Extracting Discriminative Features for Identifying Abnormal Sequences in One-class Mode

Yin Song, Longbing Cao, Junfu Yin and Cheng Wang

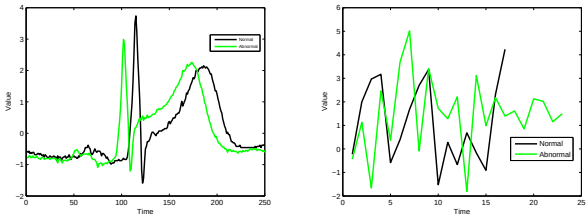
***Abstract*— This paper presents a novel framework for detecting abnormal sequences in an one-class setting (i.e., only normal data are available), which is applicable to various domains. Examples include intrusion detection, fault detection and speaker verification. Detecting abnormal sequences with only normal data presents several challenges for anomaly detection: the weak discrimination of normal and abnormal sequences; the unavailability of the abnormal data and other issues. Traditional model-based anomaly detection techniques can solve some of the above issues but with limited discrimination power (because of directly modeling the normal data). In order to enhance the discriminative power for anomaly detection, we turn to extracting discriminative features from the generative model based on the principle deducted from the corresponding theoretical analysis. Then a new anomaly detection framework is developed on top of that. The proposed approach firstly projects all the sequential data into a model-based equal length feature space (this is theoretically proven to have better discriminative power than the model itself) , and then adopts a classifier learned from the transformed data to detect anomalies. Experimental evaluation on both the synthetic and real-world data shows that our proposed approach outperforms several anomaly detection baseline algorithms for sequential data.**

I. Introduction

ANOMALY detection has traditionally been an important part of behavior analysis, whose aim is to find abnormal patterns in data that do not conform to expected (normal) behavior [1]. Most of the traditional anomaly detection techniques focus on static behavioral records or transactional data [2]. But in many real life scenarios, behaviors are dynamic and naturally organized as sequential data and the target of anomaly detection is collections of behaviors other than individual ones. One such example could be seen in intrusion detection for the operating system, i.e., to detect malicious programs (processes) from the normal execution processes. Each process (program) is denoted by its trace, which is a sequence of system calls used by that process from the beginning of its execution to the end. Table I shows three example programs in which normal and malicious ones are mixed. Each row records the sequential system calls (e.g., read and open) of one program. Another example could be found in detecting abnormal Electrocardiogram (ECG) signals. ECG signals record the dynamic behaviors of the heart over a period of time, which could be further utilized to characterize the heart's condition. Fig. 1(a) depicts two sampled ECG signals, one of which is from a healthy heart (i.e., normal) and the other is from an attacked heart (i.e., abnormal). From the above examples, we can intuitively find two things: firstly, these sequences are characterized by their dynamics; secondly, the normal and abnormal sequences are very similar by their appearance. For the purpose of detecting these abnormal sequential behaviors, we should consider the dynamic characteristics of sequential data, which is different from anomaly detection in static data. Another challenging issue is how to discriminate these abnormal dynamic behaviors from highly resemblant normal behaviors.

TABLE I: Some Sample Data of Operating System Call Traces [1].

open	read	mmap	mmap	open	read	$\cdots$
open	mmap	mmap	read	open	$\cdots$	$\cdots$
open	close	open	close	open	mmap	$\cdots$



(a) (b)

Fig. 1: (a) Some Sampled Signals from the ECG Data Set. (b) Some Sample Sequences from the Synthetic Data Set.

The above scenarios form a challenging issue, that is to detect abnormal behavioral sequences (which highly resemble normal behavioral sequences) in a set of sequences. To be more precise, the problem we will explore in this paper can be formally stated as follows:

Definition 1: Given a set of n training normal sequences, $\mathcal{X}^{tr}$, and a set of m test sequences $\mathcal{X}^{te}$, find a set of abnormal sequences $\mathcal{X}^{a} \subset \mathcal{X}^{te}$.

The key challenges of the above problem are listed in the following: Firstly, the sequences are quite dynamic, which is not intuitive to capture. Secondly, the abnormal sequences are usually highly similar to the normal ones in nature. This can be seen from Table I and Fig. 1(a). In addition, other related issues with anomaly detection for sequential data include variable lengths of sequences, and imbalance betweens normal and abnormal data (i.e., one-class mode in this paper).

Several techniques [3], [1] have been proposed to solve the problem of detecting abnormal sequences. Most of these techniques only consider some of the issues above and can

Yin Song, Longbing Cao, Junfu Yin and Cheng Wang are with the Advanced Analytics Institute (AAI), University of Technology, Sydney, Australia (email: {Yin.Song, Junfu.Yin}@student.uts.edu.au, {Longbing.Cao, Cheng.Wang}@uts.edu.au).

This work is supported in part by the Australian Research Council Discovery grant (DP130102691) and Linkage grants (LP120100566 and LP100200774).

be categorized into two types. One type is to degrade the problem to point (static) anomaly detection. Some techniques in this category treat a sequence as a vector of attributes assuming that the sequences are of equal length [4] and then point anomaly detection techniques are applied. This is problematic when the lengths of sequences are not equal. To avoid this problem, different similarity (or distance)-based [5] anomaly detection techniques have been proposed. However, the above approaches depend strongly on the definition of similarity (or distance) measure, which could be problematic when the data is very dynamic. For example, the behaviors of ECG signals are changing from time to time, following a stochastic nature. Thus, defining a proper and robust distance measure in this setting is difficult. To avoid this, another type of sequence anomaly detection techniques tries to model the sequences and thus is model-based. The model-based methods use statistical models to capture the dynamic characteristics of the sequences. Representative models, such as Hidden Markov Models (HMMs) [6], Finite State Automatons (FSAs) [7] and coupled HMMs (CHMMs) [8] have been studied in different application domains (e.g., operating system call data, network protocol data and financial data). The underlying assumption of these model-based algorithms is that normal sequences satisfy the normal model while abnormal ones do not. Although the model-based approaches are reasonable to some extent, we find that directly modeling the normal data has limited discriminative power in identifying abnormal sequences because abnormal sequences are highly similar to normal ones. This in turn could result in the degradation of the anomaly detection performance.

Hence, we propose a novel anomaly detection framework to deal with the issue of limited discriminative power in the traditional model-based approaches. The main contributions of this paper are listed as follows:

- Based on the analysis of Bayes error, we provide the theoretical principle of extracting discriminative features for one-class anomaly detection.
- A flexible three-phase implementation framework is proposed: *Phase 1* extracts discriminative features from the sequences based on the aforementioned theoretical feature extractor principle; *Phase 2* learns a discriminative classifier (e.g., SVM) on this new feature space; *Phase 3* applies the learned classifier to detect fraudulent sequences.

The remainder of this paper is organized as follows. Section II reviews the existing model-based anomaly detection and discusses its limitations, followed by theoretical analysis for enhancing the discriminative power for anomaly detection in Section III. Section IV proposes an implementation framework based on the theoretical analysis. After that, Section V and VI describe empirical results on both synthetic and real-world data sets. Section VII concludes this paper.

II. Model-based Anomaly Detection and Its Limitations

In this part, we briefly review the commonly-used model-based framework to handle one-class anomaly detection for sequential data [9], [6], [8] and point out its limitation from the theoretical perspective.

A. The Anomaly Detection Algorithm

The goal of sequence anomaly detection is to take an input sequence $\mathbf{x}$ and assign it to two discrete classes y where $y = 1, -1$ (1 denotes normal class and -1 denotes abnormal class). Generally speaking, the model-based framework detects anomaly by thresholding the likelihood

$$P_{\theta_1^*}(\mathbf{x}) < Th_0 \tag{1}$$

where $P_{\theta_1^*}(\mathbf{x}) = P(\mathbf{x}; \theta^* | y = 1)$ (and this form of notation has similar meanings in the rest of the paper), θ_1^* is the normal model parameters (and usually estimated as $\hat{\theta}_1$ from training data $\mathcal{X}^{tr}$) for normal class. The sample $\mathbf{x}$ satisfies Equation 1 is detected as an anomaly. The model-based algorithm consists of two stages: the first stage is to profile the normal sequence with a generative model $\hat{\theta}_1$ while the second stage is to detect abnormal sequences in the test data set according to Equation 1.

B. Limitations: Theoretical Analysis

As reviewed in the above, the one-class sequence anomaly detection is to predict discrete class labels (i.e., normal or abnormal), which is similar to the aim of classification problem. In fact, the difference between the problem considered in this paper and the classification one is the availability of training data. In this paper, only normal data is available for training and thus can be seen as a special case of classification problem, which is helpful to theoretical analysis.

For a standard classification problem, assuming we know the 'oracle' (i.e., true) parameters θ^* (θ_1^* denotes the parameters for the normal class and θ_{-1}^* denotes the parameters for the abnormal class) for generating the data, classifying an input $\mathbf{x}$ is to threshold the posterior probability $P(y = 1|\mathbf{x}; \theta^*)$ [10] with a threshold $\frac{1}{2}$, which is equivalent to the following oracle classifier (and the proof can be found in Appendix A):

$$P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta_{-1}^*}(\mathbf{x}) \tag{2}$$

The sample $\mathbf{x}$ satisfies Equation 2 is detected as an anomaly. Compared to Equation 1, we can see that the model-based anomaly detection algorithm does not consider the term $P_{\theta_{-1}^*}(\mathbf{x})$ for classification decision making and thus has less discriminative power for classification, which could harm the anomaly detection result. Here the Bayes error [10] is adopted to measure the performance of the anomaly detection algorithms. It is also an indicator of the discriminative power since good discrimination leads to good anomaly detection performance. Suppose the oracle classifier expressed as Equation 2 has the oracle Bayes error L^* for all $\mathbf{x} \in \mathcal{X}$, the performance of the model-based anomaly detection

algorithm expressed as Equation 1 could not achieve good approximation to L^* in general cases. To enhance the discriminative power for anomaly detection, we try to find another method whose classification performance could have a better approximation to L^*, which will be discussed in the following sections.

III. How to Enhance the Discriminative Power: Theoretical Analysis

The above section has pointed out the limitation of the model-based anomaly detection algorithm and our aim is to find a method to approximate the oracle Bayes error L^*. Inspired by [11], Section III-A first proposes a well-founded performance measure to theoretically evaluate the approximation, and then an approximation method of extracting proper features combined with a classifier is suggested in Section III-B.

A. Objective Function

It is straightforward to see that the oracle classifier Equation 2 has the most discriminative power for classifying normal and abnormal sequences which achieves the oracle Bayes error L^*. Thus, it is desirable that the theoretical Bayes error of the proposed anomaly detection algorithm should approach L^* as close as possible. Here we consider a linear classifier $\mathbf{w^T f}_{\hat{\theta}}(\mathbf{x}) + b$ combined with a feature extractor $f_{\hat{\theta}}(x)$ ($f_{\hat{\theta}}(x) : \mathcal{X} \to \mathbb{R}^D$ and $\mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$) to approximate the oracle classifier. The corresponding Bayes error is

$$R(f_{\hat{\theta}}) = \min_{\mathbf{w}\in\mathcal{S}, b\in\mathbb{R}} E_{x,y}\Phi[-y(\mathbf{w}^T f_{\hat{\theta}}(\mathbf{x}) + b)] \quad (3)$$

where $\mathcal{S} = \{\mathbf{w}|\mathbf{w} \in \mathbb{R}^D\}$, $\Phi[a]$ is the step function (which is 1 if $a > 0$ and 0 otherwise), and $E_{x,y}$ denotes the expectation with respect to the true distribution $p(\mathbf{x}, y|\theta^*)$. $R(f_{\hat{\theta}})$ is at least as large as the oracle Bayes error L^* and $R(f_{\hat{\theta}}) = L^*$ only if the linear classifier implements the same decision rule as the oracle classifier [12]. Usually $\mathbf{w}$ and b can be determined by a learning algorithm and we assume the optimal learning algorithm is used. When $\mathbf{w}$ and b are optimally chosen, the remaining part to determine is the feature extractor $f_{\hat{\theta}}(x)$ that minimize $R(f_{\hat{\theta}}) - L^*$, which describes how close the Bayes error to the oracle one.

Now it is natural to design a feature extractor that minimizes the objective function $R(f_{\hat{\theta}}) - L^*$. Direct optimization of this function is difficult because there exists a non differentiable function Φ. Alternatively, we turn to minimize its upper bound $2D(f_{\hat{\theta}})$, which generally has the following relationship with the objective function [10]:

$$R(f_{\hat{\theta}}) - L^* \leq 2D(f_{\hat{\theta}}). \quad (4)$$

where $D(f_{\hat{\theta}}) = \min_{\mathbf{w}\in\mathbb{R}^D, b\in\mathbb{R}} E_{\mathbf{x}}|F(\mathbf{w}^T f_{\hat{\theta}}(\mathbf{x}) + b) - P(y = 1 \mid \mathbf{x}; \theta^*)|$ and $F(t) = \frac{1}{(1+exp(-t))}$. Thus, $D(f_{\hat{\theta}})$ becomes an alternative object function to minimize whose minimization leads to the minimization of $R(f_{\hat{\theta}}) - L^*$ in terms of upper bounds.

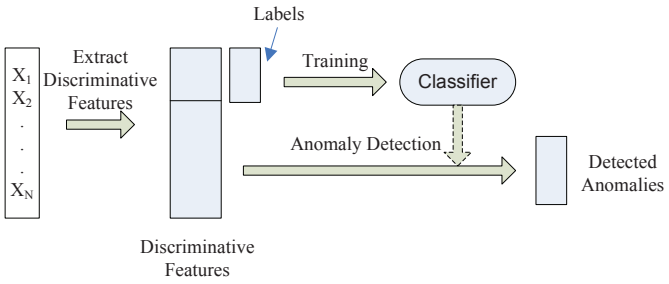


Fig. 2: The Flow Chart and Algorithm of the Proposed Framework

B. Proposed Feature Extractor

On the basis of the above object function, we further propose a feature extractor that achieves small $D(f_{\hat{\theta}})$. It is straightforward to see that a feature extractor $f_{\hat{\theta}}(\mathbf{x})$ satisfies

$$\mathbf{w}^T\mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b = F^{-1}(P(y = 1|\mathbf{x}; \theta^*)) \text{ for all } \mathbf{x} \in \mathcal{X} \quad (5)$$

with certain values of $\mathbf{w}$ and b, we have $D(f_{\hat{\theta}}) = 0$, which is the minimum point. However, since the oracle parameter θ^* is unknown, we cannot construct this optimal feature extractor $f_{\hat{\theta}}$ according to $F^{-1}(P(y = 1|\mathbf{x}; \theta^*))$. However, it can be approximated by its Taylor expansion at the point $\hat{\theta}_1$ estimated from the training data. The corresponding approximate optimal feature extractor is as follows:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta^*_{1_1}} g(\hat{\theta}_1), \cdots, \partial_{\theta^*_{1_p}} g(\hat{\theta}_1))^T \quad (6)$$

where $g(\theta^*_1) = \log P_{\theta^*_1}(\mathbf{x})$, $\partial_{\theta^*_{1_i}} g(\hat{\theta}_1)$ $(1 \leq i \leq p)$ is $g(\theta^*_1)$'s gradient with respect to θ^*_{1i} at point $\hat{\theta}_1$ and can be seen as a function of $\mathbf{x}$ since $\hat{\theta}_1$ is fixed. Thus the extracted feature is a set of functions of $\mathbf{x}$. The proof can be found in Appendix B. It is also notable that the theoretical performance of the proposed feature extractor with optimal classifier is better than that of the model-based algorithm and the proof can be found in Appendix C.

IV. Proposed Implementation Framework

Motivated by the theoretical analysis of enhancing the discriminative power for the model-based anomaly detection algorithm, we further propose an efficient implementation framework, called model-based discriminative feature (MDF) anomaly detection framework. A key challenge regarding implementation is to choose proper $\mathbf{w}$ of the classifier for anomaly detection, since the principle of feature extractor is already given. An overview of the MDF framework is shown in Fig. 2. More specifically, *Phase 1* is to extract the features on the basis of $f_{\hat{\theta}}$ in the form of Equation 6. Then in *Phase 2*, based on the extracted features, the corresponding optimal $\mathbf{w}$ is learned using a one-class support vector machine (SVM). Finally, the anomaly detection task is performed by the learned classifier produced in *Phase 3*. The following sections will describe the details of the three phases.

A. Phase 1: Feature Extraction

For the first phase, we need to choose a proper model to extract features based on it. In this paper, we assume that sequences could be well modeled by hidden Markov

Models (HMMs), because its expressive power of modeling real-world dynamic behavioral process, such as speech signal [13], biological sequence[14], gestures [15] and videos [16].

Here we first review the basic notions of HMMs and then give out the form of derivatives $\partial_{\theta_{1i}^*} g(\hat{\theta}_1)$ $(1 \leq i \leq p)$ used for feature extraction. Formally, a first-order HMM can be formally defined by:

- A set of Q possible hidden states denoted as $\mathcal{Q} = \{1, 2, \cdots, Q\}$, where $i (1 \leq i \leq Q)$ is a possible hidden state. The state at time t is denoted as q_t and $q_t \in \mathcal{Q}$.
- The hidden state transition matrix is $A = a_{ij}$, where $a_{ij} = P(q_{t+1} = j | q_t = i), 1 \leq i, j \leq Q$ is the probability for the transition from i to j.
- The observation vector $\mathbf{x}_t$ at time t is supposed to be governed by the corresponding conditional probability distribution $b_j(\mathbf{x}_t)$ $(1 \leq j \leq Q)$. When the observation vectors are discrete symbols, $b_j(\mathbf{x}_t)$ $(1 \leq j \leq Q)$ for each hidden state j is usually associated with the multinomial distribution as $b_j(\mathbf{x}_t) = \prod_{k=1}^{K} \mu_{jk}^{x_{tk}}$. Here we use the 1-of-K scheme (i.e., $\mathbf{x}_t = [x_{t1}, \cdots, x_{tK}]^T$, subects to $\sum_k x_{tk} = 1$) to represent the discrete observation as a K-dimensional vector where K is the number of vocabulary for the discrete symbols. When the observation vectors are continuous, $\mathbf{x}_t$ (with hidden state j) is usually assumed to subject to a mixture of Gaussian distributions $\sum_{k=1}^{K} c_{jk} \mathcal{N}(\mathbf{x}_t | \mu_{jk}, \Sigma_{jk})$, where c_{jk} is the mixture coefficient for the k^{th} Gaussian mixture in the state j, $\mathcal{N}$ is a Gaussian distribution density with the mean vector μ_{jk} and the covariance matrix Σ_{jk}.
- The initial state probability distribution $\pi = (\pi_1, \pi_2, \cdots, \pi_Q)$, where $\pi_i = P(q_1 = i), 1 \leq i \leq Q$.

Thus, an HMM can be denoted as $\theta = \{A, B, \pi\}$. Let $\mathbf{x}$ be an observation sequence, the parameters of an HMM are approximately learned using the Baum-Welch algorithm [17] given a set of sequences $\mathcal{X}^{tr}$. On the other hand, the partial derivatives of $g(\theta_1^*)$ at the point of $\hat{\theta}_\mathbf{1} = \{\hat{A}, \hat{B}, \hat{\pi}\}$ can be calculated by using $\hat{\xi}_t$ and $\hat{\gamma}_t$, which can be obtained by the forward-backward algorithm [13]. Specifically, $\hat{\xi}_t(i, j)$ is the probability of being in state i at time t and state j at time $t + 1$ given the model $\hat{\theta}_\mathbf{1}$ and the observation sequence $\mathbf{x}$, which is $\hat{\xi}_t(i, j) = P(q_t = i, q_{t+1} = j | \mathbf{x}; \hat{\theta}_\mathbf{1})$. For discrete observations, $\hat{\gamma}_t(j)$ is the probability of being in state j at time t, which is $\hat{\gamma}_t(j) = P(q_t = j | \mathbf{x}; \hat{\theta}_\mathbf{1})$; for continuous observations, $\hat{\gamma}_t(j, k)$ is the probability of being in state j at time t with the k^{th} Gaussian mixture component accounting for $\mathbf{x}_t$, which is $\hat{\gamma}_t(j, k) = P(q_t = j, M_{jt} = k | \mathbf{x}; \hat{\theta}_\mathbf{1})$, where M_{jt} is a random variable indicating the mixture component at time t in state j. Then partial derivatives of $g(\theta_1^*)$ with respect to the parameters θ_1^* at a point $\hat{\theta}_\mathbf{1}$ (estimated from the training data) are as following [18]:

$$\partial_{a_{ij}^*} g(\hat{\theta}_1) = \sum_{t=1}^{T-1} \frac{\hat{\xi}_t(i, j)}{\hat{a}_{ij}} \tag{7}$$

for discrete observations:

$$\begin{aligned} \partial_{\pi_i^*} g(\hat{\theta}_1) &= \frac{\hat{\gamma}_t(i)}{\hat{\pi}_i} \\ \partial_{\mu_{jk}^*} g(\hat{\theta}_1) &= \frac{\sum_{t=1}^{T} \hat{\gamma}_t(j) x_{tk}}{\hat{\mu}_{jk}} \end{aligned} \tag{8}$$

for continuous observations:

$$\begin{aligned} \partial_{\pi_i^*} g(\hat{\theta}_1) &= \frac{\hat{\gamma}_t(i, 1)}{\hat{\pi}_i} \\ \partial_{c_{jk}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \frac{\hat{\gamma}_t(j, k)}{\hat{c}_{ij}} \\ \partial_{\mu_{ij}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \hat{\gamma}_t(j, k) [\hat{\Sigma}_{jk}^{-1}]^T (\mathbf{x}_t - \hat{\mu}_{jk}) \\ \partial_{\hat{\Sigma}_{jk}^*} g(\hat{\theta}_1) &= \sum_{t=1}^{T} \frac{\hat{\gamma}_t(j, k)}{2} [G - vec(\hat{\Sigma}_{jk}^{-1})] \end{aligned} \tag{9}$$

where $vec(F) = [F_{11}, F_{12}, \cdots, F_{M1}, F_{MN}]^T$ when F is a matrix of size $M \times N$. $G = [(\mathbf{x}_t - \hat{\mu}_{jk})^T \Sigma_{jk}^{-1} \otimes (\mathbf{x}_t - \hat{\mu}_{jk})^T \Sigma_{jk}^{-1}]^T$ and $\otimes$ denotes the kronecker product. Then the algorithm for the feature extractor can be summarized in Algorithm 1: step 1 estimates parameters $\hat{\theta}_\mathbf{1}$ of the HMM from the training data; then step 2-9 extract the discriminative feature using Equation 7-9 for each sequence $\mathbf{x} \in \mathcal{X}^{tr} \cup \mathcal{X}^{te}$.

Algorithm 1 The Proposed Feature Extractor

Input: A Training set $\mathcal{X}^{tr}$, A Testing set $\mathcal{X}^{te}$,
Output: The transformed features $\mathcal{S}$.

1: Given $\mathcal{X}^{tr}$ train an HMM $\hat{\theta}_\mathbf{1}$.
2: **for all** $\mathbf{x} \in \mathcal{X}^{tr} \cup \mathcal{X}^{te}$ **do**
3: Given $\hat{\theta}_\mathbf{1}$, construct the corresponding discriminative features
4: **if** $\mathbf{x}$ is discrete **then**
5: Construct features according to Equation 7 and 8 as:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) = (\partial_{a_{11}^*} g(\hat{\theta}_1), \cdots, \partial_{\pi_1^*} g(\hat{\theta}_1), \cdots, \partial_{\mu_{11}^*} g(\hat{\theta}_1), \cdots)^T$$

6: **else**
7: Construct features according to Equation 7 and 9 as:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) = (\partial_{a_{11}^*} g(\hat{\theta}_1), \cdots, \partial_{\pi_1^*} g(\hat{\theta}_1), \cdots, \partial_{c_{11}^*} g(\hat{\theta}_1), \cdots, \partial_{\mu_{11}^*} g(\hat{\theta}_1), \cdots, \partial_{\sigma_{11}^*} g(\hat{\theta}_1), \cdots)^T$$

8: **end if**
9: $\mathbf{s} \rightarrow \mathcal{S}$
10: **end for**

B. Phase 2: Learning of the Optimal Linear Classifier

This phase tries to construct a linear classifier with the optimal $\mathbf{w}$, one-class SVM has become a natural choice, since it is linear classifier and only the normal sequences are provided for training. Suppose there is a training data set $\mathcal{S}^{tr}$ consists of m training sequences $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(m)}$, the learning objective function based on the maximum margin theory is [19]:

$$\min_{\mathbf{w}, \rho, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{1}{\nu m} \sum_i \xi_i - \rho, \tag{10}$$

$$\text{subject to } \mathbf{w}\Phi(\mathbf{x}^{(i)}) \leq \rho - \xi_i, \xi_i \geq 0, 1 \leq i \leq m. \tag{11}$$

TABLE II: Parameters of the HMMs Generating the Normal and Abnormal Sequences

	A	B	π
θ_1	$\begin{pmatrix}0.6 & 0.4\\ 0.4 & 0.6\end{pmatrix}$	$(\mathcal{N}(0,1),\mathcal{N}(3,1))$	$(0.5, 0.5)$
θ_{-1}	$\begin{pmatrix}0.4 & 0.6\\ 0.4 & 0.6\end{pmatrix}$	$(\mathcal{N}(0,1),\mathcal{N}(3,1))$	$(0.5, 0.5)$

Then, the estimated optimal $\mathbf{w}^*$ is obtained using α (which maximize Equation 10) as below:

$$\mathbf{w}^* = \sum_i \alpha_i \Phi(\mathbf{x}^{(i)}). \tag{12}$$

where $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \Phi(\mathbf{x}^{(i)})\Phi(\mathbf{x}^{(j)})$ is a kernel function and the $\mathbf{w}^*$ becomes the output classifier C.

C. Phase 3: Anomaly Detection

This phase is straightforward, for any sequence $\mathbf{x} \in \mathcal{S}^{te}$, apply the learned classifier C (i.e., $\mathbf{w}^{*\mathbf{T}}\mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b$) and Th_2 to detect anomaly. That is, if $\mathbf{w}^{*\mathbf{T}}\mathbf{f}_{\hat{\theta}}(\mathbf{x}) + b < Th_2$, $\mathbf{x}$ is detected as anomaly and put into the anomaly set $\mathcal{S}^a$, which is the output.

V. Experimental Settings

A. Data Sets

The details of both synthetic and real-world data sets are reported in this section. The synthetic data is used to illustrate the performance of the proposed algorithm without considering the influence of the approximate modeling. This is because all the synthetic data are sampled from generative HMMs and thus can be reasonably modeled as HMMs. In addition, we also use a variety of real-world data sets extracted from different application domains when the behavioral sequences can be approximately modeled as HMMs.

1) The Synthetic Data: Here we consider a toy example to test the performance of our proposed algorithm. We assume that normal and abnormal sequences are generated from two 2-state Gaussian HMMs (θ_1, θ_{-1}) specified in Table II respectively ('1' is the label for normal class and '-1' is the label for abnormal class).

Since the two models generating the sequences are very similar (and only have a slight difference in A), the generated sequences are very similar and quite difficult to differentiate by their appearance. Fig. 1(b) shows two sample sequences from the synthetic data. As can be seen from the chart, these sequences are quite stochastic and how to distinguish them directly is unclear. Thus, this synthetic data set provides a very challenging scenario for one-class mode sequence anomaly detection, because the abnormal sequences can only be differentiated from the normal sequences by their dynamical characteristics that are different in the model generating them. In other words, the abnormal sequences are very similar to the normal sequences. Thus, it is suitable for testing the discriminative power of our proposed framework. The length of each individual sequence is obtained by sampling a uniform pdf in the range of $[\mu_L(1-V/100)\ \mu_L(1+V/100)]$, where μ_L is the sequence's mean length and V is a parameter that refers to as the percentage of variation in the length ($V = 40$ in this paper). By doing so, we hope to examine the influence of sequence length on the anomaly detection performance. All the given results are averaged over 50 randomly generated data sets.

2) The Real-world Data: To evaluate the performance of the proposed algorithm in real world, 5 publicly available data sets are used. From the perspective of data types, these data sets can be grouped into two categories: discrete sequences and multi-(uni-)variate time series. From the perspective of data characteristics, the data sets are from different domains of intrusion detection (ID), fault detection (FD), electrocardiogram (ECG) signals, character trajectory (CT) records and Japanese Vowels (JV) speech. The details of the real-world data sets used are given in the following:

a) ID Data: This data set[1] were collected by the University of New Mexico to evaluate the performance of intrusion detection for system calls. The normal sequences consist of sequence of system calls generated in an operating system during the normal operation of a computer program, such as sendmail, ftp, lpr etc. The anomalous sequences consist of sequence of system calls generated when the program was run in an abnormal mode, corresponding to the operation of a hacked computer. A subset of data sets available in the repository is used here, which was processed by the same process mentioned in [20].

b) Fault Detection Data: This repository[2] is the basic security module (BSM) audit data, collected from a victim Solaris machine, in the DARPA Lincoln Labs 1998 network simulation data sets. The data is similar to the intrusion detection data described above.

c) Electrocardiogram (ECG) Data: This data set[3] corresponds to an ECG recording for one subject suffering with a particular heart condition. The ECG recording was segmented into short sequences of equal lengths. Sequences that contain any annotation of a heart condition are added to the anomalous set and the remaining sequences form the normal set.

d) Character Trajectory: This data set[4] consists of trajectories captured by a digitizing tablet when writing 20 different characters and each sample is a 3-dimensional time series differentiated and smoothed using a Gaussian kernel. In experiments, we use the sequences of one character as the normal set and use the samples of another character as the abnormal set, giving a total of 19 experiments (each experiment was repeated 10 times).

e) Japanese Vowels: The data set[5] collects several utterances of nine male speakers producing two Japanese vowels /ae/ successively. 12 dimension linear predictive coding (LPC) cepstrum coefficients have been extracted from each utterance, which forms a 12-dimension time series. In experiments, we use the sequences of one speaker as the normal set and use the samples of another speaker as

[1] Available at http://www.cs.unm.edu/∼immsec/systemcalls.htm.
[2] Available at http://www.ll.mit.edu/mission/communications/ist/.
[3] Available at http://www.physionet.org/physiobank/database/edb/.
[4] Available at http://archive.ics.uci.edu/ml/datasets/Character+Trajectories.
[5] Available at http://archive.ics.uci.edu/ml/datasets/Japanese+Vowels.

TABLE III: The Details of the Real Data Sets

Dataset	ID	FD	ECG	CT	JV
D	discrete	discrete	1	3	12
μ_L	839	143	250	166	16
$\lvert\mathcal{X}_N\rvert$	2030	2000	500	186	30
$\lvert\mathcal{X}_A\rvert$	130	67	50	119-171	30
$\lvert\mathcal{X}_{tr}\rvert$	1030	1000	500	136	10
$\lvert\mathcal{X}_{te}\rvert$	1050	1050	550	60	30

the abnormal set, giving a total of 8 experiments (each experiment was repeated 10 times).

Table III summarizes the data sets for experimental evaluation, where D is the dimension of each observation in the sequences, μ_L is the averaged length of the sequences and $|\mathcal{X}_i|$ $(i \in N, A, tr, te)$ is the number of sequences. For each data set, we have done repetitive experiments and report the averaged results of 10 times at least. The general methodology to create the data sets is as the following [20]: For each data set, a normal data set, $\mathcal{X}^N$, and an anomalous data set $\mathcal{X}^A$ are created. A training data set $\mathcal{X}^{tr}$ is created by randomly sampling a fixed number of sequences from $\mathcal{X}^N$. A test data set $\mathcal{X}^{te}$ is created by randomly sampling a fixed number of normal sequences from $\mathcal{X}^N - X^{tr}$ and a fixed number of anomalous sequences from $\mathcal{X}^A$.

B. Comparative Algorithms

We compare two variants of our proposed MDF framework (using linear and Gaussian radial basis SVM as the classifiers in phase 2) with the model-based algorithm, and four baseline methods without learning as following:

- MDF with linear SVM (MDF-SVM), which means a linear SVM is applied as the classifier in phase 2 of the MDF framework.
- MDF with Gaussian radial basis SVM (MDF-SVMrb), which means a non-linear SVM is applied as the classifier in phase 2 of the MDF framework.
- The Model-based Algorithm (use HMM as the model, as described in Section II-A).
- MDF with k-nearest neighbor classifier (MDF-kNN), which means a lazy classifier kNN is applied directly after phase 1 of the MDF framework without phase 2. In particular, we set $k = 4$, which is suggested by [20].
- Oracle Model (ORACLE). This baseline method uses the true model information of both the normal and the abnormal sequences. The classifier is constructed using the Bayes Rule. In particular, for a given sequence X_i, $P(y = 1|X_i; \theta_1, \theta_{-1})$ is calculated. If it is lower than a predefined threshold Th_0 then X_i is detected as anomaly.
- Semi-Oracle Model (Semi-ORACLE). This baseline method uses the true model information of only the normal sequences. The other setting is similar to the ORACLE model.
- Random Model (RANDOM). As indicated by the name, this model predicts the class label for each sequence randomly.

C. Performance Measures

To evaluate the performance of the above anomaly detection algorithms, we choose the area under receiver operating characteristic curve (AUC) [21] and a higher AUC usually means a better classification performance. The reason for this choice is the anomaly detection problem in this paper can be treated as a special case of a binary classification problem, and the AUC is widely accepted for evaluating the classification results summarizing the performance at various threshold settings.

VI. Experimental Results

A. Synthetic Data

Fig. 3(a) shows the results of the performance comparison of different anomaly detection techniques against different numbers of training sequences. It can be seen that, the number of training sequences does not have significant impact on the performance of the algorithms. This may be because of the sequences are generated by simple synthetic models and can be modeled by the HMMs using relatively small samples. Fig. 3(b) shows the results of the performance comparison of different anomaly detection techniques against different mean sequence lengths. As shown in the picture, the algorithms tend to have better performance when the length of sequences increases. This conforms to our intuition that longer sequences have clearer dynamic characteristics to capture, which is very helpful to further anomaly detection. Fig. 3(c) shows the results of the performance comparison of different anomaly detection techniques against different number hidden states Q of the HMMs. As can be seen from the chart, the performance of MDF-SVM, MDF-SVMrb and MDF-kNN decreases when the model structure varies. A possible explanation is that improper model structures may generate redundant dimensions in the extracted feature space and degrade the anomaly detection result. Fig. 3(d) shows the results of the performance comparison of different anomaly detection techniques against different ratios of the normal and abnormal sequences in a testing data set. It can be clearly seen from the picture that the ratio of the normal and abnormal sequences has little impact on the anomaly detection performance.

To sum up, the proposed MDF-SVM and MDF-SVMrb have the best result (close to ORACLE) consistently in most of different settings, which proves the stability of our proposed framework. This is because the proposed feature extractor could capture enough discriminative information to classify the normal and abnormal data and thus different settings have little impact on the anomaly detection performance. It is also noted that MDF-SVM and MDF-SVMrb generally outperforms MDF-kNN in most cases, which may benefit from their learning process in phase 2 of the framework compared to MDF-kNN. Thus, they are expected to have better performance in the real-world data sets, whose results will be reported in the following.

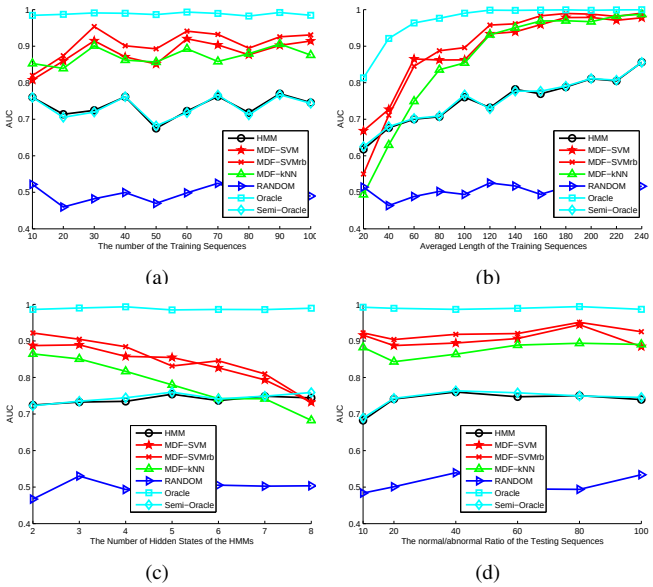


Fig. 3: The Experimental Results from the Synthetic Data Sets.

TABLE IV: The Experimental Results of the Real Data Sets

Dataset	Q	HMM	MDF-SVM	MDF-SVMrb	MDF-kNN	RANDOM
ID	2	0.94 ± 0.00	0.18 ± 0.18	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.51 ± 0.04
	3	0.94 ± 0.00	0.15 ± 0.09	$\mathbf{0.99 \pm 0.01}$	$\mathbf{0.99 \pm 0.00}$	0.48 ± 0.02
	4	0.94 ± 0.00	0.18 ± 0.18	$\mathbf{0.99 \pm 0.00}$	$\mathbf{0.99 \pm 0.00}$	0.51 ± 0.04
FD	2	0.39 ± 0.00	0.53 ± 0.2	$\mathbf{0.91 \pm 0.01}$	$\mathbf{0.91 \pm 0.00}$	0.50 ± 0.06
	3	0.39 ± 0.00	0.4 ± 0.12	$\mathbf{0.92 \pm 0.01}$	$\mathbf{0.92 \pm 0.00}$	0.51 ± 0.05
	4	0.39 ± 0.00	0.58 ± 0.13	$\mathbf{0.93 \pm 0.01}$	0.91 ± 0.00	0.50 ± 0.05
ECG	2	0.27 ± 0.00	$\mathbf{0.67 \pm 0.00}$	$\mathbf{0.67 \pm 0.00}$	0.61 ± 0.00	0.49 ± 0.04
	3	0.28 ± 0.00	$\mathbf{0.64 \pm 0.02}$	$\mathbf{0.64 \pm 0.02}$	0.61 ± 0.01	0.50 ± 0.04
	4	0.28 ± 0.00	$\mathbf{0.65 \pm 0.00}$	$\mathbf{0.65 \pm 0.00}$	0.61 ± 0.00	0.50 ± 0.05
CT	2	0.82 ± 0.2	0.71 ± 0.33	$\mathbf{0.96 \pm 0.04}$	$\mathbf{0.96 \pm 0.04}$	0.50 ± 0.10
	3	0.91 ± 0.1	0.75 ± 0.30	$\mathbf{0.97 \pm 0.03}$	$\mathbf{0.97 \pm 0.03}$	0.52 ± 0.10
	4	0.94 ± 0.07	0.77 ± 0.28	$\mathbf{0.98 \pm 0.06}$	$\mathbf{0.98 \pm 0.03}$	0.51 ± 0.10
JV	2	0.94 ± 0.07	0.95 ± 0.05	$\mathbf{0.96 \pm 0.06}$	0.94 ± 0.06	0.52 ± 0.13
	3	0.92 ± 0.07	$\mathbf{0.95 \pm 0.06}$	$\mathbf{0.95 \pm 0.06}$	$\mathbf{0.95 \pm 0.06}$	0.50 ± 0.15
	4	0.94 ± 0.06	0.95 ± 0.05	$\mathbf{0.96 \pm 0.05}$	0.95 ± 0.04	0.50 ± 0.14

B. Real-world Data

Table IV shows experimental results (averaged AUC value of at least 10 repetitive experiments) on the five real-world data sets, with the comparison of five algorithms. In the table, Q denotes the number of hidden states of the HMMs and the ORACLE and Semi-ORACLE algorithms are excluded since we do not know the true parameters of the model in real-world data sets. All in all, the MDF-SVMrb noticeably outperforms the rest of the alternatives. This is because the MDF-SVMrb not only extracts discriminative features but also learns a non-linear decision boundary in the extracted feature space to detect the anomalies, while others may fail to do so. MDF-SVM works very well on some data sets because the normal and abnormal sequences may be linearly seperatable in the MDF space under these cases. A remarkable fact is that the proposed algorithms do not suffer a severe performance loss as the number of hidden states increases. This is because the true models of the data are more complex and our models are relatively simple, which give proper approximations to the true models with no significant difference. This indicates the robustness of the algorithms when the true model is much more complicated. It is also worth to note that the proposed MDF-SVM and MDF-SVMrb generally perform better when the averaged length of the sequences increase, which agrees with the observation from the results obtained with synthetic data.

In addition, the computational cost of the proposed framework mainly spends on the feature extractor stage and it scales to $O(Q^2TN)$, where Q is the number of hidden states, T is the averaged length of the sequences and N is the number of sequences. Thus, the computational time of the MDF-SVM, MDF-SVMrb and MDF-kNN is very similar but a little higher than the HMM and RANDOM algorithms (the proposed framework, however, has much better anomaly detection performance). This is proved empirically in the experiments and we do not report the details here due to the space limit.

VII. Conclusions

This paper examines a challenging issue of detecting abnormal sequences in an one-class setting and presents a reasonable MDF framework by theoretically analyzing the nature of the problem. To be more specific, the proposed framework is composed of three phases: the generative model-based feature extractor phase, the optimal classifier training phase and the anomaly detection phase. Theoretical analysis has demonstrated that the proposed method leads to a better approximation to the oracle Bayes error (i.e., the anomaly detection performance in this paper). To evaluate the superiority of our proposed framework, several experiments have been conducted on synthetic data sets. The empirical results show that the proposed framework generally outperforms the other comparative schemes. We also explore a wide range of real-world problems, such as speaker verification and ECG signal detection (i.e., detecting hearts with problematic conditions) and the corresponding experimental results show the effectiveness of our proposed framework.

The problem of sequence anomaly detection considered in this paper is inherently in one-class mode (i.e., only the normal data is available for training). However, in many real-world scenarios, it is unrealistic to obtain data that ideally contains only normal instances. In these situations, the anomaly detection techniques need to be operated in a mixed setting (i.e., the training data contains both normal and anomalous sequences without labels, under the assumption that anomalous sequences are very rare). The extension to a mixed mode is a possible future research direction.

Appendices

A. Appendix A

Lemma 1: $P(y=1|\mathbf{x};\theta^*) < \frac{1}{2}$ is equivalent to $P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta_{-1}^*}(\mathbf{x})$.

Proof: According to Bayes' theorem:

$$P(y=1|\mathbf{x},\theta^*) = \frac{P_{\theta_1^*}(\mathbf{x})P(y=1)}{\sum_y P_{\theta_y^*}(\mathbf{x})P(y)} < \frac{1}{2} \tag{13}$$

After proper transformation, the above formulation becomes:

$$\frac{P_{\theta_1^*}(\mathbf{x})}{P_{\theta_{-1}^*}(\mathbf{x})} < \frac{P(y=-1)}{P(y=1)} = Th_1 \tag{14}$$

$$P_{\theta_1^*}(\mathbf{x}) < Th_1 \cdot P_{\theta^*_{-1}}(\mathbf{x}) \quad (15)$$

B. Appendix B

Lemma 2: The approximate optimal feature extractor $f_{\hat{\theta}}(\mathbf{x})$ with approximate oracle Bayes error L^* is given by:

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\theta^*_{1p}} g(\hat{\theta}_1))^T \quad (16)$$

Proof: Let us define $v(\theta^*) = F^{-1}(P(y = 1|\mathbf{x}; \theta^*)) = \log(P_{\theta_1^*}(\mathbf{x})) - \log(P_{\theta^*_{-1}}(\mathbf{x})) = g(\theta_1^*) - g(\theta^*_{-1})$, then By Taylor expansion around the estimated $\hat{\theta}$ up to the first order, we can approximate $v(\theta^*)$ as

$$\begin{aligned} v(\theta^*) &\approx v(\hat{\theta}) + \sum_{i=1}^{p} \partial_{\theta^*_{1i}} v(\hat{\theta}_{\mathbf{1}})(\theta^*_{1i} - \hat{\theta}_{1i}) \\ &\quad + \sum_{j=1}^{p} \partial_{\theta^*_{-1j}} v(\hat{\theta}_{\mathbf{-1}})(\theta^*_{-1j} - \hat{\theta}_{-1j}) \quad (17) \end{aligned}$$

where $\partial_{\theta^*_{ki}} v = \frac{\partial v}{\partial \theta^*_{ki}}$ and $\partial_{\theta^*_{ki}} v(\hat{\theta}_k)$ denotes v's derivative at the point $\hat{\theta}_k$ ($k \in \{1, -1\}$ and $1 \leq i \leq p$).

We use $\hat{\theta}_{\mathbf{1}}$ to approximate $\hat{\theta}_{\mathbf{-1}}$. This is reasonable because the abnormal sequences are highly similar to the normal ones (i.e., $\theta^*_{\mathbf{1}} \approx \theta^*_{\mathbf{-1}}$). Then Equation 17 becomes:

$$v(\theta^*) \approx \sum_{i=1}^{p} \partial_{\theta^*_{1i}} g(\hat{\theta}_1)(\theta^*_{1i} - \theta^*_{-1i}) \quad (18)$$

Consequently, by setting

$$\mathbf{f}_{\hat{\theta}}(\mathbf{x}) := (\partial_{\theta^*_{11}} g(\hat{\theta}_1), \cdots, \partial_{\theta^*_{1p}} g(\hat{\theta}_1))^T \quad (19)$$

and

$$\mathbf{w} := \mathbf{w}^* = (\theta^*_{11} - \theta^*_{-11}, \cdots, \theta^*_{1p} - \theta^*_{-1p})^T, b = 0. \quad (20)$$

the proposed feature extractor with the optimal classifer achieves a reasonable small $D(f_{\hat{\theta}}) \approx 0$ for the upper bound of classification error difference.

C. Appendix C

In this section, we theoretically compare the proposed feature extractor with the model-based anomaly detection in terms of approximation to the oracle Bayes error. $P(y = 1|\mathbf{x}; \theta)$ is assumed to $\in (0, 1)$[6] and $\nabla_\theta P(y = 1|\mathbf{x}; \theta))$ and $\nabla^2_\theta P(y = 1|\mathbf{x}; \theta))$ are assumed to be bounded, where $\nabla_\theta f = (\partial_{\theta_1} f, \cdots, \partial_{\theta_p} f)^T$ and the (i, j)th element of ∇^2_θ is $\frac{\partial^2 f}{\partial \theta_i \partial \theta_j}$. Then we have the upper bound of classification error difference between the model-based algorithm and the oracle classifier[7] is:

$$D(\hat{\theta}) = E_x |P(y = 1|\mathbf{x}; \hat{\theta}) - P(y = 1|\mathbf{x}; \theta^*)|. \quad (21)$$

Define $\mathbf{\Delta}\theta = \theta^* - \hat{\theta}$. By Taylor expansion around $\hat{\theta}$, we have

$$\begin{aligned} D(\hat{\theta}) &\approx Ex|(\mathbf{\Delta}\theta)^T \nabla_\theta P(y = 1|\mathbf{x}, \theta^*) \\ &+ \frac{1}{2}(\mathbf{\Delta}\theta)^T \nabla^2_\theta P(y = 1|\mathbf{x}, \theta_{\mathbf{0}})(\mathbf{\Delta}\theta)| \\ &= O(\|\mathbf{\Delta}\theta\|). \quad (22) \end{aligned}$$

[6]To prevent $|v(\theta)|$ from going to infinity.

[7]Here for simplicity, we use $P(y = 1|\mathbf{x}; \hat{\theta})$ to replace $P(\mathbf{x}|y = 1; \hat{\theta})$, where $P(\mathbf{x}|y = 1; \hat{\theta}_{-1})$ is estimated as a constant.

By contrast, when the proposed feature extractor is used,

$$D(f_{\hat{\theta}}) = E_{\mathbf{x}} |F((\mathbf{w}^*)^T f_{\hat{\theta}}(\mathbf{x})) - P_{\theta^*}(y = 1|\mathbf{x})|, \quad (23)$$

where $\mathbf{w}^*$ is defined as in Equation 20. Since F is Lipschitz continuous, there is a finite positive constant M such that $|F(a) - F(b)| \leq M|a - b|$ [11]. Thus,

$$\begin{aligned} D(f_{\hat{\theta}}) &\leq M E_{\mathbf{x}} |(\mathbf{w}^*)^T f_{\hat{\theta}}(\mathbf{x}) - F^{-1}(P_{\theta^*}(y = 1|\mathbf{x}))| \\ &= O(\|\mathbf{\Delta}\theta\|^2). \quad (24) \end{aligned}$$

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[2] V. Barnett and T. Lewis, *Outliers in statistical data*. Wiley Chichester, 1994.

[3] S. Budalakoti, A. Srivastava, and M. Otey, "Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 1, pp. 101–113, 2009.

[4] R. Blender, K. Fraedrich, and F. Lunkeit, "Identification of cyclone-track regimes in the north atlantic," *Quarterly Journal of the Royal Meteorological Society*, vol. 123, no. 539, pp. 727–741, 1997.

[5] S. Budalakoti, A. Srivastava, R. Akella, and E. Turkov, "Anomaly detection in large sets of high-dimensional symbol sequences," *NASA Ames Research Center, Tech. Rep. NASA TM-2006-214553*, 2006.

[6] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pp. 133–145.

[7] R. Sekar, M. Bendre, D. Dhurjati, and P. Bollineni, "A fast automaton-based method for detecting anomalous program behaviors." IEEE, pp. 144–155, security and Privacy, 2001. S & P 2001. Proceedings. 2001 IEEE Symposium on.

[8] L. Cao, Y. Ou, P. Yu, and G. Wei, "Detecting abnormal coupled sequences and sequence changes in group-based manipulative trading behaviors," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 85–94.

[9] S. Joshi and V. Phoha, "Investigating hidden markov models capabilities in anomaly detection." ACM, pp. 98–103, proceedings of the 43rd annual Southeast regional conference-Volume 1.

[10] L. Devroye, L. Gyorfi, and G. Lugosi, *A probabilistic theory of pattern recognition*. Springer Verlag, 1996, vol. 31.

[11] K. Tsuda, M. Kawanabe, G. Ratsch, S. Sonnenburg, and K. Muller, "A new discriminative kernel from probabilistic models," *Neural Computation*, vol. 14, no. 10, pp. 2397–2414, 2002.

[12] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Pr, 1990.

[13] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Readings in speech recognition*, vol. 53, no. 3, pp. 267–296, 1990.

[14] P. Baldi and S. Brunak, *Bioinformatics: the machine learning approach*. MIT Press, 2001.

[15] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," p. 375, 2003.

[16] J. Wang and S. Singh, "Video analysis of human dynamics–a survey," *Real-time imaging*, vol. 9, no. 5, pp. 321–346, 2003.

[17] L. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[18] A. Velivelli, T. Huang, and A. Hauptmann, "Video shot retrieval using a kernel derived from a continuous hmm," Carnegie Mellon University, Tech. Rep. 980, 2006.

[19] B. Scholkopf and A. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.

[20] V. Chandola, D. Cheboli, and V. Kumar, "Detecting anomalies in a timeseries database," CS Technical Report 09-004, Computer Science Department, University of Minnesota, Tech. Rep., 2009.

[21] J. Han, M. Kamber, and J. Pei, *Data mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.