entropy

Article

A Pattern Dictionary Method for Anomaly Detection

Elyas Sabeti [1], Sehong Oh [2], Peter X. K. Song [3] and Alfred O. Hero [2,*]

[1] Michigan Institute for Data Science, University of Michigan, Ann Arbor, MI 48109, USA
[2] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
[3] Department of Biostatistics, University of Michigan, Ann Arbor, MI 48109, USA; pxsong@umich.edu
* Correspondence: hero@umich.edu

Abstract: In this paper, we propose a compression-based anomaly detection method for time series and sequence data using a pattern dictionary. The proposed method is capable of learning complex patterns in a training data sequence, using these learned patterns to detect potentially anomalous patterns in a test data sequence. The proposed pattern dictionary method uses a measure of complexity of the test sequence as an anomaly score that can be used to perform stand-alone anomaly detection. We also show that when combined with a universal source coder, the proposed pattern dictionary yields a powerful atypicality detector that is equally applicable to anomaly detection. The pattern dictionary-based atypicality detector uses an anomaly score defined as the difference between the complexity of the test sequence data encoded by the trained pattern dictionary (typical) encoder and the universal (atypical) encoder, respectively. We consider two complexity measures: the number of parsed phrases in the sequence, and the length of the encoded sequence (codelength). Specializing to a particular type of universal encoder, the Tree-Structured Lempel–Ziv (LZ78), we obtain a novel non-asymptotic upper bound, in terms of the Lambert W function, on the number of distinct phrases resulting from the LZ78 parser. This non-asymptotic bound determines the range of anomaly score. As a concrete application, we illustrate the pattern dictionary framework for constructing a baseline of health against which anomalous deviations can be detected.

Keywords: pattern dictionary; atypicality; Lempel–Ziv algorithm; lossless compression; anomaly detection

check for updates

Citation: Sabeti, E.; Oh, S.; Song, P.X.K.; Hero, A.O. A Pattern Dictionary Method for Anomaly Detection. *Entropy* **2022**, *24*, 1095. https://doi.org/10.3390/e24081095

Academic Editors: Chao Tian and Lizhong Zheng

Received: 14 July 2022
Accepted: 7 August 2022
Published: 9 August 2022

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 [1]. 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 leads 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 [2,3] 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) [4,5]. We use the LZ78 as the universal encoder since its compression procedure is similar to our proposed pattern dictionary, and it is (asymptotically) optimal [4,5].

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 [6]. 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 [7], threshold sequence time-delay embedding [8–11], and compression-based dissimilarity measure [12–15,15,16], 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, we briefly review the relevant literature in anomaly detection (readers who are familiar with anomaly detection can skip this section). Section 3 introduces the detection framework and the notation used in this paper. Section 4 presents our proposed pattern dictionary method and its properties. In Section 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 6 presents experiments that illustrate the proposed pattern dictionary anomaly detection procedure. Finally, Section 7 concludes our paper.

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 [17], change detection [18–21], concept drift detection [19–22], minimax quickest change detection (MQCD) [23–25], and transient detection [26–29] 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) [30,31] 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 [32,33]. 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 [34–45]. 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 [46,47]. 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 [48], 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 [48] is similar to the pattern dictionary used in this paper, there are important differences. The empirical measure proposed in [48] 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 [48], is only guaranteed when the sequences satisfy the finite-order Markov property, a condition that may be difficult to satisfy in practice. In [2,3], 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 [2]) 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 [49]. Various time series modeling techniques such as regression [50], auto regression [51], auto regression moving average [52], auto regressive integrated moving average [53], support vector regression [54], and Kalman filters [55] 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 [7,56], detection of cyberattacks in computer network [57], and detection of irregular behaviors in online banking [58] 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 [59], dynamic networks [60], and social networks [61]. Chandola et al. [34] 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) [7] are designed to detect anomalous sequences that are dissimilar to the training data. As another example, threshold sequence time-delay embedding (t-STIDE) [8–11] is established to detect anomalous sequences that contain subsequences with anomalous occurrence frequencies. The compression-based dissimilarity measure (CDM) is proposed for discord detection [12–15,15,16] to detect anomalous subsequences within a long sequence. Chandola et al. [34] 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 [34]. 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 [2]. 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 6, the detection performance of our proposed pattern dictionary method is compared with NNS, CDM, t-STIDE, and the Ziv–Merhav method of [48].

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 [62–65]. Note that discretization is also a requirement in other problem settings such as continuous optimization in genetic algorithms [66], image pattern recognition [67], and nonparametric histogram matching over codebooks in computer vision [68].

3. Framework and Notation

In the anomaly detection literature for sequence data and time series, the following three general formulations are considered [34]: (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: $x_n^m = \{x_i, i = n, n+1, \dots, m\}$, and x^l represents a sequence of length l, i.e., $\{x_n, n = 1, \dots, l\}$. $\mathcal{X}$ 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.

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 D_{max} and the set of observed patterns $\mathcal{S}_{\mathcal{D}}\left(x_1^L\right)$.

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 [2]), 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 depth-dependent 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.

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 $L^{multi}_{max} = \log D_{max} + D_{max}\log(|\mathcal{X}|)$ bits. Thus, the average indexing codelength of a multi-level dictionary is given by

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

Figures 1 and 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.

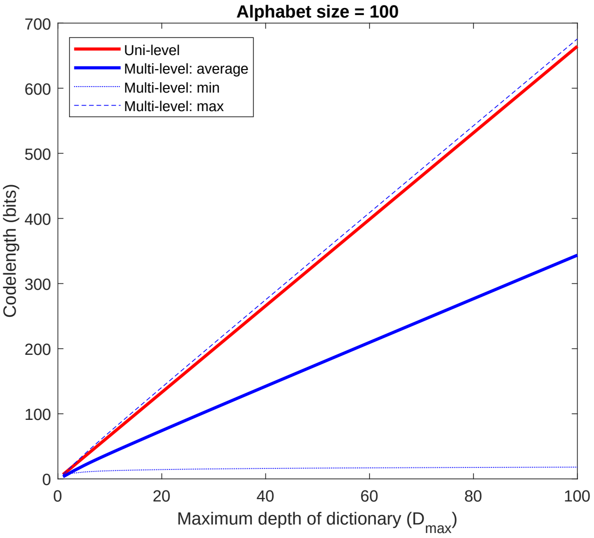

Figure 1. Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed alphabet size $|\mathcal{X}| = 100$).

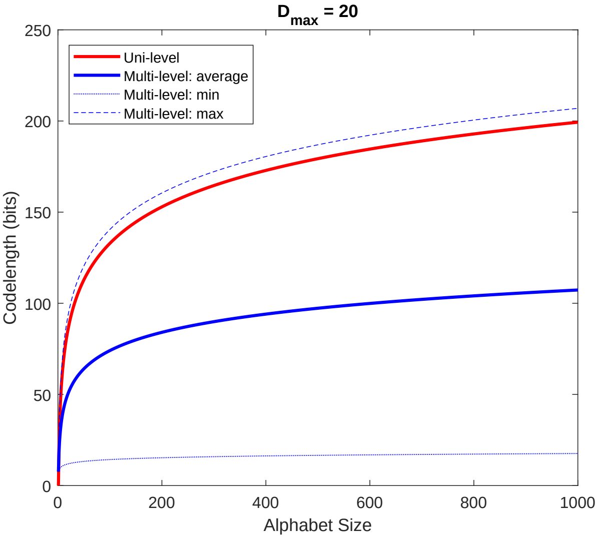

Figure 2. Comparison of indexing codelength between a uni-level dictionary and a multi-level dictionary (fixed $D_{max} = 20$).

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, 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) $L^{multi} \leq L^{uni}$; and

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

where

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

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

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

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

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

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 [69]

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

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 $L^{uni} - L^{multi} = \log \frac{A_{D_{max}}}{G_{D_{max}}}$, the proof is complete. □

Theorem 1 shows that a multi-level dictionary gives 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 [4] 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

$$L^{multi}\left(x_1^d\right) = \log D_{max} + L_{\mathcal{D}}^{(d)}\left(x_1^d\right), \tag{1}$$

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 ([4] 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 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.*

Table 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	$\Pr(x_1^d)$	$L_{\mathcal{D}}^{(1)}(x_1^d)$	x_1^d	$\Pr(x_1^d)$	$L_{\mathcal{D}}^{(2)}(x_1^d)$	x_1^d	$\Pr(x_1^d)$	$L_{\mathcal{D}}^{(3)}(x_1^d)$
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

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 ith phrase, and each phrase $x_{v_i}^{v_{i+1}-1}$ is a pattern in the pattern 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 $v_2 \le D_{max}$ and $v_2 \le l$ 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 $v_3 \le D_{max}$ and $v_3 \le l$ 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 [48] 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

$$L\left(x_1^l\right) = \sum_{i=1}^{c} L_{\mathcal{D}}\left(x_{v_i}^{v_{i+1}-1}\right) + c \log D_{max}. \tag{2}$$

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 $v_c + d - 1 < l$ 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 $x_{v_1}^{v_2-1}, x_{v_2}^{v_3-1}, \ldots, x_{v_c}^{l}$
```

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 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 [2,3] as a training-based fixed source coder (data-dependent encoder).

5. Pattern Dictionary-Based Atypicality (PDA)

In [2,3], 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 [2,3]. 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 atypicality (PDA) method.

The pattern dictionary-based source coder can be considered as a generalization of the Context Tree [70–72] based fixed source coder that was used in [2] for discrete data. The universal source coder (atypical encoder) used here is the Tree-Structured Lempel–Ziv (LZ78) [4,5]. 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 4, and it is (asymptotically) optimal [4,5]. 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 [2]. 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 $\widehat{p}=1$, the typical encoder cannot catch the anomaly while the atypical encoder can. On the other hand, if $p=\frac{1}{3}$ and $\widehat{p}=\frac{2}{3}$, the typical encoder identifies the anomaly while an atypical encoder fails to do so (since the entropy for $p=\frac{1}{3}$ and $\widehat{p}=\frac{2}{3}$ 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 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), 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 [4,5]

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

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 [4,5], 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.

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*

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

where $\alpha = |\mathcal{X}|-1$, $\beta = (|\mathcal{X}|-1)^2 l - |\mathcal{X}|$, and $\mathrm{W}(.)$ is the Lambert W function [6].

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,

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

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

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

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

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

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

$$\begin{aligned} c(l) &\leq c(l_k) + \frac{l - l_k}{k+1} \leq \frac{l_k}{k - \frac{1}{M-1}} + \frac{\triangle}{k+1} \\ &\leq \frac{l_k + \triangle}{k - \frac{1}{M-1}} = \frac{l}{k - \frac{1}{M-1}}. \end{aligned} \tag{3}$$

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,

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

where $\widehat{k} = \alpha k - 1$. The last equation can be solved using the Lambert W function [6]. 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

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

where W(.) is the Lambert W function. Using equation (3), we write

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

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 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.

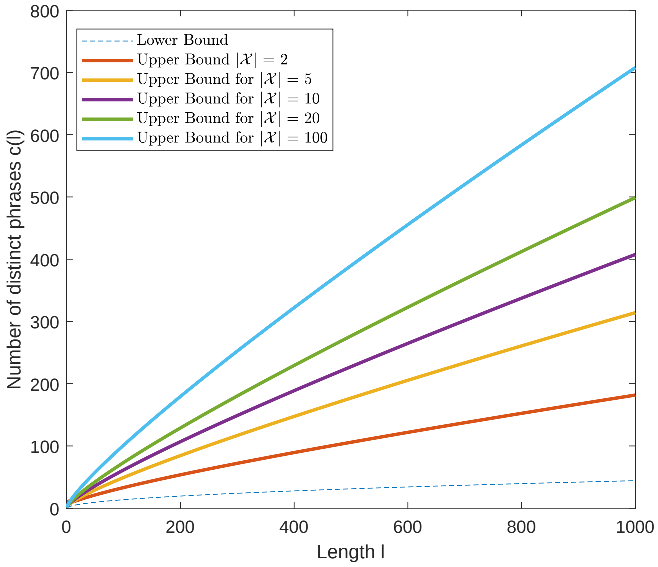

Figure 3. Plot of the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of an $|\mathcal{X}|$-ary sequence of length l.

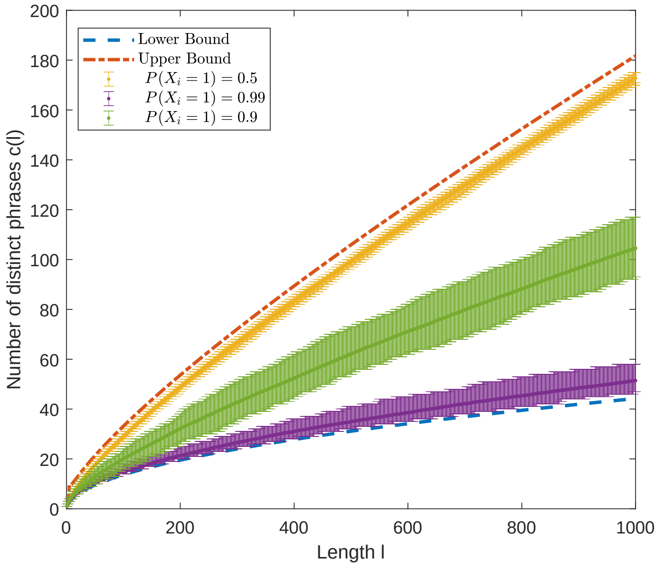

Figure 4. Simulation results compared to the lower and upper bounds of Theorem 2 on the number of distinct phrases resulting from LZ78-parsing of binary sequences of length l generated by sources with three different source probabilities $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

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 on Lambert W function $\ln x - \ln(\ln x) \leq \mathrm{W}(x)$ [73], we write

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

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

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

where $\widehat{c_M} = \log c_M - \log\log(c_M \ln M)$. Therefore, as $l \to \infty$, we have $c(l) \leq \frac{l}{\log l}$. This is consistent with the binary case $M = 2$ proved in ([4] Lemma 13.5.3) or [5]. The following Lemma extends the result of ([4] 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 ([4] Lemma 13.5.3) or ([74] 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 $c(l) \leq c(l_k) + \frac{l-l_k}{k+1} \leq \frac{l}{k-\frac{1}{M-1}}$. 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,

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

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

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

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

Next, we analyze the properties of the number of distinct phrases $c(l)$ resulting from LZ78-parsing of an $|\mathcal{X}|$-ary sequence $x_1^l = \{x_n, n=1,\dots,l\}$ when l is fixed. The error bar representation in Figure 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 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 ([75] Theorem 1), for sequences generated from a memoryless source, $c(l)$ is assumed to be a random variable with the following mean and variance:

$$
\begin{aligned}
\mathrm{E}(c(l)) &\sim \frac{hl}{\log l}, \\
\mathrm{Var}(c(l)) &\sim \frac{(h_2-h^2)l}{\log^2 l},
\end{aligned} \tag{4}
$$

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 (4) are asymptotic as $l \to \infty$. Below, we obtain a finite sample characterization of $c(l)$.

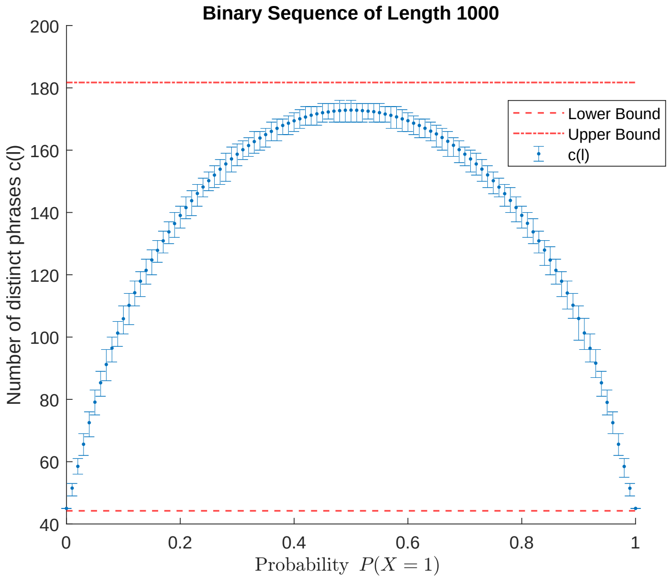

Figure 5. Similar to Figure 4, the number of distinct phrases resulting from LZ78-parsing of binary sequences of fixed length $l = 1000$ varies over the source probability parameter $P(X = 1)$. For every $P(X = 1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

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) ([4] Chapter 3) or Non-asymptotic Equipartition Properties (NEP) [76], 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

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

where h is the entropy. Then, we have

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

therefore, $\left|\mathcal{A}_\epsilon^{(n)}\right| \le 2^{n(h+\epsilon)}$. 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,

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

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

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

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 $\epsilon = 0.1$.

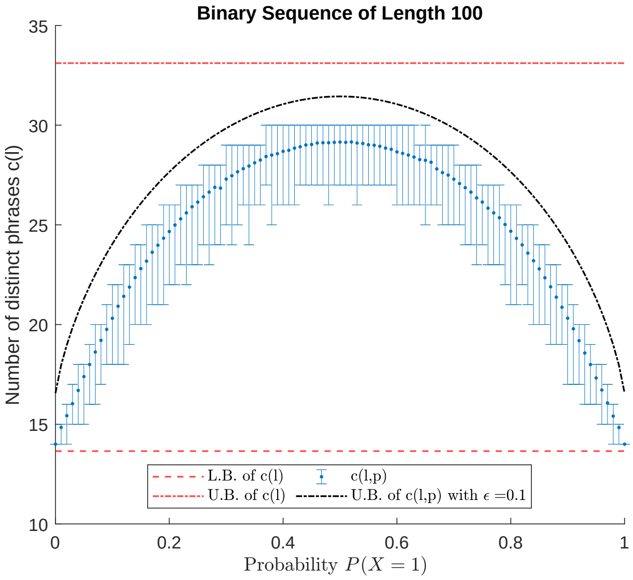

Figure 6. Simulation of the probability-dependent upper bound $c(l,p)$ for binary sequences of fixed length $l = 100$ with various probability parameters $P(X=1)$. For every $P(X=1)$, one thousand binary sequences of length l are generated. Error bars represent the maximum, minimum, and average number of distinct phrases.

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 typical 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, $\ln x - \ln(\ln x) \leq \mathrm{W}(x)$ [73], we have

$$\begin{aligned} \frac{l}{D_{max}}\left(1-\frac{D_{max}}{\log\frac{l}{\log(l\ln|\mathcal{X}|)}}\right) &\leq c_T(l)-c_A(l) \\ &\leq l\left(1-\frac{\sqrt{8l+1}-1}{2l}\right). \end{aligned} \quad (5)$$

The above bounds have asymptotic and non-asymptotic implications. The asymptotic analysis of the bounds in (5) suggests that as $l \to \infty$, for a dictionary with fixed D_{max}, we have $\frac{l}{D_{max}} \le c_T(l) - c_A(l) \le l$. 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 $D_{max} > \log \frac{l}{\log(l \ln|\mathcal{X}|)}$ as $l \to \infty$, then $c_T(l)$ can be smaller than $c_A(l)$. The non-asymptotic behavior of the bounds in (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.

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 $x_n^{n+l-1} = \{x_n, n = n, \dots, n+l-1\}$ embedded in a test sequence $\{x_n, n = 1, \dots, L\}$ from the finite alphabet $\mathcal{X}$. Using Equation (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+l-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 [2,3], and $\log^*(l) = \log l + \log\log l + \dots$ 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:

$$\triangle L(n) = \max_{l}\left\{L_T\left(x_n^{n+l-1}\right) - L_A^{'}\left(x_n^{n+l-1}\right)\right\} > \tau, \tag{6}$$

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.

6. Experiment

In this section, we illustrate the proposed pattern dictionary anomaly detection on a synthetic time series, known as Mackey–Glass [77], 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 [78], and then, anomaly detection methods are applied.

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 [77]. The nonlinear differential equation is of the form $\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-\delta)}{1+x^{10}(t-\delta)}$, $t \ge 0$, 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 7 shows a realization of the training data and the test data.

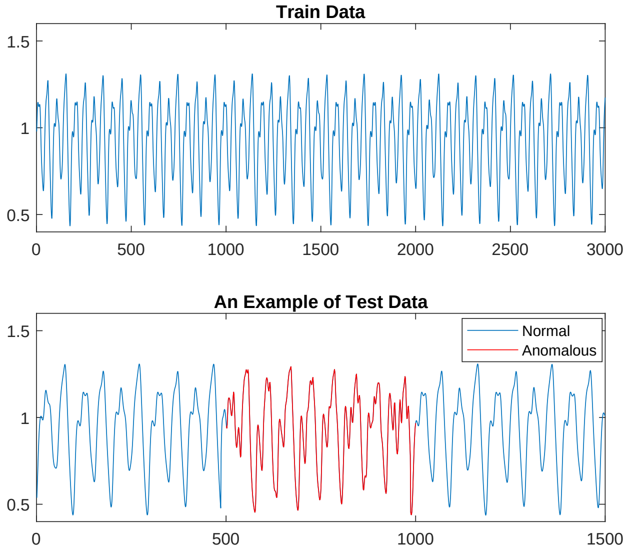

Figure 7. Mackey–Glass time series: the training data (**top**) and an example of the test data (**bottom**) in which samples in $[501, 1000]$ are anomalous (shown in red).

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 subsequences in the test data using Equation (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 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 (5).

Next, we compare anomaly detection performance of our proposed pattern dictionary methods, PDD and PDA, with the nearest neighbors-based similarity (NNS) technique [7], the compression-based dissimilarity measure (CDM) method [12–14], Ziv–Merhav method (ZM) [48], and the threshold Sequence Time-Delay Embedding (t-STIDE) technique [8–11]. 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. 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.

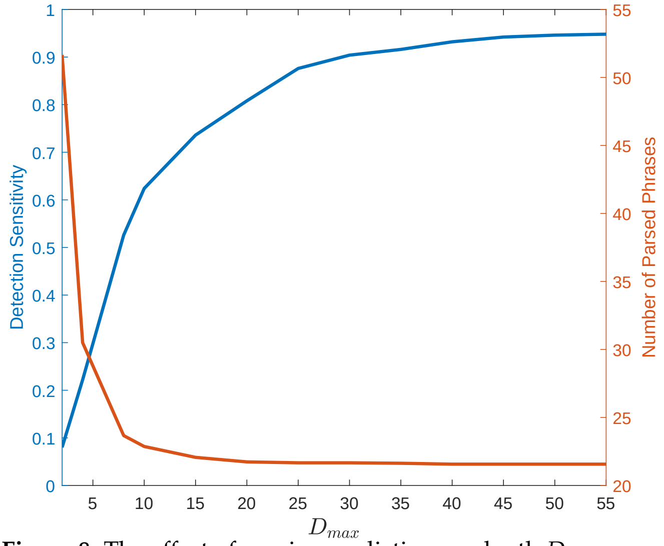

Figure 8. The effect of maximum dictionary depth D_{max} on parsing and detection sensitivity (true positive rate) of the Mackey–Glass time series presented in Figure 7.

Pattern Dictionary for Detection (PDD)

First, the training data are used to create a pattern dictionary with $D_{max} = 40$, as described in Section 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) described in Section 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 4. Then, for each subsequence x^{100} of the test data, the anomaly score is the atypicality measure described in Section 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) [48]

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) [7]

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) [12–14]

In this method, given the training data x_{train}, for each subsequence x^{100} of the test data the anomaly score is

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

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) [8–11]

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. 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 [78]) on anomaly detection performance is summarized in Table 3. Table 3 shows that our proposed PDD and PDA methods outperform in all three cases of data resolution.

Table 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	**0.963 ± 0.009**	**0.909 ± 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 [48] 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 [48], 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 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 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

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 2.60 GHz CPU, 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

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 [79]. Volunteers' skin temperature 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 [80]. 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 [78] 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 days), we used the result of Section 5.3 and the atypicality criterion of Equation (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 9 illustrates the result of anomaly detection on one subject who became infected as measured by viral shedding as shown in Figure 9C. All the anomalous patterns occur when the subject was shedding the virus. Figure 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 10C. Note that in this case, no anomalous patterns were detected.

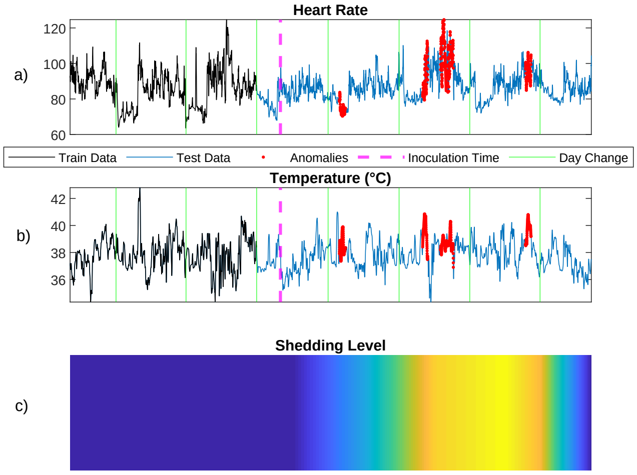

Figure 9. Anomaly detection using the proposed PDA method for a subject based on heart rate and temperature data collected from a wearable wrist sensor. Anomalies are shown in red in (**a**,**b**). (**c**) shows the subject's infection level.

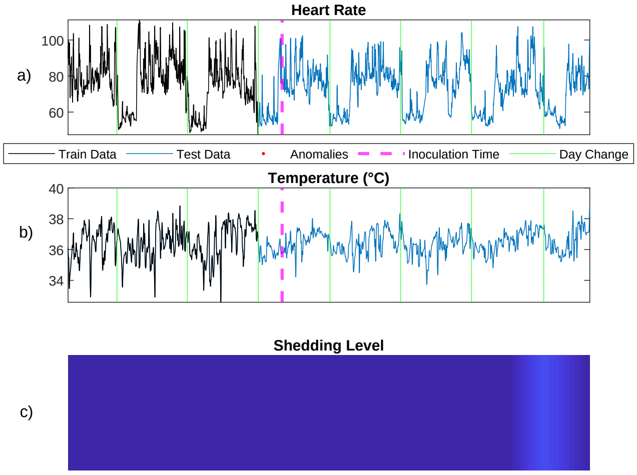

Figure 10. Anomaly detection using the proposed PDA method for a subject who had a mild infection with low level of viral shedding based on heart rate and temperature data collected from a wearable wrist sensor. Note that no anomaly has been detected: (**a**) heart rate, (**b**) temperature, and (**c**) infection level.

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 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 Lempel–Ziv (LZ78) and incorporated into an atypicality framework. We developed novel non-asymptotic lower and upper bounds of the LZ78 parser and demonstrated that the non-asymptotic 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.

Author Contributions: Data curation, E.S. and S.O.; Formal analysis, E.S.; Funding acquisition, A.O.H.; Methodology, E.S.; Project administration, A.O.H.; Software, E.S. and S.O.; Supervision, A.O.H.; Validation, E.S., P.X.K.S. and A.O.H.; Visualization, E.S. and S.O.; Writing—original draft, E.S.; Writing—review & editing, P.X.K.S. and A.O.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Michigan Institute for Data Science, and grants from the Army Research Office, grant W911NF-15-0479, the Defense Advanced Research Projects Agency, grant N66001-17-2-401, and the Department of Energy/National Nuclear Security Administration, grant DE-NA0003921.

Institutional Review Board Statement: University of Michigan ethical review and approval were waived since only de-identified data artifacts from a previously approved human challenge study were made available to the co-authors for the analysis reported in Section 6.2 Information concerning provenance of the data, i.e., the human rhinovirus (RV) challenge study experiment and its IRB approved protocol, was reported in [79].

Informed Consent Statement: The de-identified data used for our analysis in Section 6.2 came from a human rhinovirus (RV) challenge study in which informed consent was obtained from all subjects involved in the study. See the [79] for details.

Data Availability Statement: The experimental data used in Section 6.2 will be made available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 15. [CrossRef]
2. Høst-Madsen, A.; Sabeti, E.; Walton, C. Data discovery and anomaly detection using atypicality: Theory. *IEEE Trans. Inf. Theory* **2019**, *65*, 5302–5322. [CrossRef]
3. Sabeti, E.; Høst-Madsen, A. Data Discovery and Anomaly Detection Using Atypicality for Real-Valued Data. *Entropy* **2019**, *21*, 219. [CrossRef] [PubMed]
4. Cover, T.; Thomas, J. *Information Theory*, 2nd ed.; John Wiley: Hoboken, NJ, USA, 2006.
5. Ziv, J.; Lempel, A. Compression of individual sequences via variable-rate coding. *Inf. Theory IEEE Trans.* **1978**, *24*, 530–536. [CrossRef]
6. Corless, R.M.; Gonnet, G.H.; Hare, D.E.; Jeffrey, D.J.; Knuth, D.E. On the LambertW function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [CrossRef]
7. Chandola, V.; Mithal, V.; Kumar, V. Comparative evaluation of anomaly detection techniques for sequence data. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 743–748.

8. Cabrera, J.B.; Lewis, L.; Mehra, R.K. Detection and classification of intrusions and faults using sequences of system calls. *ACM SIGMOD Rec.* **2001**, *30*, 25–34. [CrossRef]
9. Hofmeyr, S.A.; Forrest, S.; Somayaji, A. Intrusion detection using sequences of system calls. *J. Comput. Secur.* **1998**, *6*, 151–180. [CrossRef]
10. Lane, T.; Brodley, C.E. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans. Inf. Syst. Secur. (TISSEC)* **1999**, *2*, 295–331. [CrossRef]
11. Warrender, C.; Forrest, S.; Pearlmutter, B. Detecting intrusions using system calls: Alternative data models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No. 99CB36344), Oakland, CA, USA, 14 May 1999; pp. 133–145.
12. Keogh, E.; Lonardi, S.; Ratanamahatana, C.A. Towards parameter-free data mining. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004; pp. 206–215.
13. Keogh, E.; Lonardi, S.; Ratanamahatana, C.A.; Wei, L.; Lee, S.H.; Handley, J. Compression-based data mining of sequential data. *Data Min. Knowl. Discov.* **2007**, *14*, 99–129. [CrossRef]
14. Keogh, E.; Keogh, L.; Handley, J.C. Compression-based data mining. In *Encyclopedia of Data Warehousing and Mining*, 2nd ed.; IGI Global: Pennsylvania, PA, USA, 2009; pp. 278–285.
15. Keogh, E.; Lonardi, S.; Chiu, B.C. 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, Edmonton, AB, Canada, 23–26 July 2002; pp. 550–556.
16. Keogh, E.; Lin, J.; Lee, S.H.; Van Herle, H. Finding the most unusual time series subsequence: Algorithms and applications. *Knowl. Inf. Syst.* **2007**, *11*, 1–27. [CrossRef]
17. Ferguson, T.S. *Mathematical Statistics: A decision Theoretic Approach*; Academic Press: Cambridge, MA, USA, 2014; Volume 1.
18. Siegmund, D.; Venkatraman, E. Using the generalized likelihood ratio statistic for sequential detection of a change-point. *Ann. Stat.* **1995**, 23, 255–271. [CrossRef]
19. Hirai, S.; Yamanishi, K. 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, Beijing, China, 12–16 August 2012; pp. 343–351.
20. Yamanishi, K.; Miyaguchi, K. Detecting gradual changes from data stream using MDL-change statistics. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 156–163.
21. Killick, R.; Fearnhead, P.; Eckley, I.A. Optimal detection of changepoints with a linear computational cost. *J. Am. Stat. Assoc.* **2012**, *107*, 1590–1598. [CrossRef]
22. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 44. [CrossRef]
23. Chernoff, H. Sequential design of experiments. *Ann. Math. Stat.* **1959**, *30*, 755–770. [CrossRef]
24. Basseville, M.; Nikiforov, I.V. *Detection of Abrupt Changes: Theory and Application*; Prentice Hall Englewood Cliffs: Hoboken, NJ, USA, 1993; Volume 104.
25. Veeravalli, V.V.; Banerjee, T. Quickest change detection. *Acad. Press Libr. Signal Process. Array Stat. Signal Process.* **2013**, *3*, 209–256.
26. Han, C.; Willett, P.; Chen, B.; Abraham, D. A detection optimal min-max test for transient signals. *Inf. Theory IEEE Trans.* **1998**, *44*, 866–869. [CrossRef]
27. Wang, Z.; Willett, P. A performance study of some transient detectors. *Signal Process. IEEE Trans.* **2000**, *48*, 2682–2685. [CrossRef]
28. Wang, Z.; Willett, P.K. All-purpose and plug-in power-law detectors for transient signals. *Signal Process. IEEE Trans.* **2001**, *49*, 2454–2466. [CrossRef]
29. Wang, Z.J.; Willett, P. A variable threshold page procedure for detection of transient signals. *IEEE Trans. Signal Process.* **2005**, *53*, 4397–4402. [CrossRef]
30. Hero, A.O. Geometric entropy minimization (GEM) for anomaly detection and localization. *NIPS* **2006**, *19*, 585–592.
31. Sricharan, K.; Hero, A. Efficient anomaly detection using bipartite k-nn graphs. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 478–486.
32. Sen, P.K. *Theory and Applications of Sequential Nonparametrics*; SIAM: Philadelphia, PA, USA, 1985.
33. Balsubramani, A.; Ramdas, A. Sequential Nonparametric Testing with the Law of the Iterated Logarithm. In Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, Jersey City, NJ, USA, 25–29 June 2016; AUAI Press: Arlington, VA, USA, 2016; pp. 42–51.
34. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly Detection for Discrete Sequences: A Survey. *Knowl. Data Eng. IEEE Trans.* **2012**, *24*, 823–839. [CrossRef]
35. Evans, S.; Barnett, B.; Bush, S.; Saulnier, G. Minimum description length principles for detection and classification of FTP exploits. In Proceedings of the Military Communications Conference, Monterey, CA, USA, 31 October–3 November 2004; Volume 1, pp. 473–479. [CrossRef]
36. Wang, N.; Han, J.; Fang, J. An Anomaly Detection Algorithm Based on Lossless Compression. In Proceedings of the 2012 IEEE 7th International Conference on Networking, Architecture and Storage (NAS), Xiamen, China, 28–30 June 2012; pp. 31–38. [CrossRef]
37. Lee, W.; Xiang, D. Information-theoretic measures for anomaly detection. In Proceedings of the 2001 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 14–16 May 2001; pp. 130–143. [CrossRef]
38. Paschalidis, I.; Smaragdakis, G. Spatio-Temporal Network Anomaly Detection by Assessing Deviations of Empirical Measures. *Netw. IEEE/ACM Trans.* **2009**, *17*, 685–697. [CrossRef]

39. Han, C.K.; Choi, H.K. Effective discovery of attacks using entropy of packet dynamics. *Netw. IEEE* **2009**, *23*, 4–12. [CrossRef]
40. Baliga, P.; Lin, T. Kolmogorov complexity based automata modeling for intrusion detection. In Proceedings of the 2005 IEEE International Conference on Granular Computing, Beijing, China, 25–27 July 2005; Volume 2, pp. 387–392. [CrossRef]
41. Shahriar, H.; Zulkernine, M. Information-Theoretic Detection of SQL Injection Attacks. In Proceedings of the 2012 IEEE 14th International Symposium on High-Assurance Systems Engineering (HASE), Omaha, NE, USA, 25–27 October 2012; pp. 40–47. [CrossRef]
42. Xiang, Y.; Li, K.; Zhou, W. Low-Rate DDoS Attacks Detection and Traceback by Using New Information Metrics. *Inf. Forensics Secur. IEEE Trans.* **2011**, *6*, 426–437. [CrossRef]
43. Pan, F.; Wang, W. Anomaly detection based-on the regularity of normal behaviors. In Proceedings of the 1st International Symposium on Systems and Control in Aerospace and Astronautics, Harbin, China, 19–21 January 2006. [CrossRef]
44. Eiland, E.; Liebrock, L. An application of information theory to intrusion detection. In Proceedings of the Fourth IEEE International Workshop on Information Assurance, London, UK , 13–14 April 2006. [CrossRef]
45. Li, M.; Chen, X.; Li, X.; Ma, B.; Vitanyi, P. The similarity metric. *Inf. Theory IEEE Trans.* **2004**, *50*, 3250–3264. [CrossRef]
46. Li, Y.; Nitinawarat, S.; Veeravalli, V.V. Universal outlier hypothesis testing. *IEEE Trans. Inf. Theory* **2014**, *60*, 4066–4082. [CrossRef]
47. Li, Y.; Nitinawarat, S.; Veeravalli, V.V. Universal outlier detection. In Proceedings of the Information Theory and Applications Workshop (ITA), San Diego, CA, USA, 10–15 February 2013; pp. 1–5.
48. Ziv, J.; Merhav, N. A measure of relative entropy between individual sequences with application to universal classification. *IEEE Trans. Inf. Theory* **1993**, *39*, 1270–1279. [CrossRef]
49. Chandola, V. Anomaly Detection for Symbolic Sequences and Time Series Data. Ph.D. Thesis, University of Minnesota, Minneapolis, MN, USA, 2009.
50. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; John Wiley & Sons: Hoboken, NJ, USA, 2005; Volume 589.
51. Wu, Q.; Shao, Z. Network anomaly detection using time series analysis. In Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking And Services-(icas-isns' 05), Papeete, France, 23–28 October 2005; p. 42.
52. Pincombe, B. Anomaly detection in time series of graphs using arma processes. *Asor Bull.* **2005**, *24*, 2.
53. Moayedi, H.Z.; Masnadi-Shirazi, M. Arima model for network traffic prediction and anomaly detection. In Proceedings of the 2008 International Symposium on Information Technology, Kuala Lumpur, Malaysia, 26–28 August 2008; Volume 4, pp. 1–6.
54. Ma, J.; Perkins, S. Online novelty detection on temporal sequences. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 613–618.
55. Knorn, F.; Leith, D.J. Adaptive kalman filtering for anomaly detection in software appliances. In Proceedings of the IEEE INFOCOM Workshops, Phoenix, AZ, USA, 13–18 April 2008; pp. 1–6.
56. Gusfield, D. Algorithms on stings, trees, and sequences: Computer science and computational biology. *Acm Sigact News* **1997**, *28*, 41–60. [CrossRef]
57. Thottan, M.; Ji, C. Anomaly detection in IP networks. *Signal Process. IEEE Trans.* **2003**, *51*, 2191–2204. [CrossRef]
58. Chakrabarti, S.; Sarawagi, S.; Dom, B. Mining surprising patterns using temporal description length. In Proceedings of the VLDB'98, 24rd International Conference on Very Large Data Bases, New York, NY, USA, 24–27 August 1998; pp. 606–617.
59. Akoglu, L.; Tong, H.; Koutra, D. Graph based anomaly detection and description: A survey. *Data Min. Knowl. Discov.* **2015**, *29*, 626–688. [CrossRef]
60. Ranshous, S.; Shen, S.; Koutra, D.; Harenberg, S.; Faloutsos, C.; Samatova, N.F. Anomaly detection in dynamic networks: A survey. *Wiley Interdiscip. Rev. Comput. Stat.* **2015**, *7*, 223–247. [CrossRef]
61. Yu, R.; Qiu, H.; Wen, Z.; Lin, C.; Liu, Y. A survey on social media anomaly detection. *ACM SIGKDD Explor. Newsl.* **2016**, *18*, 1–14. [CrossRef]
62. Aggarwal, C.C.; Philip, S.Y. An effective and efficient algorithm for high-dimensional outlier detection. *VLDB J.* **2005**, *14*, 211–221. [CrossRef]
63. Goldstein, M.; Dengel, A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In Proceedings of the KI-2012: Poster and Demo Track, Saarbrücken, Germany, 24–27 September 2012; 2012; pp. 59–63.
64. Foorthuis, R. SECODA: Segmentation-and combination-based detection of anomalies. In Proceedings of the 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 19–21 October 2017; pp. 755–764.
65. Foorthuis, R. The Impact of Discretization Method on the Detection of Six Types of Anomalies in Datasets. *arXiv* **2020**, arXiv:2008.12330.
66. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
67. Yang, Y.; Newsam, S. 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, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
68. Bradski, G.; Kaehler, A. *Learning OpenCV: Computer vision with the OpenCV Library*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
69. Tung, S. On lower and upper bounds of the difference between the arithmetic and the geometric mean. *Math. Comput.* **1975**, *29*, 834–836. [CrossRef]
70. Willems, F. The context-tree weighting method: Extensions. *Inf. Theory IEEE Trans.* **1998**, *44*, 792–798. [CrossRef]

71. Willems, F.M.J.; Shtarkov, Y.; Tjalkens, T. The context-tree weighting method: Basic properties. *Inf. Theory IEEE Trans.* **1995**, *41*, 653–664. [CrossRef]
72. Willems, F.; Shtarkov, Y.; Tjalkens, T. Reflections on "The Context Tree Weighting Method: Basic properties". *Newsl. IEEE Inf. Theory Soc.* **1997**, *47*.
73. Hoorfar, A.; Hassani, M. Inequalities on the Lambert W function and hyperpower function. *J. Inequal. Pure Appl. Math* **2008**, *9*, 5–9.
74. Lempel, A.; Ziv, J. On the Complexity of Finite Sequences. *Inf. Theory IEEE Trans.* **1976**, *22*, 75–81. [CrossRef]
75. Jacquet, P.; Szpankowski, W. Limiting Distribution of Lempel Ziv'78 Redundancy. In Proceedings of the 2011 IEEE International Symposium on Information Theory Proceedings (ISIT), St. Petersburg, Russia, 31 July–5 August 2011; pp. 1509–1513.
76. Yang, E.H.; Meng, J. Non-asymptotic equipartition properties for independent and identically distributed sources. In Proceedings of the 2012 Information Theory and Applications Workshop, San Diego, CA, USA, 5–10 February 2012; pp. 39–46.
77. Mackey, M.C.; Glass, L. Oscillation and chaos in physiological control systems. *Science* **1977**, *197*, 287–289. [CrossRef] [PubMed]
78. Gersho, A.; Gray, R.M. *Vector Quantization and Signal Compression*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 159.
79. Grzesiak, E.; Bent, B.; McClain, M.T.; Woods, C.W.; Tsalik, E.L.; Nicholson, B.P.; Veldman, T.; Burke, T.W.; Gardener, Z.; Bergstrom, E.; et al. Assessment of the Feasibility of Using Noninvasive Wearable Biometric Monitoring Sensors to Detect Influenza and the Common Cold Before Symptom Onset. *JAMA Netw. Open* **2021**, 4, e2128534. [CrossRef] [PubMed]
80. She, X.; Zhai, Y.; Henao, R.; Woods, C.; Chiu, C.; Ginsburg, G.S.; Song, P.X.; Hero, A.O. Adaptive multi-channel event segmentation and feature extraction for monitoring health outcomes. *IEEE Trans. Biomed. Eng.* **2020**, 68, 2377–2388. [CrossRef]