follows and directly from [35]:

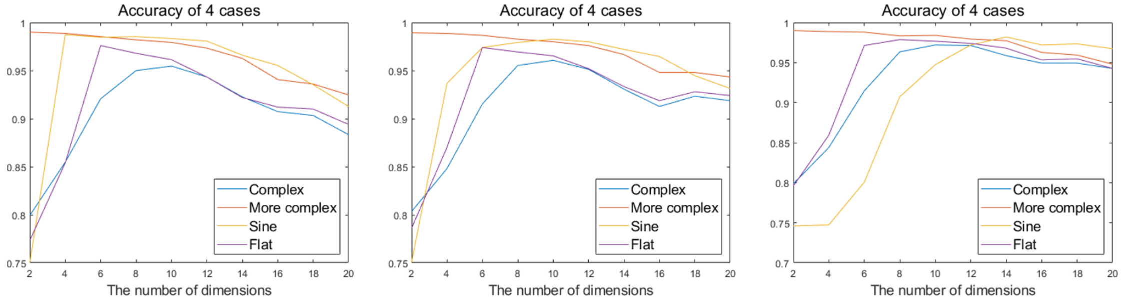
Definition 2 K-point kNNG: Let $\mathcal{X}_{n,K}$ denote one of the $\binom{n}{K}$ subsets of K distinct points from $\mathcal{X}_n$. Among all of the kNNG over each of these sets, the $K-point-kNNG$ is defined as the one having minimal length $\min_{\mathcal{X}_{n,K} \subset \mathcal{X}_n} L_{kNN}\left(\mathcal{X}_{n,k}\right)$.

Algorithm 2 L1O-kNNG anomaly detection algorithm

```
1. For each $X_i \in \mathcal{X}_{n+1}, i = 1, \ldots, n+1$, compute the kNNG total length difference
$\Delta_i L_{kNN} = L_{kNN}\left(\mathcal{X}_{n+1}\right) - L_{kNN}\left(\mathcal{X}_{n+1} - \{X_i\}\right)$ by the following steps. For each $i$ :
    1.(a): Find the $k$ edges $\mathcal{E}^k_{i \to *}$ of all of the kNN's of $X_i$.
    1.(b): Find the edges $\mathcal{E}^k_{* \to i}$ of other points in $\mathcal{X}_{n+1} - \{X_i\}$ that have $X_i$
           as one of their kNNs. For these points find the edges $\mathcal{E}^{k+1}_*$ to their respective
           $k+1$ st NN point.
    1.(c):Compute $\Delta_i L_{kNN} = \sum_{e \in \mathcal{E}^k_{i \to *}} |e|^\gamma + \sum_{e \in \mathcal{E}^k_{* \to i}} |e|^\gamma - \sum_{e \in \mathcal{E}^{k+1}*} |e|^\gamma$
2:  Define the kNNG most ”outlying point” as  $\mathrm{X}_o = \mathrm{argmax}_{i=1,\ldots,n+1} \Delta_i L_{kNN}$.
3 : **Declare** the test sample $X_{n+1}$ an anomaly if $X_{n+1} = X_o$.
```

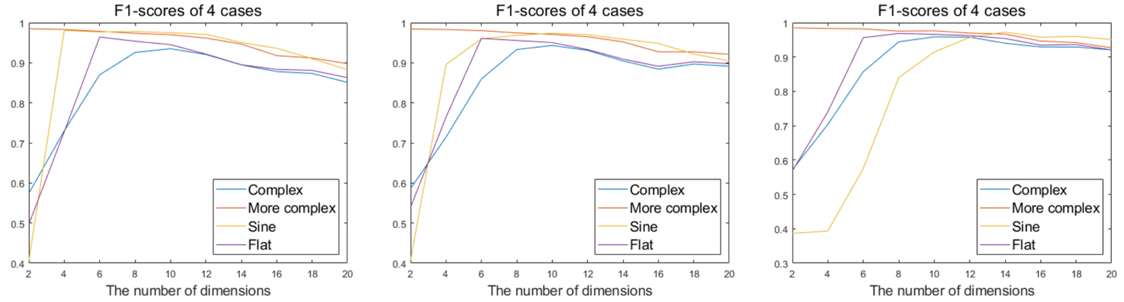
3.3.3 Experiments and Results

The experiment was conducted under the same conditions as the previous Section 3.2.2. As the number of dimensions increases, the detection ranges (time window sizes) are increased from 10 to 100, and multiple trials are performed. Unlike the results in Section 3.2.3, the performance of the L1O-kNNG method does not necessarily increase as the dimensions increase depending on the characteristics of the data. Refer to Figure 3.12 and 3.13. However, the performance is generally better when using multi-dimensional typicality and atypicality compared to using typicality and atypicality independently. Referring to Table 3.1 - 3.2, with the exception of the sine anomaly, the performance is superior when using the high-dimensional score. For this experiment, typicality and atypicality are calculated by $D_max = 20$ and 45 medium resolution (The number of characters is 45). Finally, we compare



(a) k = 5 (b) k = 10 (c) k = 20

Figure 3.12: Accuracy of four cases based on the number of dimensions for different values of the kNN parameter k in the L1O-kNNG. The number of samples = 300.



(a) k = 5 (b) k = 10 (c) k = 20

Figure 3.13: F1-score of four cases based on the number of dimensions for different values of the kNN parameter k in the L1O-kNNG. The number of samples = 300.

the performance of L1O-kNNG to the PDA method proposed in Chapter 2. Refer to Table 3.3. L1O-kNNG with multivariate score gives superior performance to PDA with univariate score computed as the difference between typicality and atypicality.

3.4 Conclusion

In this chapter, we have presented a model that can detect various anomalies by using the 2-dimensional score (typicality, atypicality). Where typicality and atypicality were introduced, multiple sample clustering methods and a single sample L1O-kNNG method were presented. We demonstrate that the 2-dimensional score and high-dimensional score generally give better performance with multiple sample clustering methods without setting a threshold as compared to the one-dimensional scalarization

Anomalies	2-dimensional score			High-dimensional score
	Window size $= 20$	Window size $= 50$	Window size $= 100$	
Figure 3.2.a	0.9140	0.9320	0.9073	**0.9720** ($d = 10$)
Figure 3.2.b	0.9867	0.9593	0.9053	**0.9880** ($d = 6$)
Figure 3.3.a	**0.9867**	0.9727	0.9406	0.9820 ($d = 14$)
Figure 3.3.b	0.9460	0.9453	0.9033	**0.9767** ($d = 10$)

Table 3.1: Comparison by accuracy

Anomalies	2-dimensional score			High-dimensional score
	Window size $= 20$	Window size $= 50$	Window size $= 100$	
Figure 3.2.a	0.8555	0.9056	0.8755	**0.9587** ($d = 10$)
Figure 3.2.b	0.9802	0.9421	0.8725	**0.9821** ($d = 6$)
Figure 3.3.a	**0.9829**	0.9586	0.9149	0.9725 ($d = 14$)
Figure 3.3.b	0.9143	0.9236	0.8715	**0.9660** ($d = 10$)

Table 3.2: Comparison by F1-score

Resolution	PD with Atypicality	L1O-kNNG (d = 10)
Low	0.948 ± 0.011	**0.985** ± 0.006
Medium	0.955 ± 0.010	**0.981** ± 0.006
High	0.948 ± 0.009	**0.966** ± 0.013

Table 3.3: ROC AUC of PDA (Chapter 2) and L1O-kNNG (this Chapter) for the case shown in Figure 3.2.(a). The resolution is set the same as the previous experiment in Table 2.3

of scores used by the PDA method. Finally, we compared the PDA method with the L1O-kNNG method by ROC AUC in Table 3.3 and presented that the L1O-kNNG has better performance.

CHAPTER IV

Application: Pattern Dictionary Method for Image Data

4.1 Introduction

There is no doubt that visual information has been the most reliable source of information for humans from the past to the present. With the rapid growth of digital imaging and the dramatic increase in the utilization of visual data, it has become increasingly important to detect abnormal data in images as they are being used in various fields. Unlike in the past, the current volume of image data has accumulated to such a vast amount that it is impossible for people to classify or detect anomalies. The main objective of image anomaly detection is to pinpoint instances in the data that significantly deviate from what is considered normal, signifying potential problems or rare events that warrant attention. Anomaly detection in image data can be divided into two main areas. The first is the classification of entire images. For example, if we consider images of dogs as normal, then any other images, such as cats, buses, or pictures without dogs, would be considered anomalies. The second area is called sub-image (or pixel-level) anomaly detection, which focuses on identifying anomalous regions or parts within an image. In this chapter, we only deal with the former case. In recent years, a wide variety of methods and techniques have been introduced for image anomaly detection, spanning from traditional statistical approaches to state-of-the-art deep learning techniques. The primary aim of this chapter is to demonstrate the applicability of the pattern dictionary method to image data. By applying the pattern dictionary method to image data, we show that this method can be used not only with time-series data but also with image data in conjunction with pretrained ResNet models.

4.2 Related Works

One-class classification concentrates on studying samples from a single group of interest and the methods can be used as anomaly detection methods for image anomaly detection [91]. One-class support vector machine (OCSVM) is a kernel-based method derived from support vector machines (SVMs) that construct a hyperplane maximizing the distance from the origin, separating outliers from inliers [92]. Another kernel-based one-class classification technique, Support Vector Data Description (SVDD), forms a hypersphere with the smallest radius, encompassing target samples, and regards any sample outside the hypersphere as an outlier [93]. Deep SVDD (Support Vector Data Description) is a one-class classification method that combines the principles of SVDD with deep learning techniques instead of kernels to learn useful feature representations from complex, high-dimensional data and to minimize a hypersphere [94].

4.3 Feature Extraction

It is challenging to directly use 2D or 3D image data in its raw form. Therefore, image feature extraction is a crucial step in computer vision and image processing tasks, as it aims to transform raw image data into a more compact and meaningful representation that captures the key characteristics of the image. This process helps

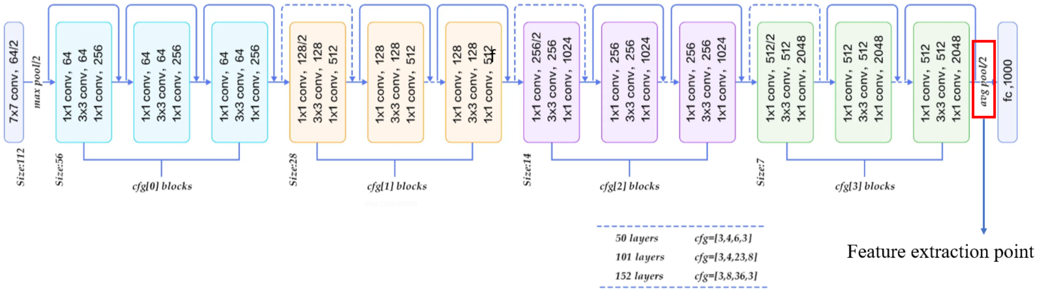


Figure 4.1: Resnet50, 101, 152 Architecture and feature extraction

reduce the dimensionality of data, making it more suitable for various tasks. While there are traditional handcrafted feature extraction methods like SIFT, SURF, and HOG, nowadays, features are often extracted from pre-trained CNN-based models due to their superior performance in capturing image characteristics. In this chapter, the ResNet models [95] are used to extract the image features.

Table 4.1: ROC AUC(% and Std) with MNIST (over 10 seeds) per method

NORMAL CLASS	OC-SVM/ SVDD	KDE	IF	ONE-CLASS DEEP SVDD	PD with Resnet 50	PD with Resnet 101	PD with Resnet 152
0	**98.6** ± 0.0	97.1	98.0 ± 0.3	98.0 ± 0.7	95.6	96.2	96.6
1	99.5 ± 0.0	98.9	97.3 ± 0.4	**99.7** ± 0.1	99.5	99.3	99.3
2	82.5 ± 0.1	79.0	88.6 ± 0.5	**91.7** ± 0.8	86.0	85.3	86.2
3	88.1 ± 0.0	86.2	89.9 ± 0.4	**91.9** ± 1.5	88.8	89.7	91.3
4	94.9 ± 0.0	87.9	92.7 ± 0.6	94.9 ± 0.8	91.4	93.7	**95.4**
5	77.1 ± 0.0	73.8	85.5 ± 0.8	88.5 ± 0.9	87.2	85.9	**91.3**
6	96.5 ± 0.0	87.6	95.6 ± 0.3	**98.3** ± 0.5	93.0	91.3	93.8
7	93.7 ± 0.0	91.4	92.0 ± 0.4	**94.6** ± 0.9	92.0	91.4	93.0
8	88.9 ± 0.0	79.2	89.9 ± 0.4	**93.9** ± 1.6	89.9	90.9	91.1
9	93.1 ± 0.0	88.2	93.5 ± 0.3	**96.5** ± 0.3	93.1	92.3	93.1

4.4 Experiments and Results

4.4.1 MNIST [1] and CIFAR10 [2]

In order to compare our method to others, the MNIST and CIFAR10 datasets were used. Both the MNIST and CIFAR-10 datasets consist of ten distinct classes,

Table 4.2: ROC AUC(% and Std) with CIFAR10 (over 10 seeds) per method.

NORMAL	OC-SVM/	KDE	IF	ONE-CLASS	PD with	PD with	PD with
CLASS	SVDD			DEEP SVDD	Resnet 50	Resnet 101	Resnet 152
AIRPLANE	61.6 ± 0.9	61.2	60.1 ± 0.7	**61.7** ±4.1	54.7	52.7	53.3
AUTOMOBILE	63.8 ± 0.6	64.0	50.8 ± 0.6	65.9 ±2.1	64.4	**68.7**	67.3
BIRD	50.0 ± 0.5	50.1	49.2 ± 0.4	50.8 ± 0.8	51.4	52.6	**53.2**
CAT	55.9 ± 1.3	56.4	55.1 ± 0.4	59.1 ±1.4	57.9	61.5	**62.4**
DEER	66.0 ± 0.7	66.2	49.8 ± 0.4	60.9 ± 1.1	68.7	**71.4**	71.2
DOG	62.4 ± 0.8	62.4	58.5 ± 0.4	**65.7** ± 2.5	53.5	58.1	55.0
FROG	74.7 ± 0.3	**74.9**	42.9 ± 0.6	67.7 ± 2.6	73.1	73.1	73.8
HORSE	62.6 ± 0.6	62.6	55.1 ± 0.7	**67.3** ± 0.9	64.2	64.1	64.6
SHIP	74.9 ± 0.4	75.1	74.2 ± 0.6	**75.9** ± 1.2	60.3	58.9	59.6
TRUCK	75.9 ± 0.3	**76.0**	58.9 ± 0.7	73.1 ± 1.2	65.9	73.2	70.6

which we use to create ten separate one-class classification configurations. In each configuration, one class serves as the normal class, while samples from the other nine classes represent anomalies. We maintain the original training and test splits for our experiments, using only the training set examples corresponding to the respective normal classes. This results in training set sizes of approximately 6,000 for MNIST and 5,000 for CIFAR-10. Each test set contains 10,000 samples, including samples from the nine anomalous classes in every configuration.

In both datasets, we set one class as normal and the others as anomalies. Table 4.1 and 4.2 show the results. For the experiments, we set $D_{\max} = 10$ and medium resolution (the number of characters is 45). The tables include the results adapted from [94], R. Lukas et al.(2018) except for those of the PD (Pattern dictionary) method. Furthermore, we show that the pattern dictionary method can achieve a certain performance level with a small number of training samples in Figure 4.2.

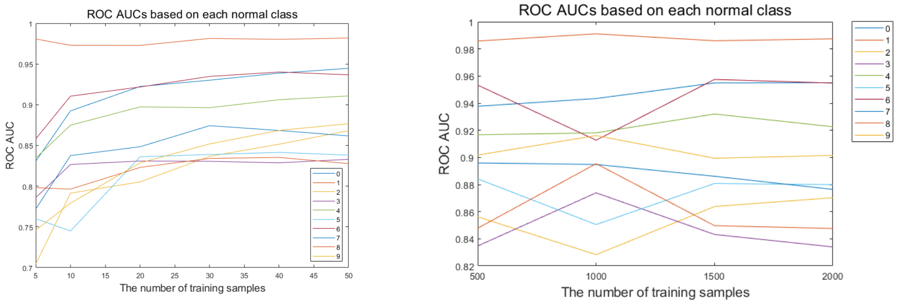


Figure 4.2: ROC AUCs based on the number of training samples. The number of samples is from 5 to 50 (left) and from 500 to 2000 (right). One label is set as normal data and the others are considered as anomalies.

4.4.2 Cats and No cats

In this section, we apply the pattern dictionary method to a new dataset that consists of cat images[1] and indoor scene images[2] without a cat. The motivation for this project stems from an error where a robot designed to detect people indoors mistakenly identifies images with no people as having people present. Of course, various object detection techniques have made it increasingly sophisticated and accurate in detecting objects. However, our method can also be applied when it is necessary to double-check whether the detected image is correct. The dataset consists of 1250 cat images and 250 indoor scene images, respectively. Then, we conducted k-fold cross-validation (k = 5) to evaluate the performance and reliability of our model. The ROC AUC from the test is 0.9248 ± 0.0066. In Figure 4.3 and Figure 4.4, the misdetection cases are shown.

4.5 Conclusion

In this chapter, we applied the pattern dictionary method introduced in Chapter 2 to the image datasets with ResNet50. The pattern dictionary showed reasonable performance on the MNIST dataset which is relatively simple. However, on the

[1] https://www.kaggle.com/datasets/amirhosseinpour/cats-and-dogs-25000-images
[2] https://www.kaggle.com/datasets/itsahmad/indoor-scenes-cvpr-2019

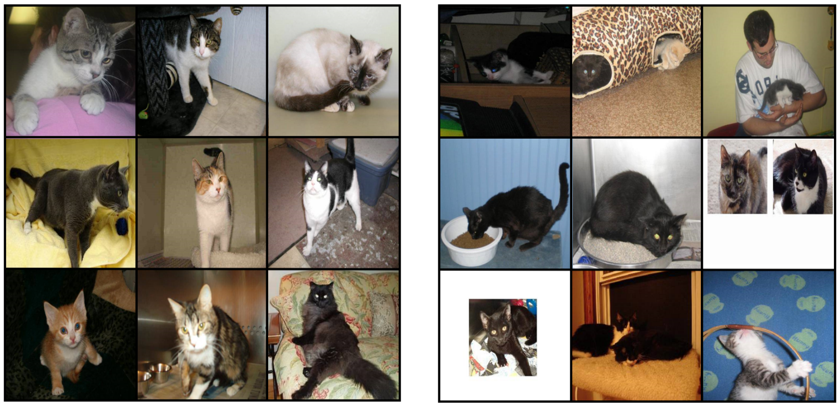


Figure 4.3: Cats: Detected as normal images (left) and anomalous images (right)

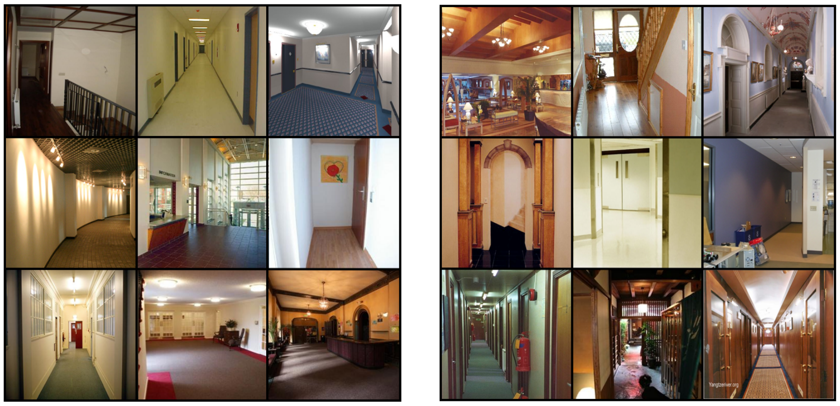


Figure 4.4: Indoor Scene: Detected as normal images (left) and anomalous images (right)

CIFAR10 dataset, its performance was lower in specific classes compared to the One-Class Deep SVDD method. Nevertheless, the pattern dictionary method applied to image data demonstrates a certain level of performance with a small number of training samples as shown in Figure 4.2. Based on this, we evaluated the performance of the pattern dictionary method using images of cats and indoor scenes without a cat. As a result, the ROC AUC is 0.9248 ± 0.0066, indicating reasonable performance.

CHAPTER V

Conclusion

In this thesis, we have presented a universal non-parametric anomaly detection method for time series and image data via the pattern dictionary and atypicality. We illustrated that the proposed pattern dictionary method can be used as a standalone anomaly detector, or integrated with Tree-Structured Lempel– Ziv (LZ78) for atypicality in Chapter 2. Furthermore, we combined the proposed pattern dictionary method with the L1O-kNNG clustering method to utilize high-dimensional typicality and atypicality. Thus, we achieve a better detection performance with a high-dimensional score as shown in Table 3.3 even though the pattern dictionary method with a one-dimensional score performs well for time series data. In Chapter 4, we demonstrated that the pattern dictionary method can be applied to an image dataset with image features from the pretrained ResNet models. The applied method achieved reasonable performance with a small number of training samples compared to other methods.

For future work, the pattern dictionary method can be improved by a new approach for assigning code lengths. By implementing this alternative method, the discrepancy in codelengths between normal and anomalous images can be amplified. Consequently, this enhancement facilitates a more accurate detection of anomalous images.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Yann LeCun, Corinna Cortes, Chris Burges, et al. Mnist handwritten digit database, 2010.

[2] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[4] Anders Høst-Madsen, Elyas Sabeti, and Chad Walton. Data discovery and anomaly detection using atypicality: Theory. *IEEE Transactions on Information Theory*, 65(9):5302–5322, 2019.

[5] André Nies. *Computability and randomness*, volume 51. OUP Oxford, 2012.

[6] Ming Li, Paul Vitányi, et al. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer, 2008.

[7] J. Cover, T.; Thomas. Information theory, 2nd ed. *John Wiley: Hoboken, NJ, USA,*, 2006.

[8] Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.

[9] Elyas Sabeti, Sehong Oh, Peter XK Song, and Alfred O Hero. A pattern dictionary method for anomaly detection. *Entropy*, 24(8):1095, 2022.

[10] Anders Høst-Madsen, Elyas Sabeti, and Chad Walton. Data discovery and anomaly detection using atypicality: Theory. *IEEE Transactions on Information Theory*, 65(9):5302–5322, 2019.

[11] Elyas Sabeti and Anders Høst-Madsen. Data discovery and anomaly detection using atypicality for real-valued data. *Entropy*, 21(3):219, 2019.

[12] Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. On the lambert w function. *Advances in Computational mathematics*, 5:329–359, 1996.

[13] Varun Chandola, Varun Mithal, and Vipin Kumar. Comparative evaluation of anomaly detection techniques for sequence data. In *2008 Eighth IEEE international conference on data mining*, pages 743–748. IEEE, 2008.

[14] Joao BD Cabrera, Lundy Lewis, and Raman K Mehra. Detection and classification of intrusions and faults using sequences of system calls. *Acm sigmod record*, 30(4):25–34, 2001.

[15] Steven A Hofmeyr, Stephanie Forrest, and Anil Somayaji. Intrusion detection using sequences of system calls. *Journal of computer security*, 6(3):151–180, 1998.

[16] Terran Lane and Carla E Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security (TISSEC)*, 2(3):295–331, 1999.

[17] Christina Warrender, Stephanie Forrest, and Barak Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE symposium on security and privacy (Cat. No. 99CB36344)*, pages 133–145. IEEE, 1999.

[18] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215, 2004.

[19] Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley. Compression-based data mining of sequential data. *Data Mining and Knowledge Discovery*, 14:99–129, 2007.

[20] Eamonn Keogh, Li Keogh, and John C Handley. Compression-based data mining. In *Encyclopedia of Data Warehousing and Mining, Second Edition*, pages 278–285. IGI Global, 2009.

[21] Eamonn Keogh, Stefano Lonardi, and Bill'Yuan-chi' Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 550–556, 2002.

[22] Eamonn Keogh, Jessica Lin, Sang-Hee Lee, and Helga Van Herle. Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems*, 11:1–27, 2007.

[23] Thomas S Ferguson. *Mathematical statistics: A decision theoretic approach*, volume 1. Academic press, 2014.

[24] David Siegmund and ES Venkatraman. Using the generalized likelihood ratio statistic for sequential detection of a change-point. *The Annals of Statistics*, pages 255–271, 1995.

[25] So Hirai and Kenji Yamanishi. Detecting changes of clustering structures using normalized maximum likelihood coding. In *Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 343–351, 2012.

[26] Kenji Yamanishi and Kohei Miyaguchi. Detecting gradual changes from data stream using mdl-change statistics. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 156–163. IEEE, 2016.

[27] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[28] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

[29] Herman Chernoff. Sequential design of experiments. *The Annals of Mathematical Statistics*, 30(3):755–770, 1959.

[30] Michele Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. prentice Hall Englewood Cliffs, 1993.

[31] Venugopal V Veeravalli and Taposh Banerjee. Quickest change detection. In *Academic press library in signal processing*, volume 3, pages 209–255. Elsevier, 2014.

[32] Chunming Han, Peter Willett, Biao Chen, and Douglas Abraham. A detection optimal min-max test for transient signals. *IEEE Transactions on Information Theory*, 44(2):866–869, 1998.

[33] Zhen Wang and Peter Willett. A performance study of some transient detectors. *IEEE transactions on signal processing*, 48(9):2682–2685, 2000.

[34] Zhen Wang and Peter K Willett. All-purpose and plug-in power-law detectors for transient signals. *IEEE transactions on signal processing*, 49(11):2454–2466, 2001.

[35] Alfred Hero. Geometric entropy minimization (gem) for anomaly detection and localization. *Advances in neural information processing systems*, 19, 2006.

[36] Kumar Sricharan and Alfred Hero. Efficient anomaly detection using bipartite k-nn graphs. *Advances in Neural Information Processing Systems*, 24, 2011.

[37] Pranab Kumar Sen. *Theory and applications of sequential nonparametrics*. SIAM, 1985.

[38] Akshay Balsubramani and Aaditya Ramdas. Sequential nonparametric testing with the law of the iterated logarithm. *arXiv preprint arXiv:1506.03486*, 2015.

[39] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection for discrete sequences: A survey. *IEEE transactions on knowledge and data engineering*, 24(5):823–839, 2010.

[40] Scott Evans, Bruce Barnett, Stephen F Bush, and Gary J Saulnier. Minimum description length principles for detection and classification of ftp exploits. In *IEEE MILCOM 2004. Military Communications Conference, 2004.*, volume 1, pages 473–479. IEEE, 2004.

[41] Nan Wang, Jizhong Han, and Jinyun Fang. An anomaly detection algorithm based on lossless compression. In *2012 IEEE Seventh International Conference on Networking, Architecture, and Storage*, pages 31–38. IEEE, 2012.

[42] Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pages 130–143. IEEE, 2000.

[43] Ioannis Ch Paschalidis and Georgios Smaragdakis. Spatio-temporal network anomaly detection by assessing deviations of empirical measures. *IEEE/ACM Transactions On Networking*, 17(3):685–697, 2008.

[44] Chan-Kyu Han and Hyoung-Kee Choi. Effective discovery of attacks using entropy of packet dynamics. *IEEE network*, 23(5):4–12, 2009.

[45] Priya Baliga and TY Lin. Kolmogorov complexity based automata modeling for intrusion detection. In *2005 IEEE International Conference on Granular Computing*, volume 2, pages 387–392. IEEE, 2005.

[46] Hossain Shahriar and Mohammad Zulkernine. Information-theoretic detection of sql injection attacks. In *2012 IEEE 14th international symposium on high-assurance systems engineering*, pages 40–47. IEEE, 2012.

[47] Yang Xiang, Ke Li, and Wanlei Zhou. Low-rate ddos attacks detection and traceback by using new information metrics. *IEEE transactions on information forensics and security*, 6(2):426–437, 2011.

[48] Feng Pan and Weinong Wang. Anomaly detection based-on the regularity of normal behaviors. In *2006 1st International Symposium on Systems and Control in Aerospace and Astronautics*, pages 6–pp. IEEE, 2006.

[49] Ming Li, Xin Chen, Xin Li, Bin Ma, and Paul MB Vitányi. The similarity metric. *IEEE transactions on Information Theory*, 50(12):3250–3264, 2004.

[50] E Earl Eiland and Lorie M Liebrock. An application of information theory to intrusion detection. In *Fourth IEEE International Workshop on Information Assurance (IWIA'06)*, pages 16–pp. IEEE, 2006.

[51] Yun Li, Sirin Nitinawarat, and Venugopal V Veeravalli. Universal outlier hypothesis testing. *IEEE Transactions on Information Theory*, 60(7):4066–4082, 2014.

[52] Yun Li, Sirin Nitinawarat, and Venugopal V Veeravalli. Universal outlier detection. In *2013 Information Theory and Applications Workshop (ITA)*, pages 1–5. IEEE, 2013.

[53] Jacob Ziv and Neri Merhav. A measure of relative entropy between individual sequences with application to universal classification. *IEEE transactions on information theory*, 39(4):1270–1279, 1993.

[54] Varun Chandola. *Anomaly detection for symbolic sequences and time series data*. University of Minnesota, 2009.

[55] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John wiley & sons, 2005.

[56] Qingtao Wu and Zhiqing Shao. Network anomaly detection using time series analysis. In *Joint international conference on autonomic and autonomous systems and international conference on networking and services-(icas-isns' 05)*, pages 42–42. IEEE, 2005.

[57] Brandon Pincombe. Anomaly detection in time series of graphs using arma processes. *Asor Bulletin*, 24(4):2, 2005.

[58] H Zare Moayedi and MA Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *2008 international symposium on information technology*, volume 4, pages 1–6. IEEE, 2008.

[59] Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–618, 2003.

[60] Florian Knorn and Douglas J Leith. Adaptive kalman filtering for anomaly detection in software appliances. In *IEEE INFOCOM Workshops 2008*, pages 1–6. IEEE, 2008.

[61] Dan Gusfield. Algorithms on stings, trees, and sequences: Computer science and computational biology. *Acm Sigact News*, 28(4):41–60, 1997.

[62] Marina Thottan and Chuanyi Ji. Anomaly detection in ip networks. *IEEE Transactions on signal processing*, 51(8):2191–2204, 2003.

[63] Soumen Chakrabarti, Sunita Sarawagi, and Byron Dom. Mining surprising patterns using temporal description length. In *VLDB*, volume 98, pages 606–617. Citeseer, 1998.

[64] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29:626–688, 2015.

[65] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F Samatova. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247, 2015.

[66] Rose Yu, Huida Qiu, Zhen Wen, ChingYung Lin, and Yan Liu. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 18(1):1–14, 2016.

[67] Charu C Aggarwal and Philip S Yu. An effective and efficient algorithm for high-dimensional outlier detection. *The VLDB journal*, 14:211–221, 2005.

[68] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: poster and demo track*, 1:59–63, 2012.

[69] Ralph Foorthuis. Secoda: Segmentation-and combination-based detection of anomalies. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 755–764. IEEE, 2017.

[70] Ralph Foorthuis. The impact of discretization method on the detection of six types of anomalies in datasets. *arXiv preprint arXiv:2008.12330*, 2020.

[71] John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

[72] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

[73] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[74] SH Tung. On lower and upper bounds of the difference between the arithmetic and the geometric mean. *Mathematics of Computation*, 29(131):834–836, 1975.

[75] Frans MJ Willems. The context-tree weighting method: Extensions. *IEEE Transactions on Information Theory*, 44(2):792–798, 1998.

[76] Frans MJ Willems, Yuri M Shtarkov, and Tjalling J Tjalkens. The context-tree weighting method: Basic properties. *IEEE transactions on information theory*, 41(3):653–664, 1995.

[77] Frans Willems, Yuri Shtarkov, and Tjalling Tjalkens. Reflections on "the context tree weighting method: Basic properties". *Newsletter of the IEEE Information Theory Society*, 47(1), 1997.

[78] Abdolhossein Hoorfar and Mehdi Hassani. Inequalities on the lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008.

[79] Abraham Lempel and Jacob Ziv. On the complexity of finite sequences. *IEEE Transactions on information theory*, 22(1):75–81, 1976.

[80] Philippe Jacquet and Wojciech Szpankowski. Limiting distribution of lempel ziv'78 redundancy. In *2011 IEEE International Symposium on Information Theory Proceedings*, pages 1509–1513. IEEE, 2011.

[81] En-Hui Yang and Jin Meng. Non-asymptotic equipartition properties for independent and identically distributed sources. In *2012 Information Theory and Applications Workshop*, pages 39–46. IEEE, 2012.

[82] Michael C Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[83] Allen Gersho and Robert M Gray. *Vector quantization and signal compression*, volume 159. Springer Science & Business Media, 2012.

[84] Emilia Grzesiak, Brinnae Bent, Micah T McClain, Christopher W Woods, Ephraim L Tsalik, Bradly P Nicholson, Timothy Veldman, Thomas W Burke, Zoe Gardener, Emma Bergstrom, et al. Assessment of the feasibility of using noninvasive wearable biometric monitoring sensors to detect influenza and the common cold before symptom onset. *JAMA network open*, 4(9):e2128534–e2128534, 2021.

[85] Xichen She, Yaya Zhai, Ricardo Henao, Christopher W Woods, Christopher Chiu, Geoffrey S Ginsburg, Peter XK Song, and Alfred O Hero. Adaptive multi-channel event segmentation and feature extraction for monitoring health outcomes. *IEEE Transactions on Biomedical Engineering*, 68(8):2377–2388, 2020.

[86] Laurent Galluccio, Olivier Michel, Pierre Comon, Mark Kliger, and Alfred O Hero. Clustering with a new distance measure based on a dual-rooted tree. *Information Sciences*, 251:96–113, 2013.

[87] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[88] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[89] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[90] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

[91] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.

[92] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[93] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54:45–66, 2004.

[94] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[95] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[96] Z Jane Wang and Peter Willett. A variable threshold page procedure for detection of transient signals. *IEEE transactions on signal processing*, 53(11):4397–4402, 2005.

[97] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[98] Muhammad H Arshad and Philip K Chan. Identifying outliers via clustering for anomaly detection. Technical report, 2003.

[99] Alejandro Marcos Alvarez, Makoto Yamada, Akisato Kimura, and Tomoharu Iwata. Clustering-based anomaly detection in multi-view data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1545–1548, 2013.

[100] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. Unsupervised clustering approach for network anomaly detection. In *Networked Digital Technologies: 4th International Conference, NDT 2012, Dubai, UAE, April 24-26, 2012. Proceedings, Part I 4*, pages 135–145. Springer, 2012.

[101] Jie Yang, Ruijie Xu, Zhiquan Qi, and Yong Shi. Visual anomaly detection for images: A systematic survey. *Procedia Computer Science*, 199:471–478, 2022.

[102] Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[103] Nathalie Japkowicz, Catherine Myers, Mark Gluck, et al. A novelty detection approach to classification. In *IJCAI*, volume 1, pages 518–523. Citeseer, 1995.

[104] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014.

[105] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.