INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Sequential Adversarial Anomaly Detection for One-Class Event Data

Shixiang Zhu, Henry Shaowu Yuchi, Minghe Zhang, Yao Xie

To cite this article:
Shixiang Zhu, Henry Shaowu Yuchi, Minghe Zhang, Yao Xie (2023) Sequential Adversarial Anomaly Detection for One-Class Event Data. INFORMS Journal on Data Science 2(1):45-59. https://doi.org/10.1287/ijds.2023.0026

https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Vol. 2, No. 1, April–June 2023, pp. 45–59
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

Sequential Adversarial Anomaly Detection for One-Class Event Data

Shixiang Zhu,[a] Henry Shaowu Yuchi,[b] Minghe Zhang,[b] Yao Xie[b,*]

[a] Heinz College of Information Systems and Public Policy, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213; [b] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332

*Corresponding author

Contact: shixianz@andrew.cmu.edu, https://orcid.org/0000-0002-2241-6096 (SZ); shaowu.yuchi@gatech.edu (HSY); mzhang388@gatech.edu (MZ); yao.xie@isye.gatech.edu, https://orcid.org/0000-0001-6777-2951 (YX)

Received: March 18, 2021
Revised: March 15, 2022; November 7, 2022
Accepted: December 23, 2022
Published Online in Articles in Advance: March 31, 2023

https://doi.org/10.1287/ijds.2023.0026

Abstract. We consider the sequential anomaly detection problem in the one-class setting when only the anomalous sequences are available and propose an adversarial sequential detector by solving a minimax problem to find an optimal detector against the worst-case sequences from a generator. The generator captures the dependence in sequential events using the marked point process model. The detector sequentially evaluates the likelihood of a test sequence and compares it with a time-varying threshold, also learned from data through the minimax problem. We demonstrate our proposed method's good performance using numerical experiments on simulations and proprietary large-scale credit card fraud data sets. The proposed method can generally apply to detecting anomalous sequences.

History: W. Nick Street served as the senior editor for this article.
Funding: This work is partially supported by the National Science Foundation [Grants CAREER CCF-1650913, DMS-1938106, and DMS-1830210] and grant support from Macy's Technology.
Data Ethics & Reproducibility Note: The code capsule is available on Code Ocean at https://doi.org/10.24433/CO.2329910.v1 and in the e-Companion to this article (available at https://doi.org/10.1287/ijds.2023.0026).

Keywords: sequential anomaly detection • adversarial learning • imitation learning • credit card fraud detection

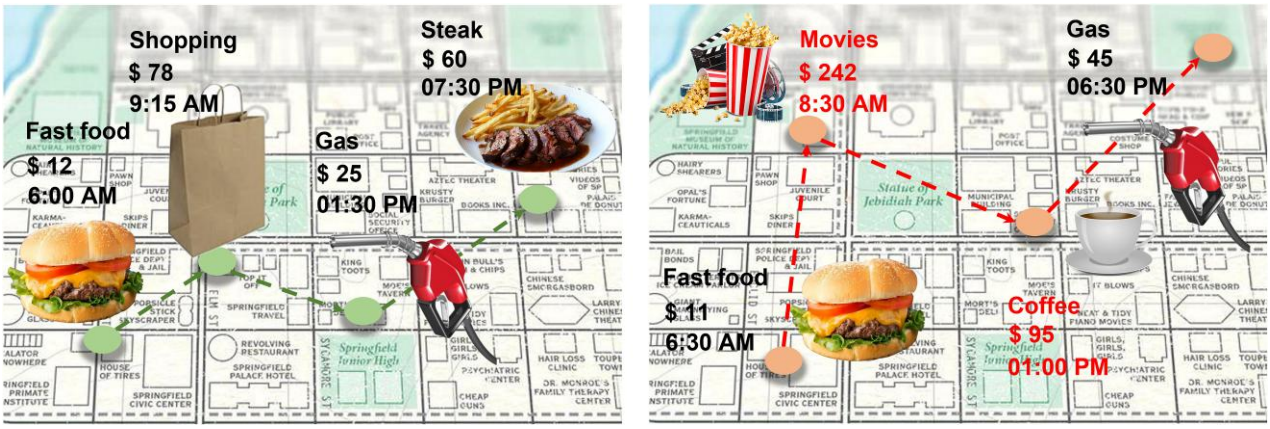
1. Introduction

Spatio-temporal event data are ubiquitous nowadays, ranging from electronic transaction records and earthquake activities recorded by seismic sensors to police reports. Such data consist of sequences of discrete events that indicate when and where each event occurred and other additional descriptions such as its category or volume. We are particularly interested in financial transaction fraud, which is often caused by stolen credit or debit card numbers from an unsecured website or due to identity theft. Collected financial transaction fraud typically consists of a series of anomalous events: unauthorized uses of a credit or debit card or similar payment tools (Automated Clearing House, Electronic Funds Transfer, recurring charge, etc.) to obtain money or property (FBI 2021). As illustrated in Figure 1, such events sequence corresponds to anomalous transaction records, typically including the time, location, amount, and type of the transactions.

Early detection of financial fraud plays a vital role in preventing further economic loss for involved parties. In today's digital world, credit card fraud and ID theft continue to rise in recent years. Losses to fraud incurred by payment card issuers worldwide reached USD$19.21 billion in 2019. Card issuers accounted for 68.97% of gross fraud losses (The Nilson Report 2019) because the liability usually comes down to the merchant or the card issuer, according to the "zero-liability policies"[1]: merchants and banks could face a significant risk of economic losses. Credit card fraud also causes much loss and trouble to the customers with stolen identities have been stolen: victims need to report unauthorized charges to the card issuer, canceling the current card, waiting for a new one in the mail, and subbing the new number into all auto-pay accounts linked to the old card. The entire process can take days or even weeks.

For applications such as financial fraud detection, we usually only have access to the anomalous event sequences. This can be due to protecting consumer privacy, so only fraudulent transaction data are collected for the study. The resulting one-class problem makes the task of anomaly detection even more challenging. However, there are distinctive patterns of anomalies that enable us to develop powerful detection algorithms. For instance, Figure 2(a) shows an example of a sequence of fraudulent transactions that we extracted from real data. A fraudster used a stolen card 12 times in just six days and made electronic transactions at stores that are physically far away from each other, ranging from California to New England. The types of transactions are

Figure 1. (Color online) Examples of a Sequence of Ordinary Events (left) and a Sequence of Anomalous Events (right) That Are Dependent: One Leads to Another



Notes. The events on the left show an ordinary pattern of transactions for a consumer. For the events on the right, it is unusual for a consumer to execute a transaction at a movie theater in the early morning and then a high-volume transaction at a coffee shop. It indicates the events are abnormal.

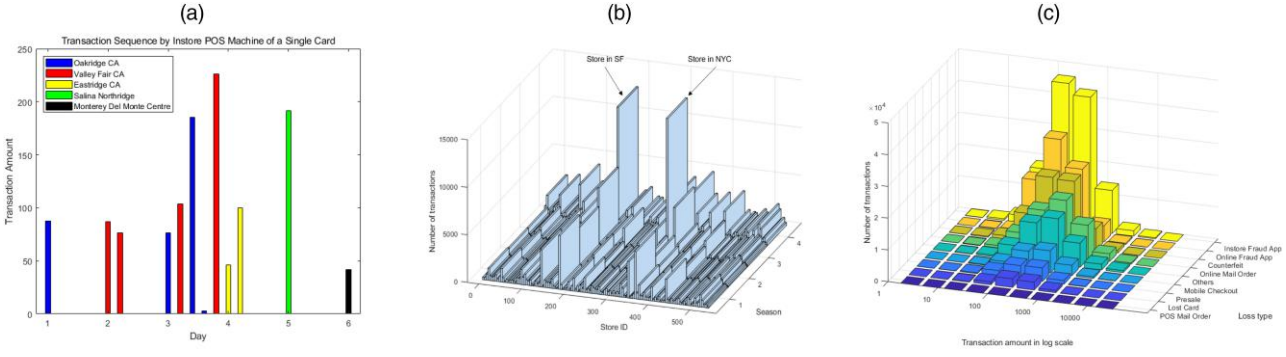
also different from the regular spending pattern. Figure 2(b) illustrates the distribution of a collection of fraudulent transactions for location (store ID), season, and the number of transactions. We can observe a significant portion of transactions at the department store in San Francisco and New York.

Although there has been much research effort in machine learning and statistics for anomaly detection using sequential data (Chandola et al. 2010, Xu 2010, Chung et al. 2015, Doshi and Yilmaz 2020), we cannot use existing methods here directly for the following reasons. First, many existing works consider detecting anomalous sequences "as a whole" rather than detecting in an online fashion. Second, the one-class data situation requires an unsupervised approach for anomaly detection; however, most sequential anomaly detection algorithms are based on supervised learning.

This paper presents an adversarial anomaly detection algorithm for one-class sequential detection, where only anomalous data are available. The adversarial sequential detector is solved from a minimax problem to find an optimal detector against the worst-case sequences from a generator that captures the dependence in sequential events using the marked point process model. The detector sequentially evaluates the likelihood of a test sequence and compares it with a time-varying threshold, which is also learned from data through the minimax problem. We demonstrate the proposed method's good performance by comparing state-of-the-art methods on synthetic and proprietary large-scale credit card fraud data provided by a major department store in the United States.

On a high level, our minimax formulation is inspired by imitation learning (Hussein et al. 2017), which

Figure 2. (Color online) Sequential Fraud Credit Card Transactions Data Set Provided by a Major Department Store in the United States



Notes. (a) Sequence of transactions made by one stolen credit card; each bar represents a fraudulent transaction, the bar's height indicates the transaction amount in dollars, and the color of the bar indicates the location of the transaction. (b) Overview of how these fraudulent transactions were distributed over stores and seasons. (c) Overview of how these fraudulent transactions were distributed over the amount of purchase for different loss types.

minimizes the maximum mean discrepancy (Gretton et al. 2012) (MMD). In particular, the generator is built on long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), which specifies the conditional distribution of the next event. We parameterize the detector by comparing the likelihood function of marked Hawkes processes with a deep Fourier kernel (Zhu et al. 2021b, c) with a threshold. The resulted likelihood function is computationally efficient to implement in the online fashion and can capture complex dependence between events in anomalous sequences. A notable feature of our framework is a *time-varying* threshold learned from data by solving the minimax problem, which achieves tight control of the false-alarms and hard to obtain precisely in theory. This is a drastic departure from prior approaches in sequential anomaly detection.

The rest of the paper is organized as follows. We first discuss the related work in sequential anomaly detection and revisit some basic definitions in imitation learning. Section 3 sets up the problem and introduces our sequential anomaly detection framework. Section 4 proposes a new marked point process model equipped with a deep Fourier kernel to model-dependent sequential data. Section 5 presents the adversarial sequence generator and learning algorithms. Finally, we present our numerical results on both real and synthetic data in Section 6. Proofs to all propositions can be found in the online appendix.

1.1. Related Work

Several research lines are related to this work, including imitation learning, the LSTM architecture for modeling sequence data, one-class anomaly detection, and fraud detection, which we review here.

Imitation learning (Hussein et al. 2017) aims to mimic the expert's behavior in a given task. An agent (a learning machine) is trained to perform a task from demonstrations by learning a mapping between observations and actions. Landmark works by Abbeel and Ng (2004) and Syed and Schapire (2007) attack this problem via *inverse reinforcement learning* (IRL). In their work, the learning process is achieved by devising a game-playing procedure involving two opponents in a zero-sum game. This alteration not only allows them to achieve the same goal of doing nearly as well as the expert as in Abbeel and Ng (2004) but achieves better performances in various settings. However, this strategy cannot be directly applied to event data modeling without adaptation. A recent work (Li et al. 2018) filled this gap by introducing a reward function with a nonparametric form, which measures the discrepancy between the training and generated sequences. Their proposed approach models the events using a temporal point process, which draws similarities in our work's spatio-temporal point process model. However, our work differs from Li et al. (2018) in two major ways. Rather than constructing a generative model, we focus on sequential anomaly detection, which is a different type of problem. Besides, we design a structured reward function that is more suitable for modeling the triggering effects between events and more computationally efficient to carry out.

There is another work on inverse reinforcement learning related to our work. The work in Ziebart et al. (2008) first proposed a probabilistic approach to the imitation learning problem via maximum entropy. The work proposed an efficient state frequency algorithm that is composed of both backward and forward passes recursively. A more recent similar article by Oh and Iyengar (2019) seeks to integrate IRL with anomaly detection based on the above maximum entropy IRL framework. They aim to learn the unknown reward function to test a given sequence. The significant difference, however, is they focus on time series data instead of event data. Also, we formulate the problem as a minimax optimization, whereas they used a Bayesian method to estimate the model parameters.

A large body of recent works performs sequential anomaly detection using LSTM, similar to the proposed stochastic LSTM used as the adversarial generator in our work. In Malhotra et al. (2016), the authors proposed an encoder-decoder scheme using LSTM to learn the normal behavior of data and used reconstruction errors to find anomalies. The work of Nanduri and Sherry (2016) built a recurrent neural network (RNN) model with LSTM structure to conduct anomaly detection for multivariate time series data for flight operations. Another paper from Luo et al. (2017) looks into anomaly detection in videos by convolutional neural networks with the LSTM modeling. It is clear that the LSTM model is versatile for various applications and can model unknown complex sequential data. However, the LSTM used here in this paper is stochastic (as a generative model to capture data distribution), whereas most LSTM architectures are deterministic. Specifically, the input at each time step in the stochastic LSTM is drawn from a random variable whose distribution is specified by the LSTM's parameters.

There is a wide array of existing research in anomaly detection. Principle component analysis (PCA) has traditionally been used to detect outliers, which naturally fits into anomaly detection. In Chalapathy et al. (2017), the authors propose a robust auto-encoder model, which is closely related to PCA for anomaly detection, and a deep neural network is introduced for the training process. Additionally, Chalapathy et al. (2018) proposed a one-class neural network to detect anomalies in complex data sets by creating a tight envelope around the normal data. It is improved from the one-class singular value decomposition formulation to be more robust. A closely related work is Ruff et al. (2018), which looks into one-class anomaly detection through a deep support vector

data description model that finds a data-enclosing hypersphere with minimum volume. However, most previous studies on one-class anomaly detection assumed independent and identically distributed data samples, whereas we consider data dependency.

As an important application of anomaly detection, credit card fraud detection has also drawn a lot of research interest (Bolton and Hand 2002, Kou et al. 2004). Most commonly, supervised methods have been adopted to use a database of known fraudulent/legitimate cases to construct a model that yields a suspicion score for new cases. Traditional statistical classification methods, such as linear discriminant analysis (Wang and Xu 2018), logistic classification (Sahin and Duman 2011), and *k*-nearest neighbors (Malini and Pushpa 2017), have proved to be effective tools for many applications. However, more powerful tools (Ghosh and Reilly 1994, Maes et al. 2002), especially neural networks, have also been extensively applied. Unsupervised methods are used when there are no prior sets of legitimate and fraudulent observations. A large body of approaches (Bolton et al. 2001, Srivastava et al. 2008, Tran et al. 2018) used here is usually a combination of profiling and outlier detection methods, which models a baseline distribution to represent the normal behavior and then detect observations departure from this. Compared with the previous unsupervised studies in credit card fraud detection, the most notable feature of our approach is to learn the fraudulent behaviors by "mimicking" the limited amount of anomalies via an adversarial learning framework. Our anomaly detector equipped with the deep Fourier kernel is more flexible than conventional approaches in capturing intricate marked spatio-temporal dynamics between events while being computationally efficient.

Our work is a significant extension of the previous conference paper (Zhu et al. 2020), which studies the one-class sequential anomaly detection using a framework of generative adversarial network (GAN) (Goodfellow 2014, Goodfellow et al. 2014) based on the cross-entropy between the real and generated distributions. Here, we focus on a different loss function motivated by imitation learning and MMD distances that is more computationally efficient. In addition, we introduce a new time-varying threshold, which can be learned in a data-driven manner.

2. Background: Inverse Reinforcement Learning

Because imitation learning is a form of reinforcement learning (RL), in the following, we will provide some necessary background about RL. Consider an agent interacting with the environment. At each step, the agent selects an action based on its current state, to which the environment responds with a reward value and the next state. The *return* is the sum of (discounted) rewards through the agent's trajectory of interactions with the environment. The *value function* of a policy describes the expected *return* from taking action from a state. The *inverse RL* (IRL) aims to find a reward function from the expert demonstrations explaining the expert behavior. Seminal works (Ng and Russell 2000, Abbeel and Ng 2004) provide a max-min formulation to address the problem. The authors propose a strategy to match an observed expert policy's value function and a learner's behavior. Let π denote the expert policy, and π_φ denote the learner policy, respectively. The optimal reward function r can be found as the saddle-point of the following max-min problem (Syed and Schapire 2007), that is,

$$\max_{r\in\mathcal{F}} \min_{\varphi\in\mathcal{G}} \left\{ \mathbb{E}_{\boldsymbol{x}\sim\pi}\left[\sum_{i=1}^{N_x} r(x_i, s_i)\right] - \mathbb{E}_{\boldsymbol{z}\sim\pi_\varphi}\left[\sum_{j=1}^{N_z} r(z_j, s_j)\right] \right\},$$

where $\mathcal{F}$ is the family class for reward function and $\mathcal{G}$ is the family class for learner policy. Here, $\boldsymbol{x} = \{x_1, \ldots, x_{N_x}\}$ is a sequence of actions generated by the expert policy π, $\boldsymbol{z} = \{z_1, \ldots, z_{N_z}\}$ is a roll-out sequence generated from the learner policy π_φ, and N_x and N_z are the numbers of actions for sequences $\boldsymbol{x}$ and $\boldsymbol{z}$, respectively. The formulation means that a proper reward function should provide the expert policy a higher reward than any other learner policy in $\mathcal{G}$. The learner can also approach the expert performance by maximizing this reward.

3. Adversarial Sequential Anomaly Detection

We aim to develop an algorithm to detect anomalous sequences when the training data set consists of only abnormal sequences and without normal sequences. In particular, the algorithm will process data sequentially and raise the alarm as soon as possible after the sequence has been identified as anomalous. Denote such a detector as ℓ with parameter θ. At each time t, the detector evaluates a statistic and compares it with a threshold. For a length-N sequence $\boldsymbol{x}$, define $\boldsymbol{x}_{1:i} := [x_1, \ldots, x_i]^\top$, $i = 1, 2, \ldots, N$ be its first i observations. We define the detector as a stopping rule, which stops and raises the alarm the first time that the detection statistic exceeds the threshold:

$$T = \inf\{t : \ell(\boldsymbol{x}_{1:i}; \theta) > \eta_i,\ t_i \le t < t_{i+1}\}, \quad i = 1, \ldots, N.$$

Once an alarm is raised, the sequence is flagged as an anomaly. If there is no alarm raised until the end of the time horizon, the sequence is considered normal. The test sequence can be an arbitrary (finite) length.

3.1. Proposed: Adversarial Anomaly Detection

Assume a set of anomalous sequences drawn from an empirical distribution π. Because normal sequences are not available, we introduce an *adversarial generator*, which produces "normal" sequences that are statistically similar

to the real anomalous sequences. The detector has to discriminate the true anomalous sequence from the counterfeit "normal" sequences. We introduce competition between the anomaly detector and the generator to drive both models to improve their performances until anomalies can distinguish from the worst-case counterfeits. We can also view this approach as finding the "worst-case" distribution that defines the "border region" for detection. Formally, we formulate this as a minimax problem as follows:

$$\min_{\varphi\in\mathcal{G}} \max_{\theta\in\Theta} J(\theta,\varphi) := \mathbb{E}_{\boldsymbol{x}\sim\pi}\, \ell(\boldsymbol{x};\theta) - \mathbb{E}_{\boldsymbol{z}\sim G_z(\varphi)}\, \ell(\boldsymbol{z};\theta), \quad (1)$$

where G_z is an adversarial generator specified by the parameter $\varphi \in \mathcal{G}$, and $\mathcal{G}$ is a family of candidate generators. Here the detection statistic corresponds to $\ell(\theta)$, the log-likelihood function of the sequence specified by $\theta \in \Theta$ and Θ is its parameter space. The choices of the adversarial generator and the detector are further discussed in Section 4. The detector compares the detection statistic to a threshold. We define the following.

Definition 1 (Adversarial Sequential Anomaly Detector). Denote the solution to the minimax problem (1) as (θ^*,φ^*). A sequential adversarial detector raises an alarm at the time i if

$$\ell(\boldsymbol{x}_{1:i};\theta^*) > \eta_i^*,$$

where the time-varying threshold $\eta_i^* \propto \mathbb{E}_{\boldsymbol{z}\sim G_z(\varphi^*)}\, \ell(\boldsymbol{z}_{1:i};\theta^*)$.

3.2. Time-Varying Threshold

We choose the time-varying threshold $\eta_i^* \propto \mathbb{E}_{\boldsymbol{z}\sim G_z(\varphi^*)}\, \ell(\boldsymbol{z}_{1:i};\theta^*)$. Because the value of log-likelihood function $\ell(x_{1:i};\theta^*)$ for partial sequence observation $\boldsymbol{x}_{1:i}$ may vary over the time step i (the ith event is occurred), we need to adjust the threshold accordingly for making decisions as a function of i. Our time-varying threshold η_i^* is different from sequential statistical analysis, where the threshold for performing detection is usually constant or preset (not adaptive to data) based on the known distributions of the data sequence (e.g., set the threshold growing over time as $\sqrt{t}$; Siegmund 1985). The rationale behind the design of the threshold η_i^* is that, at any given time step, the log-likelihood of the data sequence is larger than that of the generated adversarial sequence; therefore, η_i^* provides the tight lower bound for the likelihood of anomalous sequences $\ell(\boldsymbol{x};\theta^*)$ due to the minimization in (1). That is, for any $\varphi \in \mathcal{G}$,

$$\begin{aligned} 0 &\le \mathbb{E}_{\boldsymbol{x}\sim\pi}\, \ell(\boldsymbol{x}_{1:i};\theta^*) - \eta_i^* \\ &\le \mathbb{E}_{\boldsymbol{x}\sim\pi}\, \ell(\boldsymbol{x}_{1:i};\theta^*) - \mathbb{E}_{\boldsymbol{z}\sim G_z(\varphi)}\, \ell(\boldsymbol{z}_{1:i};\theta^*). \end{aligned}$$

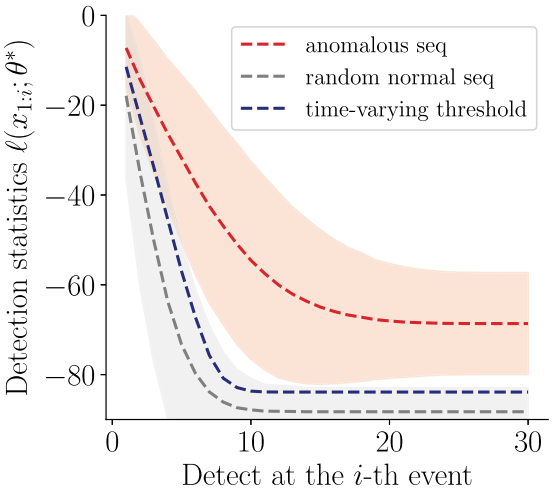
The adversarial sequences drawn from $G_z(\varphi^*)$ can be viewed as the normal sequences that are statistically "closest" to anomalous sequences. Therefore, the log-likelihood of such sequences in the "worst-case" scenario defines the "border region" for detection. In practice, the threshold η_i^* can be estimated by $1/n' \sum_{l=1}^{n'} \ell(\boldsymbol{z}_{1:i}^l;\theta^*)$, where $\{\boldsymbol{z}^l\}_{l=1,\dots,n'}$ are adversarial sequences sampled from $G_z(\varphi)$ and n' is the number of the sequences. As a real example presented in Figure 3, the time-varying threshold in the darker dashed line can sharply separate the anomalous sequences from the normal sequences. More experimental results are presented in Section 6.

3.3. Connection to Imitation Learning

The problem formulation (1) resembles the minimax formulation in IRL proposed by seminal works (Ng and Russell 2000, Abbeel and Ng 2004). As shown in Figure 4, an observed anomalous samples $\boldsymbol{x} \sim \pi$ can be regarded as an expert demonstration sampled from the expert policy π, where each $\boldsymbol{x} = \{x_1,\dots,x_N\}$ is a sequence of events with length of N and the sequences may be of different lengths. Each event $x_i, i = 1,\dots,N$ of the sequence is analogous to the ith action made by the expert given the history of past events $\{x_1, x_2,\dots,x_{i-1}\}$ as the corresponding state. Accordingly, the generator can be regarded as a learner that generates convincing counterfeit trajectories.

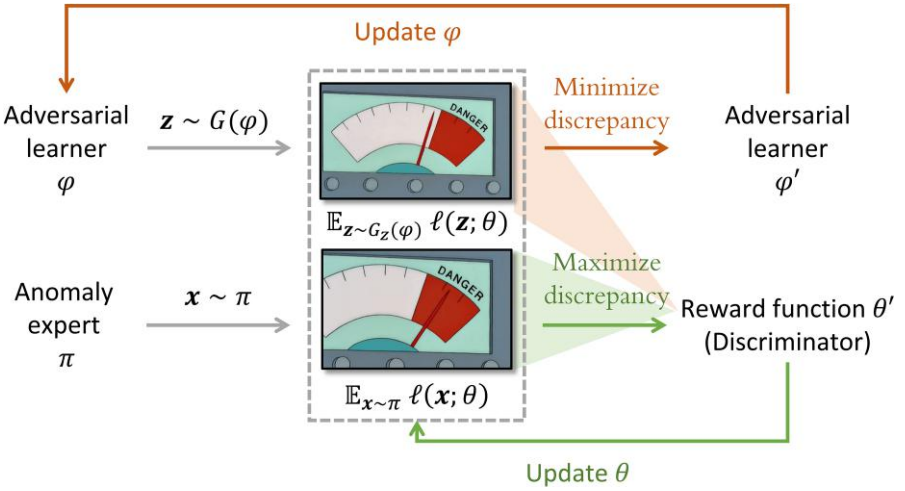
The log-likelihood of observed sequences can be interpreted as undiscounted *return*, that is, the accumulated

Figure 3. (Color online) Adversarial Anomaly Detection and Threshold Using Synthetic Data with 1,000 Synthetic Sequences



Notes. The two lighter dashed lines represent mean detection statistics ((θ^*)) for anomalous and normal sequences. The dashed line in the middle corresponds to the time-varying threshold suggested by our model. Clearly, the threshold can separate the anomalous sequences from the normal sequences.

Figure 4. (Color online) Imitation Learning Interpretation

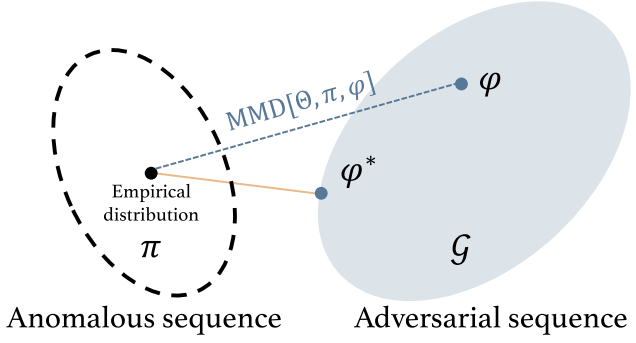


sum of rewards evaluated at past actions, where the logarithm of the conditional probability of each event (action) can be regarded as the event's reward. The ultimate goal of the proposed framework (1) is to close the gap between the return of the expert demonstrations and the return of the learner trajectories so that the counterfeit trajectories can meet the lower bound of the real demonstrations.

3.4. Connection to MMD-Like Distance

The proposed approach can also be viewed as minimizing a MMD-like distance metric (Gretton et al. 2012) as illustrated in Figure 5. More specifically, the maximization in (1) is analogous to an MMD metric in a reduced function class specified by Θ, that is, $\sup_{\theta\in\Theta} \mathbb{E}_{x\sim\pi}\, \ell(x;\theta) - \mathbb{E}_{z\sim\varphi}\, \ell(z;\theta)$, where Θ may not necessarily be a space of continuous, bounded functions on sample space. As shown in Gretton et al. (2012), if Θ is sufficiently expressive (universal), for example, the function class on reproducing kernel Hilbert space (RKHS), then maximization over such Θ is equivalent to the original definition. Based on this, we select a function class that serves our purpose for anomaly detection (characterizing the sequence's log-likelihood function), which has enough expressive power for our purposes. Therefore, the problem defined in (1) can be regarded as minimizing such an MMD-like metric between the empirical distribution of anomalous sequences and the distribution of adversarial sequences. The minimal MMD distance corresponds to the best "detection radius" that we can find without observing normal sequences.

Figure 5. (Color online) Empirical Distribution of Anomalous Sequences Is π



Notes. The assumed family of candidate generators is $\mathcal{G}$. Our proposed framework aims to minimize the maximum mean discrepancy (MMD) in a reduced function class Θ between π and $\varphi \in \mathcal{G}$.

4. Point Process with Deep Fourier Kernels

In this section, we present a model for the discrete events, which will lead to the detection statistic (i.e., the form of the likelihood function $\ell(x;\theta)$). We present a marked Hawkes process model that captures marked spatio-temporal dynamics between events. The most salient feature of the model is that we develop a novel deep Fourier kernel for Hawkes process (see section 6.6 in Mohri et al. (2012) for discussion of Fourier kernel), where the deep Fourier kernel empowers the model to characterize the intricate nonlinear dependence between events while enabling efficient computation of the likelihood function by leading to a closed-form expression of an integral in the likelihood function: a notorious difficulty in evaluating the likelihood function for Hawkes processes.

Assume each observation is a *marked spatio-temporal tuple* that consists of time, location, and marks: $x_i = (t_i, m_i)$, where $t_i \in [0, T)$ is the time of occurrence of the ith event, and $m_i \in \mathcal{M} \subseteq \mathbb{R}^d$ is the d-dimensional mark (here we treat location as one of a mark). The event's time is important because it defines the event's order and the time interval, which carry the key information.

4.1. Preliminary: Marked Temporal Point Processes

The marked temporal point processes (MTPPs) (Hawkes 1971, Reinhart 2018) offer a versatile mathematical framework for modeling sequential data consisting of an ordered sequence of discrete events localized in time and mark spaces (space or other additional information). They have proven useful in a wide range of applications (Embrechts et al. 1997, Clifton et al. 2011, Luca et al. 2014, Rambaldi et al. 2018, Li et al. 2017). Recent works (Du et al. 2016; Mei and Eisner 2017; Xiao et al. 2017a, b; Li et al. 2018; Upadhyay et al. 2018; Zhu et al. 2020) have achieved many successes in modeling temporal event data (some with marks) correlated in the time domain using RNNs.

Let $\{x_1, x_2, \ldots, x_{N_T}\}$ represent a sequence of observations. Denote N_T as the number of the points generated in the time horizon $[0, T)$. The events' distributions in MTPPs are characterized via a conditional intensity function $\lambda(t, m|\mathcal{H}_t)$, which is the probability of observing an event in the marked temporal space $[0, T) \times \mathcal{M}$ given the events' history $\mathcal{H}_t = \{(t_i, m_i)|t_i < t\}$, that is,

$$\lambda(t, m|\mathcal{H}_t) = \frac{\mathbb{E}[N([t, t+dt) \times B(m, dm))|\mathcal{H}_t]}{|B(m, dm)|dt}, \tag{2}$$

where $N(A)$ is the counting measure of events over the set $A \subseteq [0, T) \times \mathcal{M}$ and $|B(m, dm)|$ is the Lebesgue measure of the ball $B(m, dm)$ centered at m with radius dm. Assuming that influence from past events are linearly additive for the current event, the conditional intensity function of a Hawkes process is defined as

$$\lambda(t, m|\mathcal{H}_t) = \mu + \sum_{t_i < t} g(t - t_i, m - m_i), \tag{3}$$

where $\mu \geq 0$ is the background intensity of events, $g(\cdot, \cdot) \geq 0$ is the *triggering function* that captures spatio-temporal and marked dependencies of the past events. The triggering function can be chosen in advance, for example, in one-dimensional cases, $g(t - t_i) = \alpha \exp\{-\beta(t - t_i)\}$.

Let t_n denote the last occurred event before time t. The conditional probability density function of a point process is defined as

$$f(t, m|\mathcal{H}_t) = \lambda(t, m|\mathcal{H}_t) \exp\left\{ - \int_{t_n}^{t} \int_{\mathcal{M}} \lambda(t', m'|\mathcal{H}_{t'}) dm' dt' \right\}.$$

The log-likelihood of observing a sequence with N_T events denoted as $\boldsymbol{x} = \{(t_i, m_i)\}_{i=1}^{N_T}$ can be obtained by

$$\ell(\boldsymbol{x}; \theta) = \sum_{i=1}^{N_T} \log \lambda(t_i, m_i|\mathcal{H}_{t_i}) - \int_0^T \int_{\mathcal{M}} \lambda(t, m|\mathcal{H}_t) dm dt. \tag{4}$$

4.2. Hawkes Processes with Deep Fourier Kernel

One major computational challenge in evaluating the log-likelihood function is the computation of the integral in (4), which is multidimensional and performed in the possibly continuous mark and time-space. It can be intractable for a general model without a carefully crafted structure.

To tackle this challenge, we adopt an approach to represent the Hawkes process's triggering function via a Fourier kernel. The Fourier features spectrum is parameterized by a deep neural network, as shown in Figure 6. For the sake of notational simplicity, we denote $x := (t, m) \in \mathcal{X}$ as the most recent event and $x' := (t', m') \in \mathcal{X}$, $t' < t$ as an occurred event in the past, where $\mathcal{X} := [0, T] \times \mathcal{M} \subset \mathbb{R}^{d+1}$ is the space for time and mark. Define the conditional intensity function as

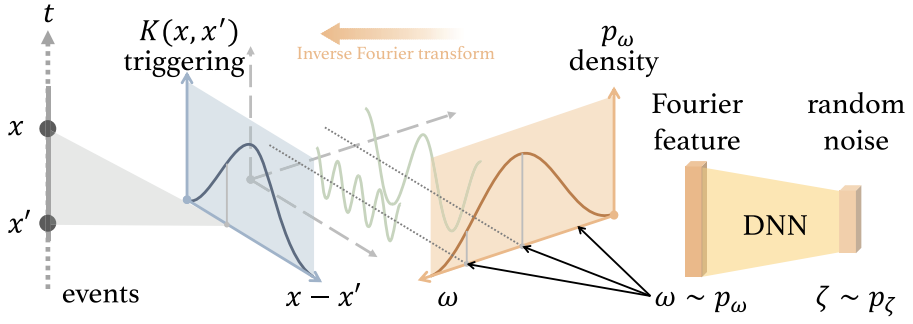
$$\lambda(x|\mathcal{H}_t; \theta) = \mu + \alpha \sum_{t' < t} K(x, x'), \tag{5}$$

where α represents the magnitude of the influence from the past, and $\mu \geq 0$ is the background intensity of events. The kernel function $K(x, x')$ measures the influence of the past event on the current event $x, x' \in \mathcal{X}$, and we will parameterize its kernel-induced feature mapping using a deep neural network $\theta \in \Theta$.

The formulation of deep Fourier kernel function relies on Bochner's theorem (Rudin 1962), which states that any bounded, continuous, and shift-invariant kernel is a Fourier transform of a bounded nonnegative measure:

Theorem 1 (Bochner (Rudin 1962)). *A continuous kernel of the form $K(x, x') = g(x - x')$ defined over a locally*

Figure 6. (Color online) Fourier Kernel Function $K(x, x')$ and Its Fourier Representation



Note. A deep neural network represents the spectrum of Fourier features.

compact set $\mathcal{X}$ is positive definite if and only if g is the Fourier transform of a nonnegative measure:

$$K(x,x') = g(x-x') = \int_{\Omega} p_{\omega}(\omega) e^{i w^{\top}(x-x')} d\omega, \qquad (6)$$

where $i = \sqrt{-1}$, p_{ω} is a nonnegative measure, Ω is the Fourier feature space, and kernels of the form $K(x,x')$ are called shift-invariant kernel.

If a shift-invariant kernel in (6) is positive semidefinite and scaled such that $g(0) = 1$, Bochner's theorem ensures that its Fourier transform p_{ω} can be viewed as a probability distribution function because it normalizes to one and is nonnegative. In this sense, the spectrum p_{ω} can be viewed as the distribution of r-dimensional Fourier features indexed by $\omega \in \Omega \subset \mathbb{R}^r$. Hence, we may obtain a triggering function in (5) between two events $x, x' \in \mathcal{X} \subset \mathbb{R}^{d+1}$, which satisfies the "kernel embedding."

Proposition 1. *Let the triggering function K be a continuous real-valued shift-invariant kernel and p_{ω} a probability distribution function. Then*

$$K(x,x') := \mathbb{E}_{\omega \sim p_{\omega}}[\phi_{\omega}(x) \cdot \phi_{\omega}(x')], \qquad (7)$$

where $\phi_{\omega}(x) := \sqrt{2}\cos(\omega^{\top} W x + u)$ and $W \in \mathbb{R}^{r \times (d+1)}$ is a weight matrix. These Fourier features $\omega \in \Omega \subset \mathbb{R}^r$ are sampled from p_{ω}, and u is drawn uniformly from $[0, 2\pi]$.

In practice, Expression (7) can be approximated empirically, that is,

$$\tilde{K}(x,x') = \frac{1}{D}\sum_{k=1}^{D} \phi_{\omega_k}(x) \cdot \phi_{\omega_k}(x') = \Phi(x)^{\top}\Phi(x'), \qquad (8)$$

where ω_k, $k = 1, \ldots, D$ are D Fourier features sampled from the distribution p_{ω}. The vector $\Phi(x) := [\phi_{\omega_1}(x), \ldots, \phi_{\omega_D}(x)]^{\top}$ can be viewed as the approximation of the kernel-induced feature mapping for the score. In the experiments, we substitute $\exp\{i w^{\top}(x-x')\}$ with a real-valued feature mapping, such that the probability distribution p_{ω} and the kernel K are real (Rahimi and Recht 2008).

The next proposition shows the empirical estimation (8) converges to the population value uniformly over all points in a compact domain $\mathcal{X}$ as the sample size D grows. It is a lower-variance approximation to (7).

Proposition 2. *Assume $\sigma_p^2 = \mathbb{E}_{\omega \sim p_{\omega}}[\omega^{\top}\omega] < \infty$ and a compact set $\mathcal{X} \subset \mathbb{R}^{d+1}$. Let R denote the radius of the Euclidean ball containing $\mathcal{X}$. Then for the kernel-induced feature mapping Φ defined in (8), we have*

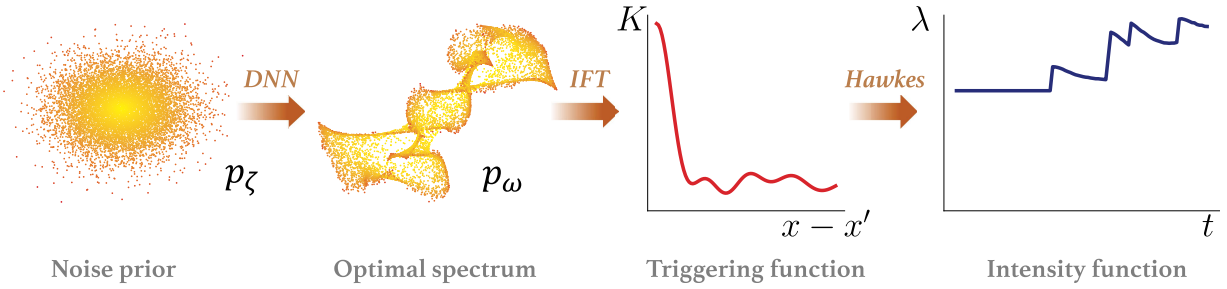
$$\mathbb{P}\left\{ \sup_{x,x' \in \mathcal{X}} \left| \Phi(x)^{\top}\Phi(x') - K(x,x') \right| \geq \epsilon \right\} \leq \left(\frac{48 R \sigma_p}{\epsilon}\right)^2 \exp\left\{ -\frac{D\epsilon^2}{4(d+3)} \right\}. \qquad (9)$$

The proposition ensures that kernel function can be consistently estimated using a finite number of Fourier features. In particular, for an error bound ϵ, the number of samples needed is on the order of $D = O((d+1)\log(R\sigma_p/\epsilon)/\epsilon^2)$, which grows linearly as data dimension d increases, implying the sample complexity is mild in the high-dimensional setting.

To represent the distribution p_{ω}, we assume it is a transformation of random noise $\zeta \sim p_{\zeta}$ through a nonlinear mapping $\psi_0 : \mathbb{R}^q \to \mathbb{R}^r$, as shown in Figure 6, where ψ_0 is a differentiable, it is represented by a deep neural network, and q is the dimension of the noise. Roughly speaking, p_{ω} is the probability density function of $\psi_0(\zeta)$, $\zeta \sim p_{\zeta}$. The triggering kernel is jointly controlled by the deep network parameters and the weight matrix W. We represent the Fourier feature generator as G_{ζ} and denote its parameters as $\theta \in \Theta$.

Figure 7 gives an illustrative example of representing the conditional intensity given sequence history using our approach. We choose $q = r = 2$ to visualize the noise prior p_{ζ} and the optimal spectra p_{ω}^* in a two-dimensional space. The optimal spectrum learned from data uniquely specifies a kernel function capable of capturing various nonlinear triggering effects. Unlike Hawkes processes, underlying long-term influences of some events, in this case, can be preserved in the intensity function.

Figure 7. (Color online) Instance of Calculating the Conditional Intensity λ Through Performing Inverse Fourier Transform



Notes. (1) Generate random noise. (2) Map the noises to the frequencies according to the optimal spectrum. (3) Perform inverse Fourier transform (IFT) in the frequency domain and obtain the triggering function. (4) Calculate the intensity function based on the triggering function.

4.3. Efficient Computation of Log-Likelihood Function

As discussed in Section 4.2, the technical difficulty of evaluating the log-likelihood function is to perform the multidimensional integral of the kernel function. In particular, given a sequence of events $\boldsymbol{x}$, the log-likelihood function of our model can be written by substituting the conditional intensity function in (4) with (5), and thus we need to evaluate $\int_{\mathcal{X}}\lambda(x|\mathcal{H}_t;\theta)dx$. In many existing works, this term is carried out by numerical integration, which can be computationally expensive. For instance, if we randomly sample κ points in a d-dimensional space and the total number of events is N, the computational complexity will be $O(\kappa DN)$ ($\kappa \gg N^d$) using common numerical integration techniques. Here we present a way to simplify the computation by deriving a closed-form expression for the integral as presented in the following proposition: a benefit offered by the Fourier kernel.

Proposition 3 (Integral of Conditional Intensity Function). *Let $t_{N_T+1}=T$ and $t_0=0$. Given ordered events $\{x_1,\dots,x_{N_T}\}$ in the time horizon $[0,T]$. The integral term in the log-likelihood function can be written as*

$$\int_{\mathcal{X}}\lambda(x|\mathcal{H}_t;\theta)dx=\mu T(b-a)^d+\frac{1}{D}\sum_{k=1}^{D}\sum_{i=0}^{N_T}\sum_{t_j<t_i}\cos(-\omega_k^\top W x_j)\cos\left(\frac{t_{i+1}+t_i}{2}\right)\sin\left(\frac{t_{i+1}-t_i}{2}\right)\cos^d\left(\frac{b+a}{2}\right)\sin^d\left(\frac{b-a}{2}\right)\prod_{\ell=1}^{d+1}\frac{2e^{\omega_k^\top w_\ell}}{\omega_k^\top w_\ell}, \tag{10}$$

where $w_\ell, \ell=1,\dots,d$ is the ℓth column vector in the matrix W, and $[a,b]$ are the range for each dimension of the mark space $\mathcal{M}$. The computational complexity is $O(DN)$.

Remark 1. From the right-hand side of (10), the second term only depends on the weight matrix W, D randomly sampled Fourier features, the time of events that occurred before t, and the region of the marked space. If we rescale the range of each coordinate of the mark to be $[0,2\pi]$, that is, $b=2\pi$ and $a=0$, then the second term of the integral equals to zero, and the integral defined in (10) can be further simplified as

$$\int_{\mathcal{X}}\lambda(x|\mathcal{H}_t;\theta)dx=\mu T(2\pi)^d.$$

In particular, when we only consider time ($d=0$), the integral becomes

$$\int_{\mathcal{X}}\lambda(x|\mathcal{H}_t;\theta)dx=\mu T+\frac{1}{D}\sum_{k=1}^{D}\sum_{i=0}^{N_T}\sum_{t_j<t_i}\cos(-\omega_k^\top W x_j)\cos\left(\frac{t_{i+1}+t_i}{2}\right)\sin\left(\frac{t_{i+1}-t_i}{2}\right)\frac{2e^{\omega_k^\top W}}{\omega_k^\top W}.$$

4.4. Recursive Computation of Log-Likelihood Function

Leveraging the conditional probability decomposition, we can compute of the log-likelihood function $\ell(\boldsymbol{x}_{1:i};\theta^*)$ recursively:

$$\ell(\boldsymbol{x}_{1:1};\theta^*)=\log f(x_1|\mathcal{H}_{t_1});$$
$$\ell(\boldsymbol{x}_{1:i};\theta^*)=\ell(\boldsymbol{x}_{1:i-1};\theta^*)+\log f(x_i|\mathcal{H}_{t_i};\theta^*),\quad \forall i>1, \tag{11}$$

where

$$f(x_i|\mathcal{H}_{t_i};\theta)=\lambda(x_i|\mathcal{H}_{t_i};\theta)e^{-\mu(t_i-t_{i-1})(2\pi)^d}.$$

This recursive expression makes it convenient to evaluate the detection statistic sequentially and perform online detection, which we summarize in Algorithm 1.

Algorithm 1 (Online Detection Algorithm)

Input: An unknown sequence $\boldsymbol{x}$ with N_T events and optimal model parameters θ^*,φ^*;

Generate D Fourier features from $G_\zeta(\theta^*)$ denoted as $\widehat{\Omega}=\{\omega_k\}_{k=1,\dots,D}$;

Generate n' adversarial sequences from $G_z(\varphi^*)$ denoted as $\widehat{Z}=\{\boldsymbol{z}^l\}_{l=1,\dots,n'}$;

while $i\le N_T$ **do**

  Compute the log-likelihood $\ell(\boldsymbol{x}_{1:i};\theta^*)$ given $\widehat{\Omega}$ according to (11);

  $\eta_i^*\leftarrow 1/n'\sum_{l=1}^{n'}\ell(\boldsymbol{z}_{1:i}^l;\theta^*)$;

  if $\ell(\boldsymbol{x}_{1:i};\theta)\ge\eta_i^*$ **then**

    Declare that it is an anomaly and record the stopping time t_i;

  end

  $i\leftarrow i+1$;

end

Declare that it is not an anomaly;

5. Adversarial Sequence Generator

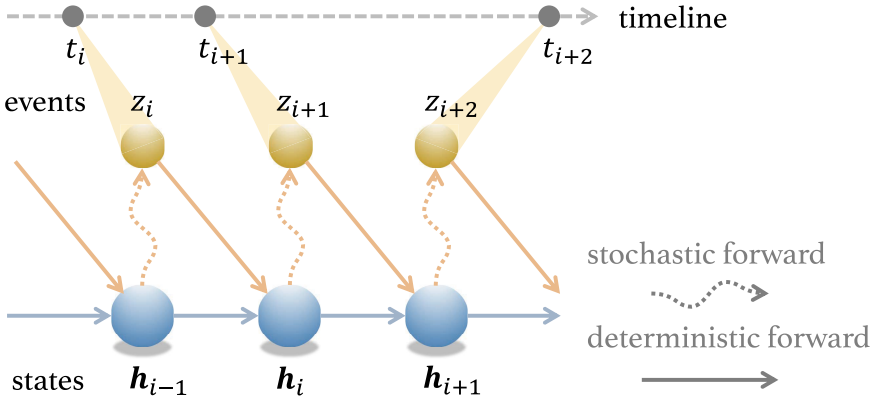
Now we describe the parameterization for the adversarial sequence generator. To achieve rich representation power for the adversarial generator G_z, we borrow the idea of the popular RNN structure.

In particular, we develop an RNN-type generator with stochastic neurons (Chung et al. 2015, Li et al. 2018) as shown in Figure 8, which can represent the nonlinear and long-range sequential dependency structure. Denote the ith generated adversarial event as $z_i:=(t_{i-1}+\Delta t_i,m_i)$, where Δt_i is the time interval between event z_{i-1},z_i. The generating process is described here:

$$\begin{aligned}[\Delta t_i,m_i^\top]^\top&\sim\mathcal{N}(\boldsymbol{\mu}_{i-1},\mathrm{diag}(\boldsymbol{\sigma}_{i-1})),\\ [\boldsymbol{\mu}_i,\boldsymbol{\sigma}_i^\top]^\top&=\psi_1(\boldsymbol{h}_i),\\ \boldsymbol{h}_i&=\psi_2(\boldsymbol{h}_{i-1},z_i),\quad i=1,\dots,N_T,\\ \boldsymbol{h}_0&=\boldsymbol{0},\end{aligned}$$

where the hidden state $\boldsymbol{h}_i\in\mathbb{R}^p$ encodes the sequence of past events $\{z_1,\dots,z_{i-1}\},z_i\in\mathcal{X}$, and $\mathcal{N}(\boldsymbol{\mu},\Sigma)$ stands for the multivariate Gaussian distribution with mean

Figure 8. (Color online) RNN-Based Adversarial Sequence Generator



$\boldsymbol{\mu} \in \mathbb{R}^{d+1}$ and covariance matrix $\Sigma \in \mathbb{R}^{(d+1)\times(d+1)}$; here, we only consider variance terms, and thus the covariance matrix is diagonal with diagonal entries specified by a vector $\boldsymbol{\sigma}_i$, and diag($\boldsymbol{x}$) means to convert the vector $\boldsymbol{x}$ to a diagonal matrix. Here we adopt the (two-sided) truncated normal distribution in our adversarial sequence generator by bounding the support of each mark to the interval (a, b). The probabilistic density function (p.d.f.) therefore is given by $\mathcal{N}(x|\mu,\sigma,a,b) = (1/\sigma)\phi((x-\mu)/\sigma)/(\Phi((b-\mu)/\sigma) - \Phi((a-\mu)/\sigma))$, where $\phi(x)$ is the p.d.f. of a standard normal distribution, and $\Phi(x)$ is the corresponding cumulative density function (c.d.f.); μ, σ are represented by the LSTM structure, and a, b are determined such that the percentage of density that lie within an interval for the normal is 99.7% (the so-called three-sigma rule-of-thumb). The process stops running until $t_i < T$ and $t_i + \Delta t_{i+1} \geq T$.

Function $\psi_2 : \mathbb{R}^{p+d+1} \to \mathbb{R}^p$ is an extended LSTM cell, and function $\psi_1 : \mathbb{R}^p \to \mathbb{R}^{(d+1)^2+d+1}$ can be any nonlinear mappings. There are two significant differences from the vanilla version of RNNs: (1) the outputs are sampled from hidden states rather than obtained by deterministic transformations (as in the vanilla version; randomly sampling will allow the learner to explore the events' space); and (2) the sampled time point will be fed back to the RNN. The model architecture for ψ_1 may be problem specific. For example, ψ_1 can be represented by convolution neural network (CNN) (LeCun et al. 1995) if the high-dimensional marks are images and can be represented by LSTM or Bidirectional Encoder Representations from Transformers (Devlin et al. 2019) if the marks are text. In this paper, because the mark is three-dimensional, we use a fully connected neural network to represent ψ_1, which achieves significantly better performance than baselines. The set of all trainable parameters in ψ_1, ψ_2 are denoted by $\varphi \in \mathcal{G}$.

Algorithm 2 (Adversarial Learning Algorithm)
input: data set $X = \{\boldsymbol{x}^i\}_{i=1,\ldots,n}$;
initialization: model parameters θ, φ;
for $1, \ldots, M_0$ **do**
(1) Randomly draw n'' training sequences from X denoted as $\widehat{X} = \{\boldsymbol{x}^l \in X\}_{l=1,\ldots,n''}$;
(2) Generate n' adversarial sequences from $G_z(\varphi)$ denoted as $\widehat{Z} = \{\boldsymbol{z}^l\}_{l=1,\ldots,n'}$;
(3) Generate D Fourier features from $G_\zeta(\theta)$ denoted as $\widehat{\Omega} = \{\omega_k\}_{k=1,\ldots,D}$;
Update φ by descending gradient given $\widehat{X}, \widehat{Z}, \widehat{\Omega}$:

$$\nabla_\varphi \frac{1}{n''}\sum_{l=1}^{n''} \ell(\boldsymbol{x}^l;\theta) - \frac{1}{n'}\sum_{l=1}^{n'} \ell(\boldsymbol{z}^l;\theta);$$

for $1, \ldots, M_1$ **do**
Redo steps (1), (2), (3) to obtain new $\widehat{X}, \widehat{Z}, \widehat{\Omega}$;
Update θ by ascending gradient given $\widehat{X}, \widehat{Z}, \widehat{\Omega}$:

$$\nabla_\theta \frac{1}{n''}\sum_{l=1}^{n''} \ell(\boldsymbol{x}^l;\theta) - \frac{1}{n'}\sum_{l=1}^{n'} \ell(\boldsymbol{z}^l;\theta);$$

end
end

We learn the adversarial detector's parameters in an offline fashion by performing alternating minimization between optimizing the generator $G_z(\varphi)$ and optimizing the anomaly discriminator $\ell(\theta)$, using stochastic gradient descent. Let M_0 be the number of iterations, and M_1 be the number of steps to apply to the discriminator. Let $n', n'' < n$ be the number of generated adversarial sequences and the number of training sequences in a mini-batch, respectively. We follow the convention of choosing mini batch size in stochastic optimization algorithm (Li et al. 2014) and only require use of the same value for both n' and n''. There is a clear tradeoff between the model generalization and the estimation accuracy. Large n' and n'' tend to converge to sharp minimizers of the training and testing functions, which lead to poorer generalization. In contrast, small n' and n'' consistently converge to flat minimizers due to the inherent noise in the gradient estimation. Large n' and

n'' may cause the training to be computationally expensive. The learning process is summarized in Algorithm 2.

6. Numerical Experiments

In this section, comprehensive numerical studies are presented to compare the proposed adversarial anomaly detector's performance with the state-of-the-art.

6.1. Comparison and Performance Metrics

We compare our method (referred to as `AIL`) with four state-of-the-art approaches: the one-class support vector machine (Zhang et al. 2007) (`One-class SVM`), the cumulative sum of features extracted by principal component analysis (Page 1954) (`PCA+CUMCUM`), the local outlier factor (Breunig et al. 2000) (`LOF`), and a recent work leveraging IRL framework for sequential anomaly detection (Oh and Iyengar 2019) (`IRL-AD`).

The performance metrics are standard, including precision, recall, and F_1 score, all of which have been widely used in the information retrieval literature (Michael et al. 2002). This choice is because anomaly detection can be viewed as a binary classification problem, where the detector identifies if an unknown sequence is an anomaly. The F_1 score combines the *precision* and *recall*. Define the set of all true anomalous sequences as U and the set of positive sequences detected by the optimal detector as V. Then precision P and recall R are defined by

$$P = |U \cap V|/|V|, \; R = |U \cap V|/|U|,$$

where $|\cdot|$ is the number of elements in the set. The F_1 score is defined as $F_1 = 2PR/(P+R)$ and the higher F_1 score the better. Because positive and negative samples in real data are highly unbalanced, we do not use the receiver operating characteristic curve (true-positive rate versus false-positive rate) in our setting.

6.2. Experiments Setup

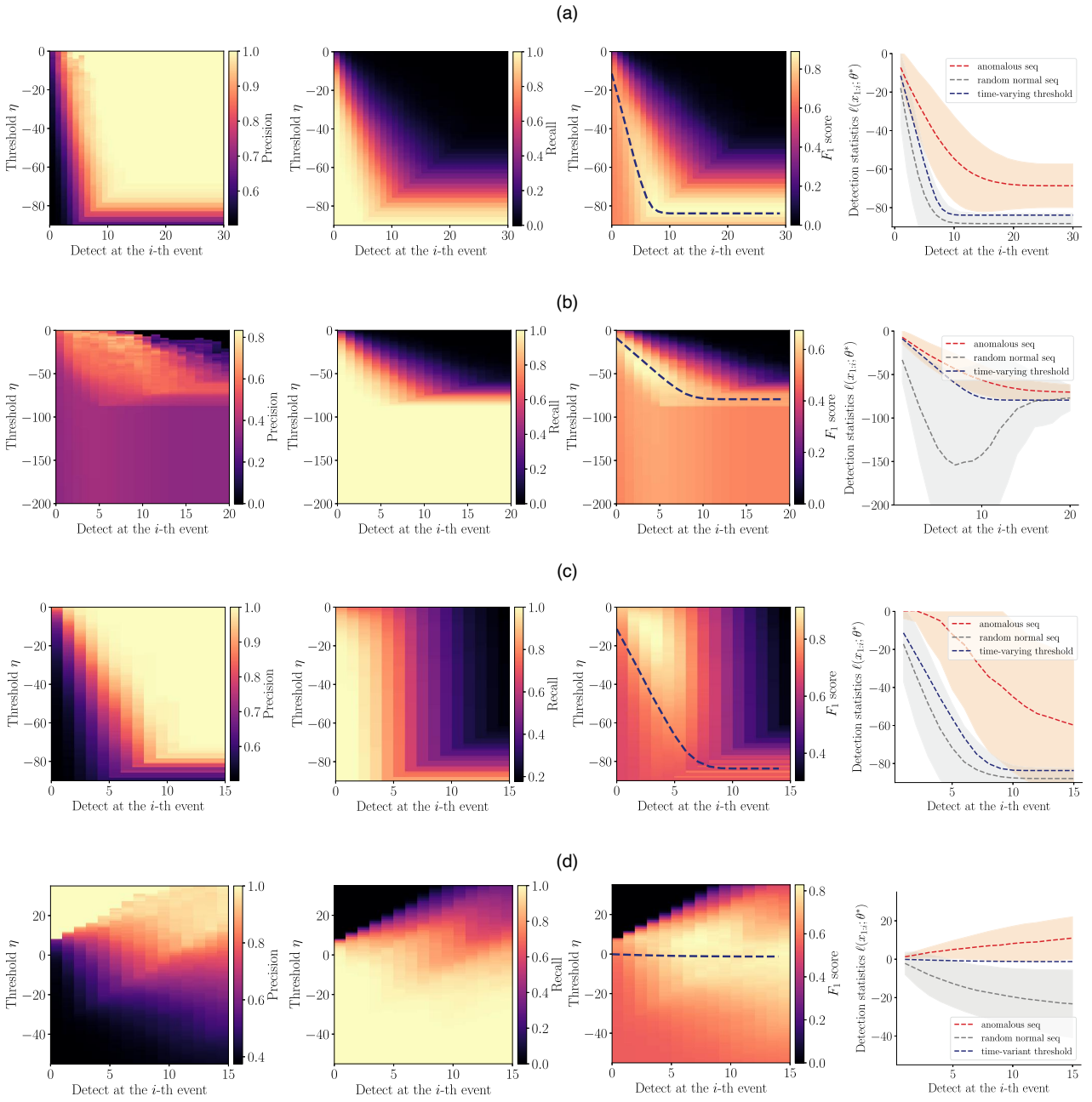
Consider two synthetic and two real data sets. (1) Singleton synthetic data consist of 1,000 anomalous sequences with an average length of 32. Each sequence is simulated by a Hawkes process with an exponential kernel specified in (3), where $\beta = 3$ and $\mu = 10, \alpha = 1$. (2) composite synthetic data consist of 1,000 mixed anomalous sequences with an average length of 29. Every 200 of the sequences are simulated by five Hawkes processes with different exponential kernels, where $\mu = 10$, $\alpha = 1$, and $\beta = 1, 2, 3, 4, 5$, respectively. (3) Real credit card fraud data consist of 1,121 fraudulent credit transaction sequences with an average length of 21. Each anomalous transaction in a sequence includes the occurrence geolocation (latitude and longitude), time, and corresponding transaction amount in the dollar. (4) Robbery data contain the 911-calls-for-service events in Atlanta from 2015 to 2017 (Zhu and Xie 2018, 2019a, b; Zhu et al. 2021a). We consider each crime series as a sequence of events: each event consists of the time (in seconds) and the geolocation (in latitude and longitude), indicating when and where the event occurred. We extract a series of events in the same category identified by the police detectives and treat them as one sequence. There exists intricate spatial and temporal dependency between these events with the same category. As indicated by Zhu and Xie (2019b), the 911 calls of some crime incidents committed by the same individual share similar crime behaviours (e.g., forced entry) and tend to aggregate in time and space. This phenomenon is called modus operandi (M.O.) (Wang et al. 2015). Within two years of data, this gives us 44 sequences with the subcategory of robbery. We test whether the algorithm can discriminate a series that is a robbery series or not. To create such an experiment, we also created 391 other types of crime series, which consist of randomly selected categories mixed together. We treat them as "anomalous" and "normal" data, respectively. In the experiments, we under-sample the Fourier features, where $D = 20$, to improve training efficiency. In addition, we select $n' = n'' = 32$ empirically based on the computational resource of the experimental setup on a standard laptop with a quad-core 4.7-GHz processor. The model obtains its convergence around $M_0 = 1,000$ iterations with $M_1 = 5$.

Our evaluation procedure is described as follows. We consider two sets of simulation data and two sets of real data, respectively. Each data set is divided into 80% for training and 20% for testing. To evaluate the performance of the fitted model, we first mix the testing set with 5,000 normal sequences, which are simulated by multiple Poisson processes, and then perform online detection. We do not simulate normal sequences for the robbery data experiment because we treat other types of crime as the alternative. The precision, recall, and F_1 score will be recorded accordingly. The method with higher precision, recall, and F_1 score at an earlier time step is more favorable than the others.

6.3. Results

First, we summarize the performance of our method on three data sets in Figure 9 and confirm that the proposed time-varying threshold can optimally separate the anomalies from normal sequences. To be specific, the fourth column in Figure 9 shows the average log-likelihood (detection statistics) and its corresponding 1σ region for both anomalous sequences and normal sequences. As we can see, the anomalous sequences attain a higher average log-likelihood than the normal sequences for all three data sets. Their log-likelihoods fall into different value ranges with rare overlap. Additionally, the time-varying threshold indicated by darker

Figure 9. (Color online) Performance of Our Method (AIL) on Four Data Sets

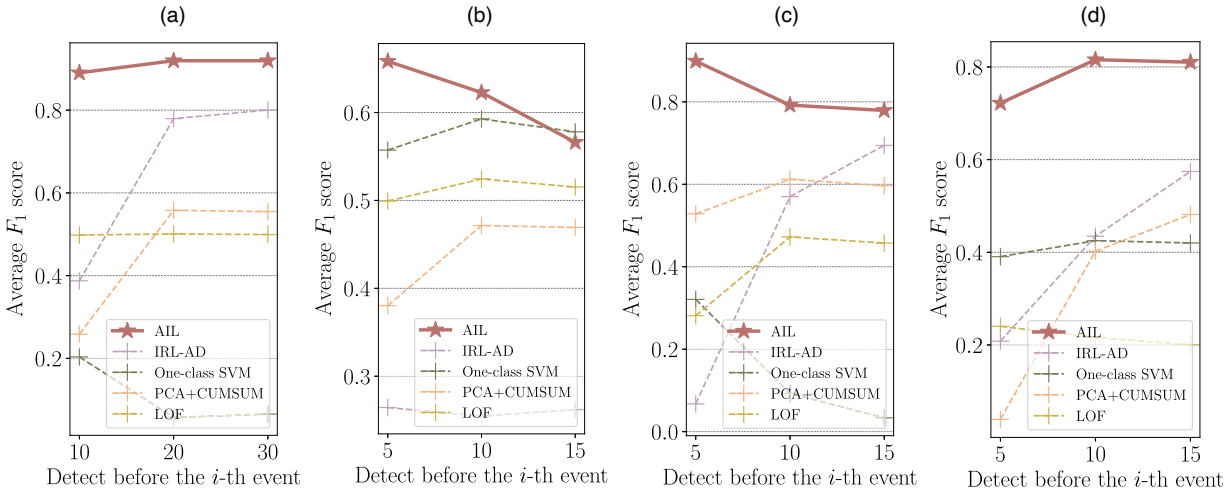


Notes. The first three columns correspond to the precision, recall, and F_1 score of our method using different thresholds. The dashed lines in the third column indicate our time-varying thresholds. The fourth column shows the step-wise detection statistics for both anomalous and normal sequences. (a) Singleton synthetic data. (b) Composite synthetic data. (c) Real credit card fraud data. (d) Real robbery data.

dash lines lies between the value ranges of anomalous and normal sequences, which produces an amicable separation of these two types of sequences at any given time. The first three columns in Figure 9 present more compelling evidence that the time-varying threshold is near-optimal. Colored cells of these heat maps are calculated with different constant thresholds η at each step i by performing cross-validation. The brightest regions indicate the "ground truth" of the optimal choices of the threshold. As shown in the third column, the time-varying thresholds (dashed line) are very close to the optimal choices found by cross-validation.

We also compare the stepwise F_1 scores of our method with the other four baselines in Figure 10. The results show that (1) from an overall standpoint, our method outperforms other baselines with significantly higher F_1 scores, and (2) our method allows for easier and faster detection of anomalous sequences (before 10

Figure 10. (Color online) Performance of Our Method (AIL) and Four Baselines on Three Data Sets



Notes. The marks show the average F_1 score tested on testing sequences when decisions are made with observing part of the sequences. (a) Singleton synthetic. (b) Composite synthetic. (c) Credit card fraud. (d) Robbery.

events being observed in our experiments), which is critically vital in sequential scenarios for most of the applications.

Finally, we present an ablation study to investigate the performance of our method using different generators. As shown in Table 1, the proposed generator based on an extended LSTM structure significantly outperforms other generators in stepwise F_1 score. As a sanity check, the generator using the vanilla Hawkes process achieves competitive performances on the singleton synthetic data because the true anomalous sequences are from a Hawkes process. However, we can observe a dramatic performance deterioration on the composite synthetic data. The anomalous sequences are generated by multiple distributions and can hardly be captured by the vanilla Hawkes process. This result confirms that using a generic generative model cannot achieve the best performance.

7. Conclusion and Discussions

We presented a novel unsupervised anomaly detection framework on sequential data based on adversarial learning. A robust detector can be found by solving a minimax problem, and the optimal generator also helps define the time-varying threshold for making decisions in an online fashion. We model the sequential event data using a marked point process model with a neural Fourier kernel. Using both synthetic and real data, we demonstrated that our proposed approach outperforms other state-of-the-art. In particular, the experimental results suggest that the proposed framework has achieved excellent performance on a proprietary large-scale credit-card fraud data set from a major department store in the United States, which shows the potential of proposed methods to apply to real-world problems.

Given the prevalence of sequential event data (in many applications, there is only one-class data), we believe our proposed method can be broadly applicable to many scenarios. Such applications include financial anomaly detection, Internet intrusion detection, and system anomaly detection such as power systems cascading failures, all of which are sequential discrete events data with complex temporal dependence. On the methodology side, we believe the proposed framework is a natural way to tackle the one-class anomaly detection problem, leveraging adversarial learning advances.

Table 1. F_1 Score Before ith Event Using Different Adversarial Generators in the Proposed Framework

	Singleton synthetic data			Composite synthetic data		
Generator in AIL	$i=5$	$i=10$	$i=15$	$i=5$	$i=10$	$i=15$
Vanilla Hawkes process	0.821	0.889	0.911	0.421	0.411	0.370
Vanilla LSTM	0.761	0.830	0.878	0.594	0.542	0.519
Proposed extended LSTM	0.888	0.916	0.916	0.658	0.623	0.566

It may provide a first step toward bridging imitation learning and sequential anomaly detection.

Endnote

[1] See https://usa.visa.com/pay-with-visa/visa-chip-technology-consumers/zero-liability-policy.html.

References

Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. *Proc. 21st Internat. Conf. on Machine Learn.* (ACM, New York).

Bolton RJ, Hand DJ (2002) Statistical fraud detection: A review. *Statist. Sci.* 17(3):235–255.

Bolton RJ, Hand DJ (2001) Unsupervised profiling methods for fraud detection. *Credit Research Centre, University of Edinburgh*, 235–255.

Breunig M, Kriegel HP, Ng RT, Sander J (2000) LOF: Identifying density-based local outliers. *Proc. ACM SIGMOD Internat. Conf. on Management of Data* (ACM, New York), 93–104.

Chalapathy R, Menon AK, Chawla S (2017) Robust, deep and inductive anomaly detection. Ceci M, Hollmén J, Todorovski L, Vens C, Dzeroski S, eds. *Proc. Eur. Conf. on Machine Learn. and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, vol. 10534 (Springer, Berlin), 36–51.

Chalapathy R, Menon AK, Chawla S (2018) Anomaly detection using one-class neural networks. Preprint, submitted February 18, https://arxiv.org/abs/1802.06360.

Chandola V, Banerjee A, Kumar V (2010) Anomaly detection for discrete sequences: A survey. *IEEE Trans. Knowledge Data Engrg.* 24(5):823–839.

Chung J, Kastner K, Dinh L, Goel K, Courville AC, Bengio Y (2015) A recurrent latent variable model for sequential data. Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, eds. *Advances in Neural Information Processing Systems*, vol. 28 (Curran Associates, Red Hook, NY), 2980–2988.

Clifton DA, Hugueny S, Tarassenko L (2011) Novelty detection with multivariate extreme value statistics. *J. Signal Processing Systems* 65(3):371–389.

Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. Burstein J, Doran C, Solorio T, eds. *Proc. Conf. of the North American Chapter of the Assoc. for Comput. Linguistics: Human Language Tech* (Association for Computational Linguistics, Stroudsburg, PA), 4171–4186.

Doshi K, Yilmaz Y (2020) Any-shot sequential anomaly detection in surveillance videos. *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops* (IEEE, Piscataway, NJ), 934–935.

Du N, Dai H, Trivedi R, Upadhyay U, Gomez-Rodriguez M, Song L (2016) Recurrent marked temporal point processes: Embedding event history to vector. *Proc. 22nd ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining* (ACM, New York), 1555–1564.

Embrechts P, Mikosch T, Klüppelberg C (1997) *Modelling Extremal Events: For Insurance and Finance* (Springer-Verlag, Berlin).

FBI (2021) Credit card fraud. https://www.fbi.gov/scams-and-safety/common-scams-and-crimes/credit-card-fraud.

Ghosh S, Reilly DL (1994) Credit card fraud detection with a neural-network. *Proc. 27th Hawaii Internat. Conf. on Systems Sci.*, vol. 3 (IEEE, Piscataway, NJ), 621–630.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, et al. (2014) Generative adversarial nets. Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ, eds. *Advances in Neural Information Processing Systems*, vol. 27 (Curran Associates, Red Hook, NY), 2672–2680.

Goodfellow IJ (2014) On distinguishability criteria for estimating generative models. Preprint, submitted December 14, https://arxiv.org/abs/1412.6515.

Gretton A, Borgwardt KM, Rasch MJ, Schölkopf B, Smola A (2012) A kernel two-sample test. *J. Machine Learn. Res.* 13(25):723–773.

Hawkes AG (1971) Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58(1):83–90.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput.* 9(8):1735–1780.

Hussein A, Gaber MM, Elyan E, Jayne C (2017) Imitation learning: A survey of learning methods. *ACM Comput. Survey* 50(2):1–35.

Kou Y, Lu CT, Sirwongwattana S, Huang YP (2004) Survey of fraud detection techniques. *Proc. IEEE Internat. Conf. on Networking, Sensing and Control*, vol. 2 (IEEE, Piscataway, NJ), 749–754.

LeCun Y, Bengio Y (1995) Convolutional networks for images, speech, and time series. *Handbook Brain Theory Neural Networks* 3361(10):1995.

Li M, Zhang T, Chen Y, Smola AJ (2014) Efficient mini-batch training for stochastic optimization. *Proc. 20th ACM SIGKDD Internat. Conf. on Knowledge Discovery and Data Mining* (ACM, New York), 661–670.

Li S, Xiao S, Zhu S, Du N, Xie Y, Song L (2018) Learning temporal point processes via reinforcement learning. *Proc. 32nd Internat. Conf. on Neural Information Processing Systems* (Curran Associates, Red Hook, NY), 10804–10814.

Li S, Xie Y, Farajtabar M, Verma A, Song L (2017) Detecting changes in dynamic events over networks. *IEEE Trans. Signal Inform. Processing Networks* 3(2):346–359.

Luca S, Karsmakers P, Cuppens K, Croonenborghs T, de Vel AV, Ceulemans B, Lagae L, et al. (2014) Detecting rare events using extreme value statistics applied to epileptic convulsions in children. *Artificial Intelligence Medicine* 60(2):89–96.

Luo W, Liu W, Gao S (2017) Remembering history with convolutional LSTM for anomaly detection. *Proc. IEEE Internat. Conf. on Multimedia and Expo* (IEEE, Piscataway, NJ), 439–444.

Maes S, Tuyls K, Vanschoenwinkel B, Manderick B (2002) Credit card fraud detection using Bayesian and neural networks. *Proc. 1st Internat. NAISO Congress on Neuro Fuzzy Technologies*, 261–270.

Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G (2016) LSTM-based encoder-decoder for multi-sensor anomaly detection. Preprint, submitted July 1, https://arxiv.org/abs/1607.00148.

Malini N, Pushpa M (2017) Analysis on credit card fraud identification techniques based on knn and outlier detection. *Proc. 3rd Internat. Conf. on Advances in Electrical, Electronics, Inform., Comm. and Bio-Informatics* (IEEE, Piscataway, NJ), 255–258.

Mei H, Eisner J (2017) The neural Hawkes process: A neurally self-modulating multivariate point process. *Proc. 31st Internat. Conf. on Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 6757–6767.

Michael S, George K, Vipin K (2002) A comparison of document clustering techniques. *Proc. KDD Workshop on Text Mining*.

Mohri M, Rostamizadeh A, Talwalkar A (2012) *Foundations of Machine Learning* (MIT Press, Cambridge, MA).

Nanduri A, Sherry L (2016) Anomaly detection in aircraft data using recurrent neural networks (RNN). *Proc. Integrated Communications Navigation and Surveillance* (IEEE, Piscataway, NJ), 5C2-1–5C2-8.

Ng AY, Russell S (2000) Algorithms for inverse reinforcement learning. *Proc. 17th Internat. Conf. on Machine Learn.* (Morgan Kaufmann, Burlington, MA), 663–670.

Oh Mh, Iyengar G (2019) Sequential anomaly detection using inverse reinforcement learning. *Proc. 25th ACM SIGKDD Internat. Conf. on Knowledge Discovery & Data Mining* (ACM, New York), 1480–1490.

Page ES (1954) Continuous inspection schemes. *Biometrika* 41(1/2):100–115.

Rahimi A, Recht B (2008) Random features for large-scale kernel machines. Platt JC, Koller D, Singer Y, Roweis ST, eds. *Advances in Neural Information Processing Systems*, vol. 20 (Curran Associates, Red Hook, NY), 1177–1184.

Rambaldi M, Filimonov V, Lillo F (2018) Detection of intensity bursts using Hawkes processes: An application to high frequency financial data. *J. Phys. Rev. E* 97(3):032318.

Reinhart A (2018) A review of self-exciting spatio-temporal point processes and their applications. *J. Statist. Sci.* 33(3):299–318.

Rudin W (1962) *Fourier Analysis on Groups*, vol. 121967 (Wiley Online Library, New York).

Ruff L, Vandermeulen R, Goernitz N, Deecke L, Siddiqui SA, Binder A, Müller E, Kloft M (2018) Deep one-class classification. Dy J, Krause A, eds. *Proc. 35th Internat. Conf. on Machine Learn.*, vol. 80 (PMLR), 4393–4402.

Sahin Y, Duman E (2011) Detecting credit card fraud by ANN and logistic regression. *Proc. Internat. Sympos. on Innovations in Intelligent Systems and Appl.* (IEEE, Piscataway, NJ), 315–319.

Siegmund D (1985) *Sequential Analysis: Tests and Confidence Intervals.* Springer Series in Statistics (Springer, Berlin).

Srivastava A, Kundu A, Sural S, Majumdar A (2008) Credit card fraud detection using hidden Markov model. *IEEE Trans. Dependent Secure Comput.* 5(1):37–48.

Syed U, Schapire RE (2007) A game-theoretic approach to apprenticeship learning. *Proc. 20th Internat. Conf. on Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 1449–1456.

The Nilson Report (2019) Payment card fraud losses reach $27.85 billion. https://nilsonreport.com/mention/407/1link/.

Tran PH, Tran KP, Huong TT, Heuchenne C, HienTran P, Le TMH (2018) Real time data-driven approaches for credit card fraud detection. *Proc. Internat. Conf. on E-Bus. and Appl.* (ACM, New York), 6–9.

Upadhyay U, De A, Gomez-Rodriguez M (2018) Deep reinforcement learning of marked temporal point processes. *Proc. 32nd Internat. Conf. on Neural Inform. Processing Systems (NIPS'18)* (Curran Associates, Red Hook, NY), 3172–3182.

Wang T, Rudin C, Wagner D, Sevieri R (2015) Finding patterns with a rotten core: Data mining for crime series with cores. *Big Data* 3(1):3–21.

Wang Y, Xu W (2018) Leveraging deep learning with LDA-based text analytics to detect automobile insurance fraud. *Decision Support Systems* 105:87–95.

Xiao S, Yan J, Yang X, Zha H, Chu SM (2017a) Modeling the intensity function of point process via recurrent neural networks. *Proc. 31st AAAI Conf. on Artificial Intelligence* (AAAI Press, Palo Alto, CA), 1597–1603.

Xiao S, Farajtabar M, Ye X, Yan J, Song L, Zha H (2017b) Wasserstein learning of deep generative point process models. *Proc. 31st Internat. Conf. on Neural Inform. Processing Systems* (Curran Associates, Red Hook, NY), 3250–3259.

Xu X (2010) Sequential anomaly detection based on temporal-difference learning: Principles, models and case studies. *Appl. Soft Comput.* 10(3):859–867.

Zhang R, Zhang S, Muthuraman S, Jiang J (2007) One class support vector machine for anomaly detection in the communication network performance data. *Proc. 5th Conf. on Appl. Electromagnetics, Wireless and Optical Comm* (WSEAS, Stevens Point, WI), 31–37.

Zhu S, Xie Y (2018) Crime incidents embedding using restricted Boltzmann machines. *Proc. IEEE Internat. Conf. on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ), 2376–2380.

Zhu S, Xie Y (2019a) Crime event embedding with unsupervised feature selection. *Proc. ICASSP IEEE Internat. Conf. on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ), 3922–3926.

Zhu S, Xie Y (2019b) Spatial-temporal-textual point processes for crime linkage detection. Preprint, submitted February 1, https://arxiv.org/abs/1902.00440.

Zhu S, Yuchi HS, Xie Y (2020) Adversarial anomaly detection for marked spatio-temporal streaming data. *Proc. ICASSP IEEE Internat. Conf. on Acoustics, Speech and Signal Processing* (IEEE, Piscataway, NJ), 8921–8925.

Zhu S, Li S, Peng Z, Xie Y (2021a) Imitation learning of neural spatio-temporal point processes. *IEEE Trans. Knowledge Data Engrg.* 34(11):5391–5402.

Zhu S, Zhang M, Ding R, Xie Y (2021b) Deep Fourier kernel for self-attentive point processes. Banerjee A, Fukumizu K, eds. *Proc. 24th Internat. Conf. on Artificial Intelligence and Statist.*, vol. 130 (PLMR), 856–864.

Zhu S, Wang H, Dong Z, Cheng X, Xie Y (2021c) Neural spectral marked point processes. Preprint, submitted June 20, https://arxiv.org/abs/2106.10773.

Ziebart BD, Maas A, Bagnell JA, Dey AK (2008) Maximum entropy inverse reinforcement learning. *Proc. 23rd National Conf. on Artificial Intelligence*, vol. 3 (AAAI Press, Palo Alto, CA), 1433–1438.