Online Anomaly Detection With Minimax Optimal Density Estimation in Nonstationary Environments

Kaan Gokcesu and Suleyman S. Kozat, *Senior Member, IEEE*

***Abstract*—We introduce a truly online anomaly detection algorithm that sequentially processes data to detect anomalies in time series. In anomaly detection, while the anomalous data are arbitrary, the normal data have similarities and generally conforms to a particular model. However, the particular model that generates the normal data is generally unknown (even nonstationary) and needs to be learned sequentially. Therefore, a two stage approach is needed, where in the first stage, we construct a probability density function to model the normal data in the time series. Then, in the second stage, we threshold the density estimation of the newly observed data to detect anomalies. We approach this problem from an information theoretic perspective and propose minimax optimal schemes for both stages to create an optimal anomaly detection algorithm in a strong deterministic sense. To this end, for the first stage, we introduce a completely online density estimation algorithm that is minimax optimal with respect to the log-loss and achieves Merhav's lower bound for general nonstationary exponential-family of distributions without any assumptions on the observation sequence. For the second stage, we propose a threshold selection scheme that is minimax optimal (with logarithmic performance bounds) against the best threshold chosen in hindsight with respect to the surrogate logistic loss. Apart from the regret bounds, through synthetic and real life experiments, we demonstrate substantial performance gains with respect to the state-of-the-art density estimation based anomaly detection algorithms in the literature.**

***Index Terms*—Anomaly detection, time series, online learning, density estimation, minimax optimal.**

I. Introduction

A. Preliminaries

We study anomaly detection [1], which has attracted significant attention in recent years due to its applications in network monitoring [2], cybersecurity [3], surveillance [4] and sensor failure [5]. Particularly, we study the sequential anomaly detection problem, where at each time t, we sequentially observe a vector $\boldsymbol{x}_t \in \mathcal{X}$ such that $\mathcal{X} \subset \mathbb{R}^m$ and our aim is to decide whether this new observation is anomalous or not, based on the past observations $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{t-1}\}$.

In the anomaly detection problem, "the normal data" displays similarities and generally conforms to a particular model even though the anomalous data may be arbitrary and may not show any similarities. However, the particular model that generates "the normal data" is generally unknown (may even change throughout time) and needs to be learned sequentially from the incoming data. Therefore, a two stage approach is needed, where in the first stage, a model of "the normal data" is constructed and in the second stage a decision is made based on the model and the observed data. A common approach is to find a typical set containing the most likely instances in an unknown measure of probability, where "the normal data" is commonly assumed to be generated from an independent and identically distributed (i.i.d.) random variable sequence belonging to this unknown measure [1]. In the density level set estimation framework [6], the theoretically optimal nominal set would consist of the samples whose probability under the unknown measure is greater than some density level set parameter that represents the fraction of outliers in the model [6]. Hence, in the first stage of our algorithm we train a density estimator for "the normal data."

However, in general, the observations $\boldsymbol{x}_t$ are prone to distortion (because of a less than ideal communication channel and noise contamination) and may not be the outputs of a stochastic process yet alone independent and identically distributed since the environment possibly exhibits nonstationary behavior and may even be chaotic or adversarial [7]. Therefore, to solve this otherwise difficult modeling of the nominal data, we approach the density estimation problem from a competitive algorithm perspective [8]–[12] and design algorithms that perform as well as the best density estimator chosen in hindsight (with possibly changing statistics) for any observation sequence. We show that our approach has strong performance guarantees in an individual sequence manner [12]. Thus, our algorithm is able to perform as well as the optimal nonstationary density in our competition class even if there is no underlying stochastic process. Since these algorithms work in an individual sequence manner, they are robust against distorted and dependent observations generated in a dynamically evolving environment as opposed to the explicit design and modeling of an evolving system [13], [14].

Even though there exist several nonparametric approaches to model the distribution of the nominal data [15]–[17], parametric models are usually more practical because of their faster learning behavior and high modeling powers [16]. The parametric models can suffer if only the assumed model is incapable of

Manuscript received February 8, 2017; revised July 7, 2017, September 10, 2017, October 20, 2017, and November 16, 2017; accepted November 24, 2017. Date of publication December 18, 2017; date of current version January 26, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jorge F. Silva. This work was supported in part by the Turkish Academy of Sciences Outstanding Researcher Programme and the Scientific and Technological Research Council of Turkey under Contract 113E517. *(Corresponding author: Kaan Gokcesu.)*

The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: gokcesu@ee.bilkent.edu.tr; kozat@ee.bilkent.edu.tr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2784390

continuously modeling the data [1] in a robust manner. Therefore, we model the source that generates the nominal data as an exponential-family distribution [18], [19], since the exponential-family of distributions cover a wide range of parametric statistical models [7] and accurately approximates many nonparametric classes of probability densities [20]. By using online convex programming [21] to estimate the natural parameter of this exponential-family distribution, we can achieve logarithmic regret bounds under log-loss, i.e., $O(\log T)$ (minimax optimal) [7], [22]. However, most real life applications such as in cybersecurity [3] or surveillance applications [4], the underlying data stream is nearly always nonstationary [1], [23], i.e., has varying statistics over time [24]. Hence, we model the nonstationary source model (i.e., the changing natural parameter) as piecewise stationary models. We emphasize that there are no restrictions on the number of regions or the length of these regions that are needed for accurate modeling. By combining these stationary models in a switching experts setting [25]–[28], we achieve the minimax optimal performance bound $O(C \log T)$, when the statistics of the source, i.e., the natural parameter, changes $C - 1$ number of times, i.e., the nonstationary model consists of C stationary models (C time segments with piecewise constant parameters), which constitutes our main contribution.

After the minimax optimal modeling of the distribution of the nominal data in the time series, we produce our anomaly decision by thresholding the density estimation, which optimally minimizes Type-1 errors in certain environments [15], [29]. Such anomaly detection methods with two stages, i.e., scoring the time series samples and thresholding them, are extensively studied in the literature [1], [7], [30]. Even though the detection of anomalous samples when their likelihood (probability or score) falls below a certain threshold is an efficient and popular strategy, the selection of this threshold is a notoriously challenging problem [7]. Therefore, we implement a dynamic thresholding scheme, which updates the threshold whenever a feedback about the observed sample is available (i.e., whether it is anomalous or not). We update our threshold using Online Newton Step [22] and achieve again logarithmic performance bounds (minimax optimal surrogate regret) with respect to the best threshold selected in hindsight.

Even though there exists various methods to detect anomalies in a time series, we, for the first time in literature, propose a truly minimax optimal (both in density estimation and threshold selection) online anomaly detection algorithm in nonstationary settings. Through synthetic and real-life experiments, we demonstrate significant performance gains with respect to the state-of-the-art methods in the literature [7], [21], [29], [31].

B. Prior Art and Comparisons

Various anomaly detection methods have been proposed in the literature that utilizes Support Vector Machines (SVM) [32], [33], nearest neighbors approach [34], [35], clustering [36] and density estimation [37], [38]. However, techniques based on probability density estimation are demonstrated to provide superior performance when "the normal data" conforms to a nominal distribution and the unknown probability measure can be estimated near perfect [39]. Moreover, even though there exist several anomaly detection methods [1] proposed for supervised [40], semi-supervised [41] and unsupervised [6] settings, none of them has performance bounds in nonstationary or adversarial settings [7].

For these reasons, we adopt the probability density based approach in an individual sequence perspective [12] to provide strong deterministic bounds both in density estimation and anomaly detection stages of our algorithm. In the literature, there are various methods that can achieve sublinear performance bounds when estimating the density in nonstationary environments. In [21], authors propose an approach that can achieve a regret bound of $O(\sqrt{CT})$ when the time horizon T and the total change C is known a priori. Without the prior knowledge of T, the algorithm in [21] can still achieve $O(\sqrt{CT})$ regret if it is run in accordance with the doubling trick [11]. One can also modify the algorithm of [21] to achieve a regret bound of $O(\sqrt{C_{\max}T})$ if an upper bound on C is known instead such that $C_{\max} \geq C$. For the case of no prior knowledge about C, an algorithm that achieves a regret bound of $O(C\sqrt{T})$ is proposed in [7]. Nonetheless, none of these methods achieve the minimax regret bound $O(C \log T)$ [42]. In [27] and [28], the authors propose methods to achieve this minimax optimal regret only in binary sources and discrete sources respectively. Achieving the minimax optimal regret $O(C \log T)$ is not possible with the state-of-the-art methods when the source belongs to a general exponential-family. To this end, we introduce a truly online density estimation algorithm that can achieve the minimax optimal regret bound $O(C \log T)$ without any knowledge of C and T beforehand. The computational complexity and storage demand of our algorithm is linear in time.

In anomaly detection with thresholding some likelihood function (such as in the probability density based methods), the optimal selection of the threshold is intrinsically difficult [7]. Therefore, we adopt a dynamically selected thresholding scheme and update our threshold in accordance with the feedback. Most of the algorithms in literature do not provide guaranteed regret bounds in this setting. In the literature [7], performance bounds of only $O(\sqrt{T})$ for surrogate regret are achieved with respect to the best threshold selected in hindsight. However, such a regret bound for the anomaly detection performance invalidates our purpose of estimating the density function of the normal data with minimax optimal, i.e., log-linear, ($O(C \log T)$) regret. Therefore, we propose a thresholding scheme that achieves logarithmic (minimax) regret bound, i.e., $O(\log T)$, to provide a truly minimax optimal anomaly detection algorithm.

C. Contributions

Our contributions are as follows:

1) For the first time in the literature, we propose a truly minimax optimal anomaly detection algorithm for nonstationary sources with logarithmic performance bounds by optimizing both the density estimation and threshold selection in an individual sequence manner.
2) Our algorithm can be used in unsupervised, semi-supervised and supervised manner because of its

individual sequence (i.e., universal prediction) perspective on its density estimation.

3) Our algorithm can assign distinct costs to making a prediction error on normal and anomalous data since in general their importance are not the same in various applications.
4) For the first time in literature, we propose a density estimation algorithm that achieves the minimax optimal regret $O(C \log T)$ for general exponential-family of sources. Our density estimation algorithm improves upon the source coding literature and asymptotically achieves Merhav's lower bound [42] for any i.i.d. exponential-family source with piecewise constant parameters.
5) Our proposed adaptive thresholding scheme achieves log-linear regret against the best threshold chosen in hindsight and improves greatly upon the $\sqrt{T}$ regret scheme in the literature [7].
6) Our algorithm is strongly sequential such that neither the time horizon T nor the number of changes C of the source statistics are known.
7) Through extensive set of experiments involving synthetic and real datasets, we demonstrate significant performance gains achieved by the proposed algorithm for both density estimation and anomaly detection with respect to the state-of-the-art methods.

D. Organization

First, we formally define our problem setting in Section II. We introduce a minimax optimal density estimator that achieves logarithmic regret bound for stationary memoryless sources in Section III. In Section IV, we eliminate the requirement of any a priori knowledge of the statistics of the source. In Section V, we introduce a minimax optimal density estimator for nonstationary sources. In Section VI, we propose a completely online anomaly detection algorithm that adaptively thresholds the density estimation. In Section VII, we illustrate significant performance gains over both real and synthetic data, and finish with concluding remarks in Section VIII.

II. Problem Description

We introduce an online algorithm for anomaly detection in time series that sequentially observes $\{\boldsymbol{x}_t\}_{t\geq 1}$, $\boldsymbol{x}_t \in \mathbb{R}^m$, at each time t, and estimates the probability of the unseen data based on our previous observations. Based on this probability estimate, we decide whether the newly observed data is anomalous or not by comparing the estimated probability with a threshold [1], [15].

We assume that the normal samples of the time series are generated by or can be closely modeled with a nonstationary source with piecewise constant parameters. We can represent such a density function as $f_t(\cdot)$ whose source parameters are given by the vector $\boldsymbol{\alpha}_t$. Since the source at hand has piecewise constant parameters, the source parameters $\boldsymbol{\alpha}_t$ remains unchanged for some segment of time indices, i.e., $\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t+1}$, if both t and $t+1$ are in the same time segment where the source statistics do not change. We represent the number of changes of the source statistics in a T length observation sequence by C which is given by

$$C \triangleq 1 + \sum_{t=2}^{T} \mathbb{1}_{\boldsymbol{\alpha}_t \neq \boldsymbol{\alpha}_{t-1}}, \tag{1}$$

where the beginning is also counted as a change and $\mathbb{1}_x$ is the indicator function that outputs 1 if the statement x is true and 0 otherwise. As an example, for stationary sources, i.e., distributions with unchanging natural parameter, the number of changes C is 1.

In our two stage algorithm, we first introduce a pdf estimation algorithm, which estimates the density of the nonstationary source that generates or closely models the normal data in the time series. In the online setting, we observe a sample vector $\boldsymbol{x}_t$ at each time t with the unknown density function $f_t(\cdot)$. We estimate this unknown density $f_t(\boldsymbol{x}_t)$ based on our past observations $\{\boldsymbol{x}_r\}_{r=1}^{t-1}$ and produce an estimate $\hat{f}_t(\boldsymbol{x}_t)$. We approach this problem of estimating the density of the nonstationary source from a competitive algorithm perspective where the competing strategy is naturally given by the true density function (if such a distribution exists) or the density function that best represents the data. To this end, as the performance measure (i.e., the loss function in our competitive framework), we use the log-loss of the density functions, i.e., $l(\hat{f}_t(\boldsymbol{x}_t)) = -\log(\hat{f}_t(\boldsymbol{x}_t))$, since it is the most widely used loss function for probability distributions [43]. We use the notion of "regret" on the log-losses to define our performance in an individual sequence manner [11], such that the regret at time t is

$$r_t = -\log(\hat{f}_t(\boldsymbol{x}_t)) + \log(f_t(\boldsymbol{x}_t)), \tag{2}$$

and the cumulative regret up to time T is

$$R_T = \sum_{t=1}^{T} \Big(-\log(\hat{f}_t(\boldsymbol{x}_t)) + \log(f_t(\boldsymbol{x}_t)) \Big). \tag{3}$$

Our goal is to achieve low regret bounds for any arbitrary observation sequence. We do not analyze the regret with its statistical properties (e.g., the expected regret, its variance or high probability bounds) whether or not the observation sequence is stochastic in nature. Instead, we will derive regret bounds for any individual observation sequence in a universal prediction perspective [12]. Our algorithms are deterministic and our accumulated regret will not change given the same observation sequence (hence, there is no randomness involved for any arbitrary observation).

We want to achieve the performance of the best distribution with piecewise constant parameters where each segment is modeled by an exponential-family distribution, since the exponential-family of distributions cover a wide range of parametric statistical models [7] and accurately approximates many nonparametric classes of probability densities [20]. For exponential-family of sources, the source statistics $\boldsymbol{\alpha}_t$ is called the natural parameter of the density function. Hence, the density function $f_t(\cdot)$ is given by an exponential-family of distribution with possible changing natural parameter $\boldsymbol{\alpha}_t$.

After modeling the normal data in the time series by creating a minimax optimal density estimation $\hat{f}_t(\boldsymbol{x}_t)$, we produce our anomaly decision by thresholding $\hat{f}_t(\boldsymbol{x}_t)$, which optimally minimizes Type-1 errors in certain environments [15], [29]. Hence, the binary anomaly decision $\hat{d}_t$ is constructed as

$$\hat{d}_t = \begin{cases} +1, & \hat{f}_t(\boldsymbol{x}_t) < \tau_t \text{ (anomalous)} \\ -1, & \hat{f}_t(\boldsymbol{x}_t) \geq \tau_t \text{ (not anomalous)} \end{cases}, \tag{4}$$

where τ_t is a time varying threshold. Our true $0-1$ loss function in this setting is defined as

$$l_A(\hat{f}_t(\boldsymbol{x}_t), \tau, d_t) = \mathbb{1}_{\text{sign}(\tau - \hat{f}_t(\boldsymbol{x}_t)) \neq d_t}, \tag{5}$$

where τ is the threshold variable.

However, the loss in (5) is difficult to analyze since the loss function is not convex in τ. To deal with this difficulty, we do a convex relaxation. We use the standard practice of replacing the comparator function with a convex surrogate function (common examples are the square loss, hinge loss, cross entropy and logistic loss) [44]. We use the logistic loss function,

$$l(\tau_t, \hat{f}_t(\boldsymbol{x}_t), d_t) = \log(1 + \exp(-(\tau_t - \hat{f}_t(\boldsymbol{x}_t))d_t)),$$

which is also exp-concave. We compete against the best threshold chosen in hindsight τ^*. Thus, the excess cost (regret) incurred up to time T by our anomaly detector with respect to the best threshold chosen in hindsight is defined as

$$R_{A,T} = \sum_{t=1}^{T} l(\tau_t, \hat{f}_t(\boldsymbol{x}_t), d_t) - \sum_{t=1}^{T} l(\tau^*, \hat{f}_t(\boldsymbol{x}_t), d_t). \tag{6}$$

Performance analysis for these surrogate loss functions are meaningful because they completely upper bound the original $0-1$ loss function, hence, a performance guarantee for the surrogate regret is likely to hold for the original regret as well.

Our aim is to achieve minimax optimal regret bounds for both (3) and (6). To this end, in Section III, we introduce a minimax optimal density estimator that achieves logarithmic regret bound for stationary memoryless sources. In Section IV, we use these density estimators as our building blocks to eliminate the requirement of any a priori knowledge of the statistics of the source. In Section V, we introduce a minimax optimal density estimator for nonstationary sources and achieve the minimax optimal regret bound $O(C \log T)$ for (3). In Section VI, we propose a completely online anomaly detection algorithm that adaptively thresholds the density estimation, which achieves the minimax optimal regret bound for (6), i.e., $O(\log T)$ regret against the best fixed threshold selected in hindsight.

III. Stationary Density Estimator

In this section, we first construct a density estimator that achieves minimax regret bound for stationary sources (unchanging statistics, hence, natural parameter). Here, at each time t, we observe $\boldsymbol{x}_t \in \mathbb{R}^m$ distributed or closely modeled according to a memoryless exponential-family distribution

$$f(\boldsymbol{x}_t) = \exp\left(-\langle \boldsymbol{\alpha}, \boldsymbol{z}_t \rangle - A(\boldsymbol{\alpha})\right), \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^d$ is the unknown natural parameter of the exponential-family distribution that minimizes the cumulative log-loss (i.e., the best offline estimation) and belongs to a convex feasible set S, $A(\cdot)$ is a known function of the parameter $\boldsymbol{\alpha}$ (normalization factor or log-partition function), $\langle \cdot, \cdot \rangle$ is the inner product and $\boldsymbol{z}_t$ is the d-dimensional known sufficient statistic of $\boldsymbol{x}_t$ [18], i.e.,

$$\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t). \tag{8}$$

Remark 1: In general, a distribution belonging to an exponential family has the form $f(x) = \exp(-\langle \boldsymbol{\alpha}, \mathcal{T}(x) \rangle - A(\boldsymbol{\alpha}) - B(x))$, where $B(x)$ is only a function of the observation x. However, this function can simply be included inside of $\mathcal{T}(x)$ whose corresponding parameter in the inner product will simply be 1 in the true probability density.

Instead of directly estimating the density function $f(\cdot)$, we cast the problem as a convex optimization problem and estimate the natural parameter $\boldsymbol{\alpha}$ according to our past observations $\{\boldsymbol{x}_r\}_{r=1}^{t-1}$ ($\boldsymbol{\alpha}$ completely describes $f(\cdot)$ and estimating $\boldsymbol{\alpha}$ instead of $f(\cdot)$ still provides logarithmic performance bounds). Hence, our density estimation is

$$\hat{f}_t(\boldsymbol{x}_t) = \exp(-\langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle - A(\hat{\boldsymbol{\alpha}}_t)). \tag{9}$$

We use Online Gradient Descent (OGD) [22] to sequentially produce our estimate $\hat{\boldsymbol{\alpha}}_t$, where we first start from an initial estimate $\hat{\boldsymbol{\alpha}}_1$, and update our recent estimation $\hat{\boldsymbol{\alpha}}_t$ based on our new observation $\boldsymbol{x}_t$. To update $\hat{\boldsymbol{\alpha}}_t$, we first observe a sample $\boldsymbol{x}_t$ and incur the loss $l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)$ according to our estimation $\hat{\boldsymbol{\alpha}}_t$, which is the log-loss, i.e., $-\log(\hat{f}_t(\boldsymbol{x}_t))$. From (9), we receive the loss

$$l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) = \langle \hat{\boldsymbol{\alpha}}_t, \boldsymbol{z}_t \rangle + A(\hat{\boldsymbol{\alpha}}_t). \tag{10}$$

Then, we calculate the gradient of the loss with respect to $\hat{\boldsymbol{\alpha}}_t$,

$$\begin{aligned} \nabla_{\hat{\boldsymbol{\alpha}}_t} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) &= \boldsymbol{z}_t + \nabla_{\hat{\boldsymbol{\alpha}}_t} A(\hat{\boldsymbol{\alpha}}_t), \\ &= \boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}, \end{aligned} \tag{11}$$

where $\mu_{\hat{\boldsymbol{\alpha}}_t}$ is the mean of $\boldsymbol{z}_t$ if $\boldsymbol{x}_t$ were distributed according to $\hat{f}_t(\cdot)$. We update the parameter $\hat{\boldsymbol{\alpha}}_t$ such that

$$\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\hat{\boldsymbol{\alpha}}_t - \eta_t(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})), \tag{12}$$

where η_t is the learning rate and $P_S(\cdot)$ is the projection onto the convex feasible set S and is defined as

$$P_S(x) = \arg\min_{y \in S} \|x - y\|. \tag{13}$$

The complete algorithm is provided in Algorithm 1 where the learning rates are given by $\eta_t = (Ht)^{-1}$ where $H > 0$ is an input to the algorithm. The optimal value for H and the instructions on its selection are given in Theorem 1. Next, we provide performance bounds of Algorithm 1. Theorem 1 shows that using Algorithm 1 with a suitable parameter H, we can achieve minimax regret $O(\log T)$.

Theorem 1: When Algorithm 1 is used with parameter H to estimate the distribution $f_t(\boldsymbol{x}_t)$, its regret is upper bounded by

$$R_T \leq \frac{D}{2H}(\log T + 1), \tag{14}$$

if H is such that $\Sigma_{\hat{\boldsymbol{\alpha}}_t} \succeq H I_{d \times d}$ for all t, where $\Sigma_{\hat{\boldsymbol{\alpha}}_t}$ is the covariance of $\boldsymbol{z}_t$ when $\boldsymbol{x}_t$ is distributed with natural parameter

Algorithm 1: Stationary Density Estimator.

```
Set $H$
Initialize learning rates $\eta_t = (Ht)^{-1}$ for $t \in \{1, 2, \ldots\}$
Select initial parameter $\hat{\boldsymbol{\alpha}}_1$
Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_1}$
for $t = 1, 2, \ldots$ do
    Declare estimation $\hat{\boldsymbol{\alpha}}_t$
    Observe $\boldsymbol{x}_t$
    Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
    Update parameter: $\tilde{\boldsymbol{\alpha}}_{t+1} = \hat{\boldsymbol{\alpha}}_t - \eta_t(\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t})$
    Project onto convex set: $\hat{\boldsymbol{\alpha}}_{t+1} = P_S(\tilde{\boldsymbol{\alpha}}_{t+1})$
    Calculate the mean $\mu_{\hat{\boldsymbol{\alpha}}_{t+1}}$
end for
```

$\hat{\boldsymbol{\alpha}}_t$, $I_{d \times d}$ is the d-by-d identity matrix and D is defined as

$$D \triangleq \frac{\sum_{t=1}^{T} t^{-1} \|\boldsymbol{z}_t - \mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{\sum_{t=1}^{T} t^{-1}}.$$

The result of Theorem 1 shows that the regret is reciprocally dependent on our input parameter H. Therefore, choosing H lower than necessary may result in a high regret. In Section IV, we mitigate this problem by adopting a mixture of experts approach [45], [46].

Proof of Theorem 1: The proof of the theorem is given in Appendix A. ■

Remark 2: Suppose instead of $\boldsymbol{x}_t$, we observe a distorted version such that $\boldsymbol{y}_t = Q(\boldsymbol{x}_t)$, where $Q(\cdot)$ is the distortion channel, e.g., an additive noise channel. Then, using an unbiased estimator $\bar{\boldsymbol{z}}_t = \bar{\mathcal{T}}(\boldsymbol{y}_t)$ such that $\mathrm{IE}[\bar{\boldsymbol{z}}_t] = \mathcal{T}(\boldsymbol{x}_t)$ produces the same results in Theorem 1 for expected regret [7].

Remark 3: Instead of OGD, any other gradient based approach such as Momentum [47], Nesterov's Accelerated Gradient [48], Adam [49], Adagrad [50], Adadelta [51] can also be used. Similar derivations will also hold for them as well.

IV. Universal Density Estimator

The Algorithm in Section III needs an input H which needs a priori knowledge we do not have on the underlying process to be set appropriately. Therefore, in this section, we propose an algorithm to estimate the density of a stationary source when we do not know H a priori.

Suppose, we run N separate Algorithm 1 and label each of the resulting estimators as $\hat{f}_t^r(x), r \in \{1, 2, \ldots, N\}$. We mix all of the estimators in a weighted manner such that our density estimation is given by

$$\hat{f}_t^u(x) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x), \tag{15}$$

where w_t^r is the mixture weight of the algorithm labeled r at time t. These weights are normalized versions of the algorithms' performance weights $\hat{w}_t^r$ such that

$$w_t^r = \frac{\hat{w}_t^r}{\sum_{r'=1}^{N} \hat{w}_t^{r'}}. \tag{16}$$

The performance weights of the algorithms are given by

$$\hat{w}_t^r = \prod_{\tau=1}^{t-1} \hat{f}_\tau^r(\boldsymbol{x}_\tau), \tag{17}$$

for $t > 1$ and $\hat{w}_1^r = 1$ at the start for $r \in \{1, 2, \ldots, N\}$. We point out that the mixture weights w_t^r can be recursively updated as

$$w_{t+1}^r = w_t^r \hat{f}_t^r / \hat{f}_t^u, \tag{18}$$

from (15), (16), (17), where $w_1^r = 1/N$. We next provide the performance bound of this mixture approach.

Theorem 2: When we combine a total of N estimators with the weighting scheme in (15) and (18), the regret incurred by our universal density estimator ($\hat{f}_t^u(\cdot)$), $R_{T,U}$, is upper bounded by

$$R_{T,U} \leq \log N + \min_{r \in \{1,\ldots,N\}} R_{T,r},$$

where $R_{T,r}$ is the regret incurred by the estimator labeled r.

Theorem 2 implies that even by mixing exponential number of estimators we can still achieve sublinear regret since the additional regret of mixing is only logarithmically dependent on the number of density estimators mixed together.

Proof of Theorem 2: The proof of the theorem is given in Appendix B. ■

Corollary 1: Suppose we do not know H exactly but know $H_{\min}$ and $H_{\max}$ such that $H_{\min} \leq H \leq H_{\max}$. For each of the estimators mixed, i.e., $\hat{f}_t^r(\cdot)$, we set the input parameter as $H_r = H_{\min} \cdot 2^{r-1}$ for $r = \{1, 2, \ldots, \lfloor \log_2 \frac{H_{\max}}{H_{\min}} \rfloor + 1\}$. We incur $R_{T,U}$ such that,

$$R_{T,U} \leq \log\left(\left\lfloor \log_2 \frac{H_{\max}}{H_{\min}} \right\rfloor + 1\right) + \frac{D}{H}(\log T + 1).$$

Proof of Corollary 1: The proof of the corollary is given in Appendix C. ■

The complete algorithm is given in Algorithm 2.

Remark 4: The result in Theorem 2 is general such that it is true for any density estimation used in the mixture. Therefore, we can incorporate various different density estimators (parametric or nonparametric, e.g., ML [29] and KDE [31]) as experts in Algorithm 2 to achieve the optimum performance in the mixture. For example, the construction of $\boldsymbol{z}_t$ requires the knowledge of sufficient statistics mapping $\mathcal{T}(\cdot)$ beforehand. Since the sufficient statistics of different kinds of distributions belonging to the exponential family may differ, knowing the mapping $\mathcal{T}(\cdot)$ means knowing the exact kind of distribution to a certain degree, e.g., whether the distribution is normal, exponential, gamma etc. Therefore, we can add to the mixture in Algorithm 2 different Algorithm 1's with different sufficient statistics mappings $\mathcal{T}'(\cdot)$. We point out that if the total number of possible candidates for $\mathcal{T}(\cdot)$ is M, our regret only increases by $\log M$.

V. Nonstationary Density Estimator

In Section IV, we have introduced an algorithm that achieves the minimax optimal regret when estimating a stationary exponential-family source without any a priori knowledge. In this section, we extend that result to nonstationary sources, i.e.,

Algorithm 2: Universal Density Estimator.

```
INPUTS:
Set $H_{\min}$ and $H_{\max}$
Set $N = \lfloor \log \frac{H_{\max}}{H_{\min}} \rfloor + 1$
Start $N$ density estimators $\hat{f}_t^r(\cdot)$ each running Algorithm 1
    with parameter $H_r = H_{\min} \cdot 2^{r-1}$ for $r = \{1, 2, \ldots, N\}$
Initialize weights $w_1^r = 1/N, \forall r$
ALGORITHM:
for $t = 1$ to $T$ do
    OUTPUT:
    Declare estimation $\hat{f}_t^u(x) = \sum_{r=1}^{N} w_t^r \hat{f}_t^r(x)$
    UPDATE:
    Observe $\boldsymbol{x}_t$
    Calculate $\boldsymbol{z}_t = \mathcal{T}(\boldsymbol{x}_t)$
    for $r = 1$ to $N$ do
        Update density estimations $\hat{f}_t^r()\cdot$ according to
        "Stationary Density Estimator"
        $w_{t+1}^r = w_t^r \hat{f}_t^r(x_t) / \hat{f}_t^u(x_t)$
    end for
end for
```

Algorithm 3: Nonstationary Density Estimator.

```
Initialize $\hat{v}_1^1 = 1$
Set $H_{\min}, H_{\max}$
Initialize estimator $\hat{g}_1^1(\cdot)$ using Algorithm 2
    with inputs $\{H_{\min}, H_{\max}\}$.
for $t = 1$ to $T$ do
    Declare estimation $\hat{g}_t^u(x) = \sum_{\tau=1}^{t} v_t^\tau \hat{g}_t^\tau(x)$
    Observe $\boldsymbol{x}_t$
    for $\tau = 1$ to $t$ do
        Update $\hat{v}_{t+1}^\tau = \frac{t+1-\tau}{t-\tau+2} \cdot \hat{v}_t^\tau \cdot \hat{g}_t^\tau(\boldsymbol{x}_t)$
        Update $\hat{g}_{t+1}^\tau(\cdot)$ according to Algorithm 2
    end for
    Create $\hat{v}_{t+1}^{t+1} = \sum_{\tau=1}^{t} \frac{1}{t-\tau+2} \cdot \hat{v}_t^\tau \cdot \hat{g}_t^\tau(\boldsymbol{x}_t)$
    Start estimator $\hat{g}_{t+1}^{t+1}(\cdot)$ according to Algorithm 2
    for $\tau = 1$ to $t+1$ do
        $v_t^\tau = \hat{v}_t^\tau / \left(\sum_{\tau=1}^{t} \hat{v}_t^\tau\right)$
    end for
end for
```

dynamic natural parameter $\boldsymbol{\alpha}_t$. To achieve the minimax regret in nonstationary sources, we can run Algorithm 2 on our observations and reset it at each instance the source statistics change. While this approach achieves the minimax optimal regret, it requires the knowledge of exact time instances the source statistics change. Since we do not know these time instances, we create a new Algorithm 2 at each time t and mix them instead of resetting the algorithm. We point out that these newly created Algorithm 2's train their estimators with only the observations after the time they were created. Hence, at each time τ, we create a new Algorithm 2 and label the estimation of this algorithm at time t ($\tau \leq t$) as $\hat{g}_t^\tau(\cdot)$. We point out that $\hat{g}_t^\tau(\cdot)$ only uses the observations $\{\boldsymbol{x}_\tau, \boldsymbol{x}_{\tau+1}, \boldsymbol{x}_{\tau+2}, \ldots, \boldsymbol{x}_{t-1}\}$ to construct its density estimation at time t. Since $\hat{g}_t^\tau(\cdot)$ is the output of Algorithm 2, it has already gone through a mixture. We adopt a double mixture approach and combine these $\hat{g}_t^\tau(\cdot)$ again with carefully selected weights as

$$\hat{g}_t^u(x) = \sum_{\tau=1}^{t} v_t^\tau \hat{g}_t^\tau(x), \tag{19}$$

where v_t^τ is the combination weight of the estimator started at τ at time t. These combination weights are normalized versions of the algorithms' performance weights $\hat{v}_t^\tau$ such that

$$v_t^\tau = \frac{\hat{v}_t^\tau}{\sum_{\tau=1}^{t} \hat{v}_t^\tau}, \tag{20}$$

for $1 \leq \tau \leq t$. The performance weights are recursively calculated as

$$\hat{v}_t^\tau = \begin{cases} \dfrac{t-\tau}{t-\tau+1}\, \hat{v}_{t-1}^\tau\, \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1}), & \tau < t \\ \displaystyle\sum_{\tau=1}^{t-1} \frac{1}{t-\tau+1}\, \hat{v}_{t-1}^\tau\, \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1}), & \tau = t \end{cases}, \tag{21}$$

where $\hat{v}_1^1 = 1$ at the start. The complete description of the algorithm is given in Algorithm 3. Next, we provide the performance bound of Algorithm 3.

Theorem 3: Algorithm 3 has the following regret bound, $R_{T,NS}$,

$$R_{T,NS} \leq \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i + 1) + \sum_{i=1}^{C} R_{i,U}, \tag{22}$$

where C is the total number of change in the source statistics as in (1) and the lengths of the piecewise stationary segments are t_i for $i \in \{1, 2, \ldots, C\}$ ($\sum_{i=1}^{C} t_i = T$). Initializing $t_0 = 0$, $R_{i,U}$ is the regret incurred by the estimator $\hat{g}_t^{\sum_{j=1}^{i-1} t_j + 1}$ in time interval $t \in [\sum_{j=1}^{i-1} t_j + 1, \sum_{j=1}^{i} t_j]$.

The result of Theorem 3 shows that the additional regret we incur from not knowing the time instances when the source statistics change is linearly dependent on the number of such changes.

Proof of Theorem 3: Combining (19) and (20), we have,

$$\hat{g}_t^u(x) = \frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(x)}{\hat{v}_t^t + \sum_{\tau=1}^{t-1} \hat{v}_t^\tau} = \frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(x)}{\sum_{\tau=1}^{t-1} \hat{v}_{t-1}^\tau \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1})},$$

for $t > 1$ and $\hat{g}_{t=1}^u(x) = \hat{g}_1^1(x)$. Thus, we receive the following accumulated log-loss, L_{NS},

$$\begin{aligned} L_{NS} &= \sum_{t=1}^{T} - \log \hat{g}_t^u(\boldsymbol{x}_t), \\ &= - \log \hat{g}_1^1(\boldsymbol{x}_1) - \sum_{t=2}^{T} \log \left(\frac{\sum_{\tau=1}^{t} \hat{v}_t^\tau \hat{g}_t^\tau(\boldsymbol{x}_t)}{\sum_{\tau=1}^{t-1} \hat{v}_{t-1}^\tau \hat{g}_{t-1}^\tau(\boldsymbol{x}_{t-1})} \right), \\ &= - \log \sum_{\tau=1}^{T} \hat{v}_T^\tau \hat{g}_T^\tau(\boldsymbol{x}_T). \end{aligned}$$

We see that this expression can be written as the log-loss of a weighted sum of losses incurred from various strategies belonging to a set S_T after we recursively substitute $\hat{v}_T^{\tau}$ using (21) and consider the distributive property of multiplication over addition. A strategy s in S_T, uses $\hat{g}_{t-1}^{\tau}(\cdot)$ at time $t-1$ and it is only allowed to continue along its last estimator to $\hat{g}_t^{\tau}(\cdot)$ or switch to $\hat{g}_t^{t}(\cdot)$ at time t. Let s_t denote the superscript (τ) of whichever estimator is utilized by s at time t. Consequently, our loss becomes,

$$\begin{aligned} L_{NS} &= -\log \sum_{s \in S_T} \left(Q(s) \prod_{t=1}^{T} \hat{g}_t^{s_t}(\boldsymbol{x}_t) \right), \\ &\leq -\log Q(s) \prod_{t=1}^{T} \hat{g}_t^{s_t}(\boldsymbol{x}_t), \end{aligned} \tag{23}$$

for all $s \in S_T$, where $Q(s)$ denotes the prior weight determined by how long each of the different estimators $\hat{g}_t^{\tau}(\cdot)$'s are used by s.

In (21), the prior components $\frac{t-\tau}{t-\tau+1}$ and $\frac{1}{t-\tau+1}$ in the calculation of $\hat{v}_t^{\tau}$ result in the following $Q(s)$ for an s which utilizes a total of C_s estimators

$$Q(s) = \begin{cases} \dfrac{1}{T}, & C_s = 1 \\ \displaystyle\prod_{i=1}^{C_s} \frac{1}{t_i} \prod_{i=1}^{C_s-1} \frac{1}{t_i+1}, & C_s > 1 \end{cases},$$

where t_i's are the durations in which fixed $\hat{g}_t^{\tau}(\cdot)$'s are used, i.e., the same estimator that started at τ. To upper-bound L_{NS}, we use $s^* \in S_T$ in (23), which changes its estimator at the exact time instances where the true density changes (for a total of $C-1$ times). Hence,

$$\begin{aligned} L_{NS} &\leq -\log Q(s^*) \prod_{t=1}^{T} \hat{g}_t^{s_t^*}(\boldsymbol{x}_t), \\ &\leq -\log \left(\prod_{t=1}^{C} \frac{1}{t_i} \prod_{i=1}^{C-1} \frac{1}{t_i+1} \prod_{i=1}^{T} \hat{g}_T^{s_i}(\boldsymbol{x}_t) \right), \\ &\leq \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i+1) + \sum_{t=1}^{T} -\log \hat{g}_t^{s_t}(\boldsymbol{x}_t). \end{aligned}$$

After subtracting the log-loss incurred by the best density from both sides, we end up with the bound given in (22). ■

Corollary 2: Combining Theorem 2 and Theorem 3, we can achieve the following regret bound where $\sum_{i=1}^{C} t_i = T$.

$$\begin{aligned} R_{T,NS} \leq& \sum_{i=1}^{C} \log t_i + \sum_{i=1}^{C-1} \log(t_i+1) \\ &+ C \log N + \sum_{i=1}^{C} \frac{D_i}{H_i} (\log(t_i) + 1). \end{aligned}$$

If there exists a $D \geq D_i$ and $H \leq H_i$ for all $i \in \{1, 2, \ldots C\}$, the bound can be written in a more compact form as

$$R_{T,NS} \leq C \left(\left(2 + \frac{D}{H} \right) \log \left(\frac{T}{C} \right) + 1 + \frac{D}{H} + \log N \right). \tag{24}$$

Therefore, our density estimation algorithm achieves Merhav's lower bound $O(C \log T)$ [42] hence achieving the minimax optimal regret.

Remark 5: In [42], the authors show the achievability of logarithmic regret bounds using the Minimum Description Length (MDL) principle. They show that the total number of bits needed for universal coding of a source with piecewise constant parameters is at least $(1.5C-1)\log T$. Hence, our algorithm has a multiplicative redundancy factor of D/H, where D increases with the set of sufficient statistics and H increases with stronger convexity. Note that the lower bounds in [42] are proven for a finite size alphabet and their extension to the continuous distributions are not straightforward. However, similar logarithmic redundancy bounds should hold for different continuous distributions albeit with different constants. Our multiplicative redundancy D/H increases with the hardness of the problem, so it is reasonable to assume that a similar constant exists for the lower bound as well, which is different for different distributions and different estimation problems. Nonetheless, logarithmic performance bounds are always welcomed in both machine learning, signal processing and information theory literature since logarithmic bounds decay exponentially fast.

VI. Anomaly Detection

To decide whether the data $\boldsymbol{x}_t$ is anomalous or not, we threshold the output of the nonstationary density estimation $\hat{g}_t^{u}(\boldsymbol{x}_t)$ of Algorithm 3 with τ_t such that

$$\hat{d}_t = \begin{cases} +1, & \hat{g}_t^{u}(\boldsymbol{x}_t) < \tau_t \text{ (anomaly)} \\ -1, & \hat{g}_t^{u}(\boldsymbol{x}_t) \geq \tau_t \text{ (normal)} \end{cases}. \tag{25}$$

This thresholding is equivalent to thresholding a general monotonically increasing function of $\hat{g}_t^{u}$. Hence, in a more general form, we have

$$\hat{d}_t = \begin{cases} +1, & \hat{p}_t(\boldsymbol{x}_t) < \tau_t \text{ (anomaly)} \\ -1, & \hat{p}_t(\boldsymbol{x}_t) \geq \tau_t \text{ (normal)} \end{cases}, \tag{26}$$

where $\hat{p}_t = \Phi(\hat{g}_t^{u})$ and $\Phi(\cdot)$ is a monotonically increasing function. Our error with $0-1$ loss is given by $\mathbb{1}_{\hat{d}_t \neq d_t}$. However, we emphasize that, in general, the cost of making a prediction error on normal and anomalous data may not be the same. To this end, our algorithm can assign different costs to these distinct errors. Let J_{d_t} be the cost of misclassification of the data with label d_t ($d_t \in \{-1, 1\}$). Then, we can rewrite the error as $J_{d_t} \mathbb{1}_{\hat{d}_t \neq d_t}$. However, the $0-1$ loss definition is not convex in τ_t, which makes optimization difficult. Therefore, we substitute this $0-1$ loss with the widely used logistic loss function $\log(1 + \exp(-(\tau_t - \hat{p}_t)d_t))$, which completely upper bounds the $0-1$ loss $\mathbb{1}_{\hat{d}_t \neq d_t}$. Hence, the loss function is given by

$$l(\tau_t, \hat{p}_t, d_t) = J_{d_t} \log(1 + \exp(-(\tau_t - \hat{p}_t)d_t)).$$

Algorithm 4: Anomaly Detector.

```
Determine error weights $J_{d_t}$ for $d_t \in \{-1, +1\}$
Select scoring mapping $\Phi(\cdot)$
Select feasible set $\mathbb{V}$
Initialize $\alpha$
Initialize $\tau_1$
for $t = 1$ to $T$ do
    Observe $\boldsymbol{x}_t$
    Get density estimation $\hat{g}_t^u(\boldsymbol{x}_t)$ according to
    Algorithm 3
    Produce score $\hat{p}_t = \Phi(\hat{g}_t^u(\boldsymbol{x}_t))$.
    Declare prediction $\hat{d}_t = \mathrm{sign}(\tau_t - \hat{p}_t)$.
    Set $l' = -J_{d_t} d_t (1 + \exp(\tau_t - \hat{p}_t) d_t)^{-1} \mathbb{1}_{\hat{d}_t \neq d_t}$
    $B_t = B_{t-1} + (l')^2$
    $K_t = K_{t-1} + ((l')^2 \tau_t - l'/\boldsymbol{\alpha})$
    $\tau_{t+1} = \arg\min_{v \in \mathbb{V}} (v - K_t/B_t)^2$
end for
```

We compete against a best fixed threshold, thus, regret in a time horizon T is defined as

$$R_{A,T} = \sum_{t=1}^{T} l(\tau_t, \hat{p}_t, d_t) - \min_{\tau \in \mathbb{V}} \sum_{t=1}^{T} l(\tau, \hat{p}_t, d_t), \tag{27}$$

where $\mathbb{V}$ is the feasible set of the threshold. Let the time indices where we make a prediction error be defined by the set T_e. Then, the actual regret that approximates the $0-1$ loss regret is given by

$$R_{A,T} = \sum_{t \in T_e} l(\tau_t, \hat{p}_t, d_t) - \min_{\tau \in \mathbb{V}} \sum_{t \in T_e} l(\tau, \hat{p}_t, d_t). \tag{28}$$

We use Online Newton Step [22] to update our threshold τ_t to minimize the regret in (28). Since the game is now defined only for the time instances we make a prediction error, the threshold is updated only in those time instances. Therefore, the gradient of our loss is given by

$$l'(\tau_t, \hat{p}_t, d_t) = \frac{-J_{d_t} d_t}{1 + \exp(-(\tau_t - \hat{p}_t) d_t)} \mathbb{1}_{\hat{d}_t \neq d_t}. \tag{29}$$

We define variables B_{t-1} and K_{t-1} as follows

$$B_{t-1} = \sum_{r=1}^{t-1} (l'(\tau_r, \hat{p}_t, d_t))^2,$$

$$K_{t-1} = \sum_{r=1}^{t-1} ((l'(\tau_r, \hat{p}_t, d_t))^2 \tau_r - \frac{1}{\alpha} l'(\tau_r, \hat{p}_t, d_t)),$$

for input parameter α. Let $\hat{p}_t$ be in a feasible subset $\mathbb{V}$ of $\mathbb{R}$. Hence, τ_t belongs to the feasible set $\mathbb{V}$, thus, the prediction τ_t is given by

$$\tau_t = \arg\min_{v \in \mathbb{V}} \left(v - \frac{K_{t-1}}{B_{t-1}} \right)^2. \tag{30}$$

The complete algorithm is given in Algorithm 4. Next, we provide regret bounds on the anomaly detector.

Theorem 4: Using Algorithm 4 with parameter

$$\alpha = \frac{1}{2} \min \left(\exp(-A), \frac{1 + \exp(-A)}{4AJ} \right),$$

we achieve the following anomaly detection regret bound for (27)

$$R_{A,T} \leq 3 \left(\exp(A) + \frac{4AJ}{1 + \exp(-A)} \right) \log T, \tag{31}$$

where A is the diameter of the feasible set $\mathbb{V}$ such that $A \triangleq \sup_{x,y \in \mathbb{V}} \|x - y\|$, $J_{\max} \triangleq \max(J_{-1}, J_{+1})$ and $J_{\min} \triangleq \min(J_{-1}, J_{+1})$.

Proof: The loss function is given by

$$l(\tau_t, \hat{p}_t, d_t) = J_{d_t} \log(1 + \exp(-(\tau_t - \hat{p}_t) d_t)). \tag{32}$$

We take the first derivative of the loss function as

$$l'(\tau_t, \hat{p}_t, d_t) = \frac{-J_{d_t} d_t}{1 + \exp((\tau_t - \hat{p}_t) d_t)}, \tag{33}$$

and second derivative as

$$l''(\tau_t, \hat{p}_t, d_t) = \frac{J_{d_t} \exp((\tau_t - \hat{p}_t) d_t)}{(1 + \exp((\tau_t - \hat{p}_t) d_t))^2}, \tag{34}$$

since $d_t^2 = 1$ for $d_t \in \{-1, +1\}$. The loss function is λ-exp-concave for $\lambda = J_{\min} \exp(-A)$ since $\hat{p}_t$ and τ_t are in the feasible set $\mathbb{V}$ and

$$\begin{aligned} \frac{(l'(\tau_t, \hat{p}_t, d_t))^2}{l''(\tau_t, \hat{p}_t, d_t)} &= J_{d_t} \exp(-(\tau_t - \hat{p}_t) d_t), \\ &\geq J_{\min} \exp(-A). \end{aligned}$$

The gradient of the loss function is upper bounded as

$$\|l'(\tau_t, \hat{p}_t, d_t)\| \leq Y = \frac{J_{\max}}{(1 + \exp(-A))}. \tag{35}$$

Using Online Newton Step [22] with $\alpha = 0.5 \min(\lambda, (4AY)^{-1})$, we achieve a regret bound of $3(\lambda^{-1} + 4AY) \log T$. Therefore

$$R_{A,T} \leq 3 \left(\frac{\exp(A)}{J_{\min}} + \frac{4AJ_{\max}}{1 + \exp(-A)} \right) \log T. \tag{36}$$

■

The result of Theorem 4 shows that our misclassification regret is logarithmically dependent on the time horizon T. However, its dependence on the diameter of the feasible set is exponential. To circumvent this problem, we can use suitable transformations such as $\hat{p}_t(\boldsymbol{x}_t) = \log(1 + \hat{g}_t^u(\boldsymbol{x}_t))$. However, attenuating the density estimation output too much may decrease robustness.

Corollary 3: Running Algorithm 4 with transformation $\Phi(x) = \log(1 + x)$ results in the following regret if density estimation is upper-bounded by $P \geq \hat{g}_t^u(\boldsymbol{x}_t)$

$$R_{A,T} \leq 3 \left((1 + P) \left(1 + \frac{4 \log(1 + P) J}{2 + P} \right) \right) \log T. \tag{37}$$

Proof of Corollary 3: Putting $A = \log(1 + P)$ directly in Theorem 4 concludes the proof. ■

Remark 6: In a semi-supervised setting, the threshold can be updated at times when a feedback is received. In this setting, the same performance bounds will hold if we define the regret for only the times feedback is received, i.e., we let the set T_e be the time indices when we receive a feedback and make a prediction error.

Remark 7: In unsupervised setting, we can use a fixed step threshold update instead of the Newton Step approach. Suppose, we want to achieve a false alarm rate of α. If we increment the threshold by δ, when we observe a sample with density estimate above the threshold and decrement it by $\delta(1-\alpha)/\alpha$ when the observed sample has a density estimate below the threshold, we can converge to the optimal threshold with false alarm rate of α for sufficiently small δ and sufficiently large number of observations since our density estimation has logarithmic performance guarantees.

VII. Experiments

In this section, we demonstrate the performance of our algorithm both on synthetic and real data. We use two synthetic datasets and three real datasets to show how our algorithm performs individually and in comparison to the state-of-the art algorithms.

In the first experiments, we are detecting outliers in a synthesized dataset which is generated by a nonstationary Gaussian process. This experiment compares the performances of the algorithms when the anomalies consists of outliers. The second experiment on the other hand is adversarial since the anomalies consist of the recently modeled normal data (because of the leakage). The third experiment shows the performance of our algorithm in comparison to the state-of-the-art in a piecewise stationary real world dataset. The fourth experiment illustrates the performances of the algorithms in a sequential stock market data. The final real world experiment consists of learning in an unsupervised setting.

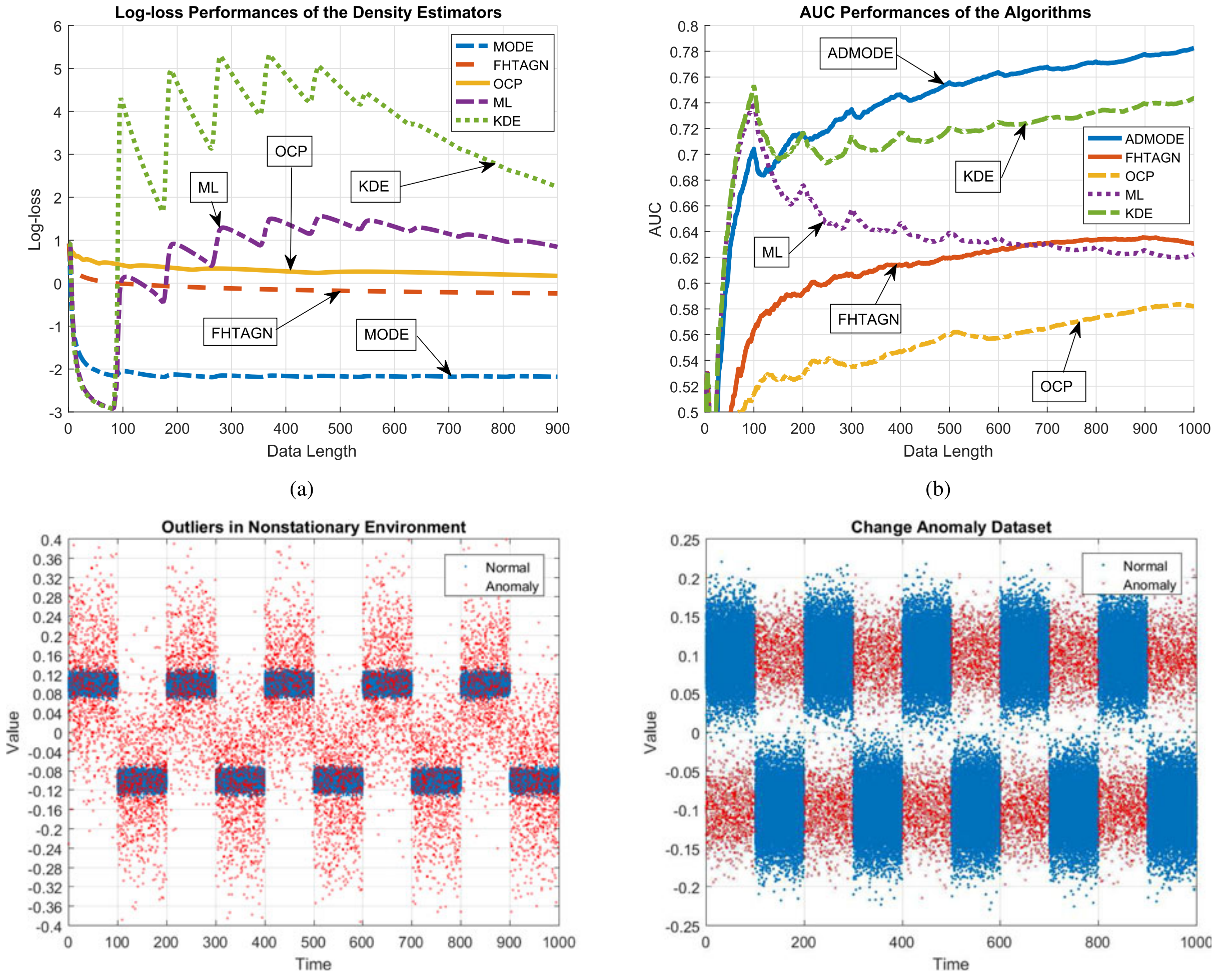
The non-density based anomaly detection approaches such as SVM [32], [33], nearest neighbors [34], [35] and clustering [36] can be thought of as the fitting of appropriate kernels to decide the outliers (e.g., SVM fits appropriate kernels to determine its boundaries and nearest neighbors fits the appropriate kernels on the training sample points respectively). Therefore, we use the Kernel Density Estimator algorithm (KDE) [31] in our performance comparisons for both the density estimation and the anomaly detection. For the benefit of KDE, we use the optimal kernel selection in accordance with the dataset and optimally tune its bandwidth parameter with Silverman's rule [52]. Moreover, we compare our algorithm against the Maximum Likelihood algorithm (ML) [29], which optimally fits an appropriate parametric model to the dataset. Additionally, we compare our method against the adaptive algorithms Filtering and Hedging for Time-varying Anomaly Recognition (FHTAGN) [7] and Online Convex Programming (OCP) [21]. All of these algorithms first estimate a density function for the normal data and thresholds that function to detect anomalies. We used the same thresholding scheme in Algorithm 4 in all of the algorithms for a fair comparison. In the unsupervised real data experiment, the algorithms use a fixed step threshold update for fixed false positive rate as in Remark 7. In all of the experiments, we showed the log-loss performances of the density estimators for normal data to compare their modeling power capabilities. The density estimation part of our algorithm is called Minimax Optimal Density Estimator (MODE) and the anomaly detector is called Anomaly Detection with Minimax Optimal Density Estimation (ADMODE).

To compare the anomaly detection performances of the algorithms, we used the Area Under Curve (AUC) parameter [53]. We used the AUC parameter in [54], [55], and approximated it by using an approach similar to [56], where we have sampled the ROC curve at multiple points by varying the discrimination threshold [56] of each method. This sampling of the ROC curve provided different True Positive Rate (TPR) and False Positive Rate (FPR) pairs. Suppose TPR_i, FPR_i for $i = 1, 2, 3, \ldots, n$ are the sampled points of the ROC curve at different values including the trivial points $(0, 0)$ and $(1, 1)$. Suppose further, these pairs are sorted in an ascending manner according to FPR_i values. Then, we can fit a piecewise linear function onto these samples to approximate the ROC curve. The area under this plot is given by

$$AUC = \sum_{i=1}^{n-1} 0.5\,(TPR_{i+1} + TPR_i)\,(FPR_{i+1} - FPR_i)\,. \tag{38}$$

We use this AUC metric [54] to evaluate the performances of the anomaly detectors. We emphasize that the online setting is different than the batch learning setting. During the course of the algorithms' run, we do not have fixed TPR/FPR pairs, they evolve through time, hence, provide an AUC metric evolving with time. We sample the ROC curve by varying the threshold that determines the behavior of the anomaly detector [56]. However, our online algorithm, i.e., Algorithm 4, dynamically updates the threshold. Hence, to sample the ROC curve at different points, we vary the cost metrics J_{-1} and J_{+1}. These cost metrics are varied one at a time, where we fix one of them to 1 and exponentially vary the other one in 100 steps from 2^{-10} to 1 (i.e., the other cost metric is selected from the set $\{1, 2^{-0.1}, 2^{-0.2}, \ldots, 2^{-9.8}, 2^{-9.9}, 2^{-10}\}$). Since the update speed of the threshold is different at false positives and false negatives, this approach provides us with multiple samples on the ROC curve, which we use to approximate AUC.

We set $H_{\min} = T^{-2}$, $H_{\max} = T$ and $H_{\max} = T^2$ for the density estimation of our algorithm in first three and last two experiments respectively. The algorithms use $\Phi(x) = \log(1+x)$ and $\Phi(x) = \log(x)$ transformation on the density estimations for the threshold update scheme in first three and last two experiments respectively. The diameter of the feasible set A is set to the maximum value of these score estimations for the algorithms. Following [21], OCP is run with the parameter $T_r^{-1/2}$ and $10T_r^{-1/2}$ in first three and last two experiments respectively for each epoch with length T_r. Following [7], FHTAGN is run with the learning rate $1/\sqrt{t}$ and $10/\sqrt{t}$ at time t in first three and last two experiments respectively. ML and KDE uses sliding windows of length $10\log t$, $5\log t$, $\log t$ and $100\log t$ at time t in



(a)

(b)

(c)

(d)

Fig. 1. (a) Average log-loss performances of the density estimation algorithms in the Outliers in Nonstationary Environment Dataset (b) Average AUC performances of the anomaly detection algorithms in the Outliers in Nonstationary Environment Dataset (c) Visualization of the Outliers in Nonstationary Environment Dataset (normal and anomalous sample points) (d) Visualization of the Change Point Anomaly Dataset (normal and anomalous sample points).

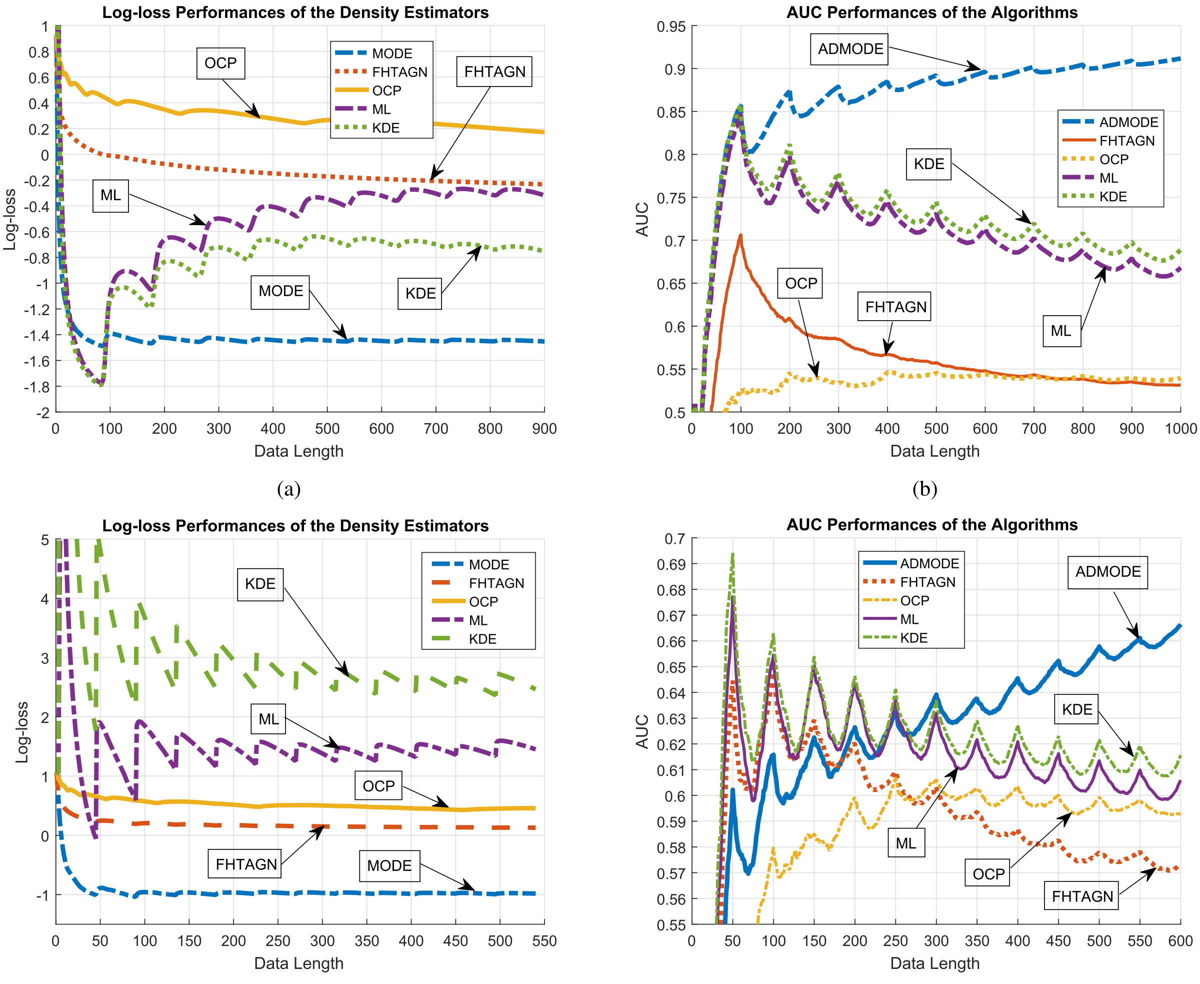
synthetic experiments, Iris dataset, ISE dataset and Occupancy Detection respectively.

A. *Outliers in Nonstationary Environment*

In the first part of the experiments, we compare the performances of the algorithms in detecting outliers in a nonstationary environment. In each trial of the experiments we create a length 1000, i.e., $T = 1000$, dataset that is generated from a Gaussian process with mean changing between 0.1 and -0.1 every 100 samples and variance of 10^{-4}. Then, we randomly select 100 indices that will contain the anomalous data. The samples at the indices containing anomalies are generated from Gaussian processes with mean equal to the mean of the normal data at that index and variance equal to 10^{-2}. We feed this dataset to all of the algorithms and repeat this experiment for 100 trials. The log-loss performances and AUC performances are illustrated in Fig. 1(a) and 1(b) respectively. The normal and anomalous data points generated in all 100 trials are plotted in Fig. 1(c) for illustration of the dataset. As can be seen in Fig. 1(c), the respective position of the outliers to the normal data stays the same.

As can be seen in Fig. 1(a), the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not affected from the environment changes. However, ML and KDE algorithm suffer greatly from the change of source statistics since their tolerance to the nonstationary environment are quite lower. Nonetheless, our algorithm, i.e., MODE achieves a much lower log-loss since its modeling capabilities greatly surpass OCP and FHTAGN.

From Fig. 1(b), we again see that FHTAGN outperforms OCP as in the log-loss performances. Similarly, ML outperforms OCP and FHTAGN at the beginning but its performance deteriorates over time. However, the deterioration speed of its AUC performance is slower than in its log-loss so that it is only outperformed by FHTAGN while still having better performance than OCP. The most interesting performance is maybe achieved by KDE. While KDE performed the worst in terms of log-loss, its AUC performance is significantly better than the other competitors. While its log-loss behavior, similar to ML, deteriorated over time, its AUC performance increases ever so slightly. This perhaps shows that even though KDE may not necessarily model the normal data in the best way, its modeling distinguishes outliers



(a)

(b)

(c)

(d)

Fig. 2. (a) Average log-loss performances of the density estimation algorithms in the Change Point Anomaly Dataset (b) Average AUC performances of the anomaly detection algorithms in the Change Point Anomaly Dataset (c) Average log-loss performances of the density estimation algorithms in the Iris Dataset (d) Average AUC performances of the anomaly detection algorithms in the Iris Dataset.

quite well. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way by keeping its AUC performance nearly unaffected to the changes in the source statistics. ADMODE is able to achieve this superior performance because of its individual sequence approach and superior modeling due to achieving the performance of the best model with the optimal parameters tuned to the underlying sequence.

B. Change Point Anomalies

In the second part of the experiments, we compare the performances of the algorithms in detecting change anomalies in a nonstationary environment, i.e., the anomalous samples in a given time segment are caused by the leakage of normal data from the previous time segment. In each trial of the experiments, we create a length 1000, i.e., $T = 1000$ dataset that is generated from a Gaussian process with mean changing between 0.1 and -0.1 every 100 samples and variance of 10^{-3}. Then, we randomly select 100 indices that will contain the anomalous data. The samples at the indices containing anomalies have their means changed with the mean of the normal data of the previous section. We feed this dataset to all of the algorithms and repeat this experiment for 100 trials. The log-loss performances and AUC performances are illustrated in Fig. 2(a) and 2(b) respectively. The normal and anomalous data points generated in all 100 trials are plotted in Fig. 1(d) for illustration of the dataset. As can be seen in Fig. 1(d), the position of the anomalies in a given time segment, e.g., $t \in \{401, 402, 403, \ldots, 498, 499, 500\}$), is similar to the position of the normal data in the previous time segment, e.g., $t \in \{301, 302, 303, \ldots, 398, 399, 400\}$), since the anomalies in this dataset are created from the leakage of normal data to the successive time segment.

As can be seen in Fig. 2(a), the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not really affected from the environment changes. However, ML and KDE algorithm suffer from the change of source statistics albeit with smaller margins compared to the previous experiment since this dataset has a higher variance. Nonetheless, our algorithm, i.e., MODE achieves a much lower log-loss since its modeling capabilities greatly surpass all of the algorithms because of its individual sequence approach.

From Fig. 2(b), we see that FHTAGN outperforms OCP at the beginning but its performance starts declining with the first environment change until its performance is lower than OCP at the end of the dataset. While ML and KDE outperform OCP and FHTAGN similarly in log-loss, their performance seriously decline with each change in environment. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way because of its superior modeling capabilities. Its AUC performance remains unaffected by the changes in the source statistics, and continues to increase as its learning continues.

C. Iris Dataset

To compare the algorithms in a real life dataset we use the famous Iris dataset [57]. Iris dataset contains 3 classes with 50 instances each, which makes a total of 150 samples. Each instance contains 4 features. We preprocess the dataset so that we separate each feature and treat each feature of each class as separate classes, hence, samples. This creates a dataset of 12 classes with 50 instances each. To get a sequential dataset since we work in the online setting, we randomly decide on an order of appearance for 12 classes and concatenate these samples to get a time series of length 600 at each trial of the experiment. Then, each 50 length section is shuffled randomly in itself. Lastly, we randomly select 5 samples independently for each section and mark them as anomalous. These 5 samples in each section are substituted according to a cyclic order of the sections we randomly decide on. We repeat this experiment for 100 trials and average the results to provide the performances. We illustrate the log-loss performances and AUC performances in Fig. 2(c) and 2(d) respectively.

As can be seen in Fig. 2(c), the algorithms OCP, FHTAGN and MODE are more robust since their log-loss performances are not really affected from the environment changes. However, the performances of ML and KDE suffer with each change of the source statistics. Nonetheless, our algorithm, i.e., MODE, again achieves a much lower log-loss since its modeling capabilities greatly surpass all of the algorithms.

From Fig. 2(d), we see that FHTAGN shows better performance than OCP at the beginning. However, its performance rapidly declines and it is finally outperformed by OCP approximately at the 250th sample. ML and KDE have better performances than OCP and FHTAGN even though they were outperformed in the log-loss performances. Moreover, their performances greatly decline with each change in the environment. Nonetheless, our algorithm, ADMODE, significantly outperforms all of the other algorithms in a stable and robust way because of its superior modeling capabilities. ADMODE is the only algorithm whose AUC performance increases in a decisive manner.

D. Financial Anomalies

We have also compared the algorithms in the “Istanbul Stock Exchange” (ISE) dataset [58], which includes the returns of various international indexes. ISE dataset is a time series of length 536 with 9 features. At each trial of the experiment, we label 54 samples (10%) as anomalous and subtract 0.1 from them (i.e., the returns of all of the indexes are decreased by 10%, e.g., 6% return becomes -4%) to emulate financial anomalies (e.g., a market crash or financial crisis). We repeat this experiment for 100 trials and average the results to provide the performances. We illustrate the log-loss performances and AUC performances in Fig. 3(a) and 3(b) respectively.

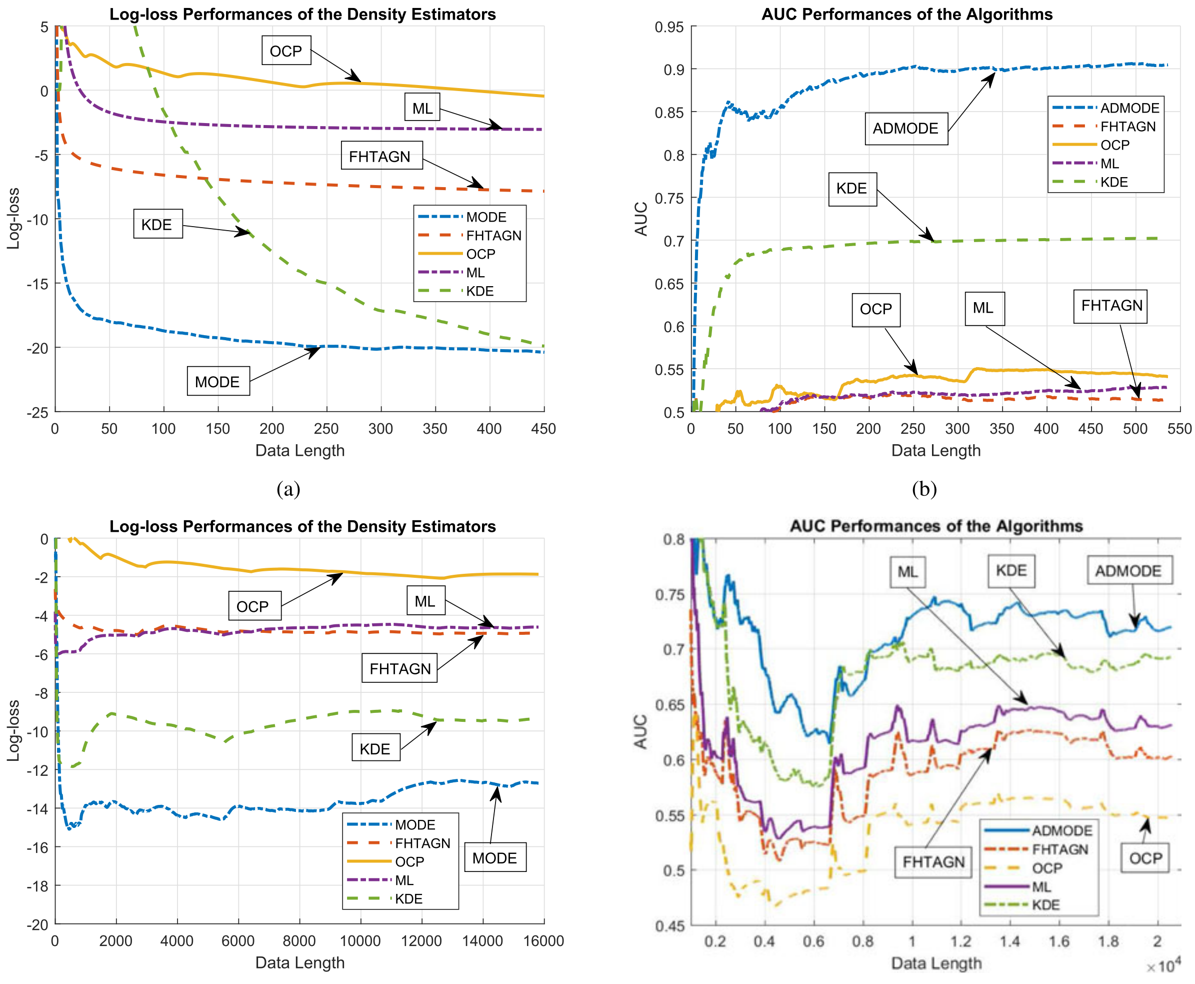
As can be seen in Fig. 3(a), the algorithms OCP, FHTAGN and ML perform poorly. Although KDE achieves a lower log-loss, its convergence is slow. Our algorithm, MODE, on the other hand, achieves the minimum log-loss very fast since it has higher modeling capabilities.

From Fig. 3(b), we see that OCP, FHTAGN and ML have very poor performances similar to before. Although KDE illustrates a higher performance, it can only reach an AUC performance of 0.7. Our algorithm, ADMODE, significantly outperforms all of the other algorithms. Even though KDE and MODE achieve similar log-loss at the end, ADMODE greatly surpasses KDE in AUC performance because of its faster convergence and adaptation.

E. Occupancy Detection

In the last experiment, we have compared the algorithms with a real dataset in an unsupervised setting. We have used the Occupancy Detection dataset [59], which contains 2 classes (occupied or not) and 5 features, which are Temperature (in Celsius), Relative Humidity (%), Light (in Lux), CO_2 (in ppm) and Humidity Ratio. It is divided into three sets of time series. For our experiments, we have concatenated these sets to create a single time series of length 20560, where 15810 of these are normal data (not occupied) and 4750 of these are anomalies (occupied). We have normalized all the features to the intervals $[0, 1]$. Since we are working in an unsupervised setting, the algorithms update their density estimators with each incoming data sample. The log-loss of the algorithms are capped at 300. For the learning rate of the unsupervised update of the threshold, we have exponentially searched over the interval $[\delta, 1/\delta]$ (with multiplicative increments of $2^{0.1}$), where $\delta = 300/T$ ($T = 20560$), since it is the minimum learning increment needed to transverse over the whole log-loss space. We illustrate the log-loss performances and AUC performances in Fig. 3(c) and 3(d) respectively. For each algorithm, the best AUC performance resulting from the optimal selection of the learning rate is plotted.

As can be seen in Fig. 3(c), the OCP algorithm has the worst log-loss performance for the nominal data. The algorithms FHTAGN and ML have similar performances, which are better than OCP. Even though KDE is able to outperform these algorithms, it still performs significantly worse than our algorithm, MODE. From Fig. 3(d), we see that the performance order of the algorithms did not change much. The only difference is surprisingly between the algorithms ML and FHTAGN. Although FHTAGN had better log-loss performance for the nominal data, ML had showed better AUC performance. Because of the characteristics of the dataset, all of the algorithms have shown a similar behavior, where their AUC performance gradually decreases until ≈ 7000th sample and then increases until convergence near the



(a)

(b)

(c)

(d)

Fig. 3. (a) Average log-loss performances of the density estimation algorithms in the Istanbul Stock Exchange Dataset (b) Average AUC performances of the anomaly detection algorithms in the Istanbul Stock Exchange Dataset (c) Average log-loss performances of the density estimation algorithms in the Occupancy Detection Dataset (d) Average AUC performances of the anomaly detection algorithms in the Occupancy Detection Dataset.

end. Since our algorithm is more robust to the changes in the statistics in the dataset, the performance decrease caused by these changes are significantly less. Thus, our algorithm, ADMODE, is able to outperform all of the other algorithms.

VIII. Conclusion

We have introduced a truly sequential anomaly detection algorithm that detects anomalies in a time series. Our algorithm has two stage and consists of first estimating the density of the nominal data in an online manner and then thresholding this density estimation to detect anomalies. We approach this problem from an information theoretic perspective and propose minimax optimal schemes for both stages to create an optimal anomaly detection algorithm in a strong deterministic sense. For the first stage of our algorithm, we, for the first time in literature, have introduced a truly sequential minimax optimal density estimation algorithm for general nonstationary exponential-family of sources without any assumptions on the observation sequence. Moreover, for the second stage, we have proposed a threshold selection scheme that is minimax optimal with respect to the best threshold selected in hindsight. In both stages of our algorithm, we achieve logarithmic regret bounds by adaptively updating its parameters in a completely sequential manner. Our algorithm showed significant performance gains against the state-of-the-art methods in both synthetic and real datasets.

Appendix A
Proof of Theorem 1

The regret at time t is defined as

$$r_t(\hat{\boldsymbol{\alpha}}_t) = l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - l(\boldsymbol{\alpha}, \boldsymbol{x}_t), \tag{39}$$

where $l(\boldsymbol{\alpha}, x)$ is as in (10).

By H-strong convexity, the regret becomes

$$r_t \leq \langle \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t), (\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}) \rangle - \frac{H}{2} \|\hat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|^2. \tag{40}$$

We bound the first term in the right hand side of (40) using the update rule (12). By definition of projection in (13), we have

$$\begin{aligned} &\|P_S(\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t)) - \boldsymbol{\alpha}\| \\ &\quad \leq \|\hat{\boldsymbol{\alpha}}_t - \eta_t \nabla_{\boldsymbol{\alpha}} l(\hat{\boldsymbol{\alpha}}_t, \boldsymbol{x}_t) - \boldsymbol{\alpha}\|. \end{aligned}$$

Substituting (12) in the left hand side provides

$$\|\hat{\boldsymbol{\alpha}}_{t+1}-\boldsymbol{\alpha}\| \leq \|\hat{\boldsymbol{\alpha}}_t-\eta_t\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t)-\boldsymbol{\alpha}\|. \tag{41}$$

Hence, we get

$$\begin{aligned}\|\hat{\boldsymbol{\alpha}}_{t+1}-\boldsymbol{\alpha}\|^2 &\leq \|\hat{\boldsymbol{\alpha}}_t-\eta_t\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t)-\boldsymbol{\alpha}\|^2,\\ &\leq \|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2-2\eta_t\langle\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t),(\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha})\rangle\\ &\quad+\eta_t^2\|\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t)\|^2. \end{aligned}\tag{42}$$

Since $\eta_t > 0$ for all t, rearranging (42) results in

$$\begin{aligned}&\langle\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t),(\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha})\rangle\\ &\leq\frac{1}{2\eta_t}\left(\|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2-\|\hat{\boldsymbol{\alpha}}_{t+1}-\boldsymbol{\alpha}\|^2\right)+\frac{\eta_t}{2}\|\nabla_{\boldsymbol{\alpha}}l(\hat{\boldsymbol{\alpha}}_t,\boldsymbol{x}_t)\|^2,\\ &\leq\frac{1}{2\eta_t}\left(\|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2-\|\hat{\boldsymbol{\alpha}}_{t+1}-\boldsymbol{\alpha}\|^2\right)+\frac{\eta_t}{2}\|\boldsymbol{z}_t-\mu_{\hat{\boldsymbol{\alpha}}_t}\|^2,\end{aligned}\tag{43}$$

where we used (11) in the last step. Putting (43) in the right hand side of (40) yields

$$\begin{aligned}r_t\leq&\frac{1}{2\eta_t}\left(\|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2-\|\hat{\boldsymbol{\alpha}}_{t+1}-\boldsymbol{\alpha}\|^2\right)\\ &+\frac{\eta_t}{2}\|\boldsymbol{z}_t-\mu_{\hat{\boldsymbol{\alpha}}_t}\|^2-\frac{H}{2}\|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2.\end{aligned}\tag{44}$$

Thus, summing (44) from $t = 1$ to T, we have the cumulative regret up to time T, which is given by

$$\begin{aligned}R_T\leq&\frac{1}{2}\sum_{t=2}^{T}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-H\right)\|\hat{\boldsymbol{\alpha}}_t-\boldsymbol{\alpha}\|^2+\frac{1}{2}\left(\frac{1}{\eta_1}-H\right)\\ &\times\|\hat{\boldsymbol{\alpha}}_1-\boldsymbol{\alpha}\|^2-\frac{1}{2\eta_T}\|\hat{\boldsymbol{\alpha}}_{T+1}-\boldsymbol{\alpha}\|^2+\frac{1}{2}\sum_{t=1}^{T}\eta_t\|\boldsymbol{z}_t-\mu_{\hat{\boldsymbol{\alpha}}_t}\|^2,\\ \leq&\sum_{t=1}^{T}\frac{\|\boldsymbol{z}_t-\mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{2Ht},\\ \leq&\frac{D}{2H}(\log T+1),\end{aligned}$$

where we used $\eta_t = (Ht)^{-1}$, and

$$D=\frac{\sum_{t=1}^{T}t^{-1}\|\boldsymbol{z}_t-\mu_{\hat{\boldsymbol{\alpha}}_t}\|^2}{\sum_{t=1}^{T}t^{-1}},\tag{45}$$

which concludes the proof of the theorem.

Appendix B
Proof of Theorem 2

We receive the accumulated log-loss $L_{T,U}$ such that

$$L_{T,U}=\sum_{t=1}^{T}-\log\hat{f}_t^u(\boldsymbol{x}_t)=\sum_{t=1}^{T}-\log\left(\sum_{r=1}^{N}w_t^r\hat{f}_t^r(\boldsymbol{x}_t)\right),$$

where we used (15). Using (16) and (17), we get

$$\begin{aligned}L_{T,U}&=-\log\frac{\sum_{r=1}^{N}\hat{f}_1^r(\boldsymbol{x}_1)}{N}-\sum_{t=2}^{T}\log\left(\frac{\sum_{r=1}^{N}\prod_{\tau=1}^{t}\hat{f}_\tau^r(\boldsymbol{x}_\tau)}{\sum_{r=1}^{N}\prod_{\tau=1}^{t-1}\hat{f}_\tau^r(\boldsymbol{x}_\tau)}\right),\\ &=-\log\frac{1}{N}\sum_{r=1}^{N}\prod_{\tau=1}^{T}\hat{f}_\tau^r(\boldsymbol{x}_\tau)\leq-\log\frac{1}{N}\prod_{\tau=1}^{T}\hat{f}_\tau^r(\boldsymbol{x}_\tau),\end{aligned}$$

for all $r = \{1, 2, \ldots, N\}$. Hence,

$$\begin{aligned}L_{T,U}&\leq\log N+\min_{r\in\{1,2,\ldots,N\}}\sum_{t=1}^{T}-\log\hat{f}_t^r(\boldsymbol{x}_t),\\ &=\log N+\min_r L_{T,r},\end{aligned}$$

where $L_{T,r}$ is the cumulative log-loss of the estimator labeled r.

Subtracting the loss incurred by the true density function from both sides we get,

$$R_{T,U}\leq\log N+\min_r R_{T,r},$$

which concludes the proof.

Appendix C
Proof of Corollary 1

In Theorem 1, we showed that by knowing H a priori and using it as the input in Algorithm 1, we can achieve the regret $R_T \leq \frac{D}{2H}(\log T + 1)$. Running Algorithm 1 with some $H_r \leq H$ incurs regret $\frac{D}{2H_r}(\log T + 1)$ as all the derivations of Theorem 1 would still hold. Since, in the mixture, there is one $H_{r'}$ such that $H/2 < H_{r'} \leq H$, the regret incurred by the corresponding $f^{r'}$ is

$$R_{T,\hat{f}^{r'}}\leq\frac{D}{2H_{r'}}(\log T+1)\leq\frac{D}{H}(\log T+1).$$

Since we mix $N = (\lfloor\log_2\frac{H_{\max}}{H_{\min}}\rfloor + 1)$ number of algorithms, our total regret is given by

$$R_{T,U}\leq\log\left(\left\lfloor\log_2\frac{H_{\max}}{H_{\min}}\right\rfloor+1\right)+\frac{D}{H}(\log T+1),$$

from Theorem 2.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 11, pp. 1–72, 2009.

[2] B. Baingana and G. B. Giannakis, "Joint community and anomaly tracking in dynamic networks," *IEEE Trans. Signal Process.*, vol. 64, no. 8, pp. 2013–2025, Apr. 2016.

[3] K. Cohen, Q. Zhao, and A. Swami, "Optimal index policies for anomaly localization in resource-constrained cyber systems," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4224–4236, Aug. 2014.

[4] J. Sharpnack, A. Rinaldo, and A. Singh, "Detecting anomalous activity on networks with the graph fourier scan statistic," *IEEE Trans. Signal Process.*, vol. 64, no. 2, pp. 364–379, Jan. 2016.

[5] T. Xie, N. M. Nasrabadi, and A. O. Hero, "Learning to classify with possible sensor failures," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 836–849, Feb. 2017.

[6] A. B. Tsybakov *et al.*, "On nonparametric estimation of density level sets," *Ann. Stat.*, vol. 25, no. 3, pp. 948–969, 1997.

[7] M. Raginsky, R. M. Willett, C. Horn, J. Silva, and R. F. Marcia, "Sequential anomaly detection in the presence of noise and limited feedback," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5544–5562, Aug. 2012.

[8] A. Gyorgy, T. Linder, and G. Lugosi, "Efficient tracking of large classes of experts," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6709–6725, Nov. 2012.

[9] A. Gyrgy, T. Linder, and G. Lugosi, "Efficient tracking of large classes of experts," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2012, pp. 885–889.

[10] A. Gyorgy, T. Linder, and G. Lugosi, "Tracking the best quantizer," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1604–1625, Apr. 2008.

[11] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, May 1997.

[12] N. Merhav and M. Feder, "Universal prediction," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2124–2147, Oct. 1998.

[13] A. Bain, *Fundamentals of Stochastic Filtering*. New York, NY, USA: Springer, 2009, vol. 3.

[14] C. R. Shalizi *et al.*, "Dynamics of Bayesian updating with dependent data and misspecified models," *Electron. J. Stat.*, vol. 3, pp. 1039–1074, 2009.

[15] H. Ozkan, F. Ozkan, and S. S. Kozat, "Online anomaly detection under Markov statistics with controllable type-I error," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1435–1445, Mar. 2016.

[16] H. Ozkan, F. Ozkan, I. Delibalta, and S. S. Kozat, "Efficient NP tests for anomaly detection over birth-death type DTMCs," *J. Signal Process. Syst.*, vol. 1, no. 1, pp. 1–10, Jun. 2016.

[17] V. Vapnik and S. Mukherjee, "Support vector method for multivariate density estimation," in *Adv. Neural Inf. Process. Syst.*, pp. 659–665, 2000.

[18] B. O. Koopman, "On distributions admitting a sufficient statistic," *Trans. Amer. Math. Soc.*, vol. 39, no. 3, pp. 399–409, 1936.

[19] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Mach. Learn.*, vol. 43, no. 3, pp. 211–246, Jun. 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010896012157

[20] A. R. Barron and C.-H. Sheu, "Approximation of density functions by sequences of exponential families," *Ann. Stat.*, vol. 19, no. 3, pp. 1347–1369, 1991.

[21] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–936.

[22] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2, pp. 169–192, 2007.

[23] C. W. Ten, J. Hong, and C. C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 865–873, Dec. 2011.

[24] K. B. Dyer, R. Capo, and R. Polikar, "Compose: A semisupervised learning framework for initially labeled nonstationary streaming data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 12–26, Jan. 2014.

[25] M. Herbster and M. K. Warmuth, "Tracking the best linear predictor," *J. Mach. Learn. Res.*, vol. 1, pp. 281–309, 2001. [Online]. Available: http://portal.acm.org/citation.cfm?id=944733.944743

[26] M. Herbster and M. K. Warmuth, "Tracking the best expert," *Mach. Learn.*, vol. 32, no. 2, pp. 151–178, 1998.

[27] F. M. J. Willems, "Coding for a binary independent piecewise-identically-distributed source," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 2210–2217, Nov. 1996.

[28] G. I. Shamir and N. Merhav, "Low complexity sequential lossless coding for piecewise stationary memoryless sources," in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 1998, p. 47.

[29] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Stone Harbor, NJ, USA: Springer, 1994.

[30] C. Horn and R. M. Willett, "Online anomaly detection with expert system feedback in social networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 1936–1939, 2011.

[31] J. S. Simonoff, *Smoothing Methods in Statistics*. Berlin, Germany: Springer Science & Business Media, 2012.

[32] D. M. Tax and R. P. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[33] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[34] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining.*, 2009, pp. 813–822.

[35] G. G. Cabral, A. L. Oliveira, and C. B. Cahú, "Combining nearest neighbor data description and structural risk minimization for one-class classification," *Neural Comput. Appl.*, vol. 18, no. 2, pp. 175–183, 2009.

[36] M. Moshtaghi *et al.*, "Clustering ellipses for anomaly detection," *Pattern Recognit.*, vol. 44, no. 1, pp. 55–69, 2011.

[37] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[38] C. Park, J. Z. Huang, and Y. Ding, "A computable plug-in estimator of minimum volume sets for novelty detection," *Oper. Res.*, vol. 58, no. 5, pp. 1469–1480, 2010.

[39] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231213011314

[40] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *J. Mach. Learn. Res.*, vol. 6, pp. 211–232, 2005.

[41] C. Scott and G. Blanchard, "Novelty detection: Unlabeled data definitely help," in *Proc. 12th Int. Conf. Artif. Intell. Stat.*, 2009, pp. 464–471.

[42] N. Merhav, "On the minimum description length principle for sources with piecewise constant parameters," *IEEE Trans. Inf. Theory*, vol. 39, no. 6, pp. 1962–1967, Nov. 1993.

[43] K. P. Murphy, *Mach. Learning: A Probabilistic Perspective*. Cambridge, MA, USA: The MIT Press, 2012.

[44] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, 2004.

[45] V. Vovk, "Aggregating strategies," in *Proc. 3rd Annu. Workshop Comput. Learn. Theory*, 1990, pp. 371–383.

[46] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, May 1997.

[47] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0893608098001166

[48] Y. Nesterov, "A method of solving a convex programming problem with convergence rate o (1/k2)," in *Proc. Soviet Math. Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, 2015.

[50] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[51] M. D. Zeiler, "Adadelta: An adaptive learning rate method," arXiv:1212.5701, 2012.

[52] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Boca Raton, FL, USA: CRC Press, 1986, vol. 26.

[53] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011.

[54] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.

[55] T. Fawcett, "An introduction to ROC analysis," *J. Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[56] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, 2014.

[57] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.

[58] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid RBF neural networks model as a forecaster," *Statist. Comput.*, vol. 24, no. 3, pp. 365–375, 2014.

[59] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co$_2$ measurements using statistical learning models," *Energy Build.*, vol. 112, pp. 28–39, 2016.

Kaan Gokcesu, photograph and biography not available at the time of publication.

Suleyman S. Kozat (SM'14), photograph and biography not available at the time of publication.