

Sequential Outlier Detection Based on Incremental Decision Trees

Kaan Gokcesu , Mohammadreza Mohaghegh Neyshabouri , Hakan Gokcesu ,
and Suleyman Serdar Kozat, *Senior Member, IEEE*

Abstract—We introduce an online outlier detection algorithm to detect outliers in a sequentially observed data stream. For this purpose, we use a two-stage filtering and hedging approach. In the first stage, we construct a multimodal probability density function to model the normal samples. In the second stage, given a new observation, we label it as an anomaly if the value of aforementioned density function is below a specified threshold at the newly observed point. In order to construct our multimodal density function, we use an incremental decision tree to construct a set of subspaces of the observation space. We train a single component density function of the exponential family using the observations, which fall inside each subspace represented on the tree. These single component density functions are then adaptively combined to produce our multimodal density function, which is shown to achieve the performance of the best convex combination of the density functions defined on the subspaces. As we observe more samples, our tree grows and produces more subspaces. As a result, our modeling power increases in time, while mitigating overfitting issues. In order to choose our threshold level to label the observations, we use an adaptive thresholding scheme. We show that our adaptive threshold level achieves the performance of the optimal prefixed threshold level, which knows the observation labels in hindsight. Our algorithm provides significant performance improvements over the state of the art in our wide set of experiments involving both synthetic as well as real data.

Index Terms—Anomaly detection, exponential family, online learning, mixture-of-experts.

I. INTRODUCTION

A. Preliminaries

WE STUDY sequential outlier or anomaly detection [1], which has been extensively studied due to its applications in a wide set of problems from network anomaly detection [2]–[4] and fraud detection [5] to medical anomaly detection [6] and industrial damage detection [7]. In the sequential outlier

Manuscript received March 9, 2018; revised July 23, 2018; accepted November 17, 2018. Date of publication December 17, 2018; date of current version January 4, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wei Liu. This work was supported by the Turkish Academy of Sciences Outstanding Researcher Programme, TUBITAK, under Contract 117E153. (*Corresponding author: Mohammadreza Mohaghegh Neyshabouri.*)

K. Gokcesu is with the EECS Department, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: gokcesu@mit.edu).

M. M. Neyshabouri and S. S. Kozat are with the Department of Electrical and Electronics Engineering, Bilkent University, Ankara 06800, Turkey (e-mail: mohammadreza@ee.bilkent.edu.tr; kozat@ee.bilkent.edu.tr).

H. Gokcesu is with the IC School, EPFL, Ecublens VD, CH-1015 Lausanne, Switzerland (e-mail: hakan.gokcesu@epfl.ch).

Digital Object Identifier 10.1109/TSP.2018.2887406

detection problem, at each round t , we observe a sample vector $\mathbf{x}_t \in \mathbb{X}$ and label it as “normal” or “anomalous” based on the previously observed sample vectors, i.e., $\mathbf{x}_{t-1}, \dots, \mathbf{x}_1$, and their possibly revealed true labels. After we declare our decision, we may or may not observe the true label of \mathbf{x}_t . The objective is to minimize the number of mislabeled samples.

The problem formulation seems similar to binary classification which has an extensive literature and can be solved with recently emerging general purpose algorithms, e.g. contextual bandits [8]. However, considering the fact that between quantities of normal and anomalous data there likely exists a disparity, and the fact that anomalous data could be scattered with no clear separation from the normal data, solutions specifically tailored for anomaly detection are needed.

For this purpose, we use a two-stage “filtering” and “hedging” method [9]. In the “filtering” stage, we build in an online manner “a model” for “normal” samples based on the information gained from the previous rounds. Then, in the “hedging” stage, we decide on the label of the new sample based on its conformity to our model of normal samples. A common approach in constructing the aforementioned model is to assume that the normal data is generated from an independent and identically distributed (i.i.d.) random sequence [1]. Hence, in the first stage of our algorithm, we model the normal samples using a probability density function, which can also be considered as a scoring function [9]. However, note that the true underlying model of the normal samples can be arbitrary in general (or may not even exist) [1]. Therefore, we approach the problem in a competitive algorithm framework [10]. In this framework, we define a class of models called “competition class” and aim to achieve the performance of the best model in this class. Selecting a rich class of powerful models as the competition class enables us to perform well in a wide set of scenarios [10]. Hence, as detailed later, we choose a strong set of probability functions to compete against and seek to sequentially learn the best density function which fits to the normal data. Hence, while refraining from making any statistical assumptions on the underlying model of the samples, we guarantee that our performance is (at least) as well as the best density function in our competition class.

We emphasize that there exist nonparametric algorithms for density estimation [11], the parametric approaches have recently gained more interest due to their faster convergence [12], [13]. However, the parametric approaches fail if the assumed model is not capable of modeling the true underlying distribution [10]. In this context, exponential-family distributions [14] have attracted

significant attention, since they cover a wide set of parametric distributions [9], and successfully approximate a wide range of nonparametric probabilistic models as well [15]. However, single component density functions are usually inadequate to model the data in highly challenging real life applications [16]. In this paper, in order to effectively model multi-modal distributions, we partition the space of samples into several subspaces using highly effective and efficient hierarchical structures, i.e., decision trees [17]. The observed samples, which fall inside each subspace are fed to a single component exponential-family density estimator. We adaptively combine all these estimators in a mixture-of-experts framework [18] to achieve the performance of their best convex combination.

We emphasize that the main challenge using a partitioning approach for multi-modal density estimation is to define a proper partition of the space of samples [16]. Here, instead of sticking to a pre-fixed partition, we use an incremental decision tree [17], [19] approach to partition the space of samples in a nested structure. Using this method we avoid overtraining, while efficiently modeling complex distributions composed of a large number of components [17]. As the first time in the anomaly detection literature, in order to increase our modeling power with time, we apply a highly powerful incremental decision tree [17]. Using this incremental tree, whenever we believe that the samples inside a subspace cannot belong to a single component distribution, we split the subspace into two disjoint subspaces and start training two new single component density estimators on the recently emerged subspaces. Hence, our modeling power can potentially increase with no limit (and increase if needed), while mitigating the overfitting issues.

In order to decide on the label of a given sample, as widely used in the literature [9], we evaluate the value of our density function in the new data point \mathbf{x}_t and compare it against a threshold. If probability density is lower than the threshold, the sample is labeled as anomalous. While this is shown to be an effective strategy for anomaly detection, setting the threshold is a notoriously difficult problem [9]. Hence, instead of committing to a fixed threshold level, we use an adaptive thresholding scheme and update the threshold level whenever we receive a feedback on the true label of the samples. We show that our thresholding scheme achieves the performance of the best fixed threshold level selected in hindsight.

B. Prior Art and Comparisons

Various anomaly detection methods have been proposed in the literature that utilize Neural Networks [20], Support Vector Machines [21], Nearest Neighbors [22], clustering [23] and statistical methods including parametric [24] and nonparametric [25] density estimation. In the case when the normal data conform to a probability density function, the anomaly detection algorithms based on the parametric density estimation method are shown to provide superior performance [26]. For this reason, we adopt the parametric probability estimation based approach. In [9], authors have introduced an online algorithm to fit a single component density function of the exponential-family distributions to the stream of data. However, since the real life

distributions are best described using multi-modal PDFs rather than single component density functions [27], we seek to fit multi-modal density functions to the observations. There are various multi-modal density estimation methods in the literature. In [16], authors propose a sequential algorithm to learn the multi-modal Gaussian distributions. However, as discussed in their paper, this algorithm provides satisfactory results only if a temporary coherency exists among subsequent observations. In [27], an online variant of the well-known Kernel Density Estimation (KDE) method is proposed. However, no performance guarantees are provided for any of the algorithms. In this paper, we provide a multi-modal density estimation method using an incremental tree with strong performance bounds, which are guaranteed to hold in an individual sequence manner through a regret formulation [9].

Decision trees are widely studied in various applications including coding [19], [28], prediction [29], [30], regression [31] and classification [32]. These structures are shown to provide highly successful results due to their ability to refrain from overtraining while providing significant modeling power. In this paper, we adapt a novel notion of incremental decision trees [19], [33] to the density estimation framework. Using this decision tree, we train a set of single-component density estimators with carefully chosen sets of data samples. We combine these single-component estimators in an ensemble learning [34] framework to approximate the underlying multi-modal density function and show that our algorithm achieves the performance of the best convex combination of the single component density estimators defined on the, possibly infinite depth, decision tree.

Adaptive thresholding schemes are widely used for anomaly detection algorithms based on density estimation [1]. While most of the algorithms in the literature do not provide guarantees for their anomaly detection performance, a surrogate regret bound of $O(\sqrt{t})$ is provided in [9]. However, since in real life applications the labels are revealed in a small portion of rounds [35], stronger performance guarantees are highly desirable. We provide an adaptive thresholding scheme with a surrogate regret bound of $O(\log t)$. Hence, our algorithm steadily achieves the performance of the best threshold level chosen in hindsight.

C. Contributions

Our main contributions are as follows:

- We adapt the notion of incremental decision trees [19] to the multi-modal density estimation framework. We use this tree, which can grow to an infinite depth, to partition the observations space into disjoint subspaces and train different density functions on each subspace. We adaptively combine these density functions to achieve the performance of the best multi-modal density function defined on the tree.
- We provide guaranteed performance bounds for our multi-modal density estimation algorithm. Due to our competitive algorithm framework, our performance bounds are guaranteed to hold in an individual sequence manner.
- Due to our individual sequence perspective, our algorithm can be used in unsupervised, semi-supervised and supervised settings.

- Our algorithm is truly sequential, such that no a priori information on the time horizon or the number of components in the underlying probability density function is required.
- We propose an adaptive thresholding scheme that achieves a regret bound of $O(\log t)$ against the best fixed threshold level chosen in hindsight. This thresholding scheme improves the state-of-the-art $O(\sqrt{t})$ regret bound provided in [9].
- We demonstrate significant performance gains in comparison to the state-of-the-art algorithms through extensive set of experiments involving both synthetic and real data.

D. Organization

In Section II, we formally define the problem setting and our notation. Next, we explain our single-component density estimation methods in Section III. In Section IV, we introduce our decision tree and explain how we use it to incorporate the single-component density estimators and create our multi-modal density function. Then, we explain the anomaly detection step of our algorithm in Section V, which completes our algorithm description. In Section VI we demonstrate the performance of our algorithm against the state-of-the-art methods on both synthetic and real data. We finish with concluding remarks in Section VII.

II. PROBLEM DESCRIPTION

In this paper, all vectors are column vectors and denoted by boldface lower case letters. For a K -element vector \mathbf{u} , u_i represents the i th element and $\|\mathbf{u}\| = \sqrt{\mathbf{u}^T \mathbf{u}}$ is the l^2 -norm, where \mathbf{u}^T is the ordinary transpose. For two vectors of the same length \mathbf{u} and \mathbf{v} , $\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$ represents the inner product. We show the indicator function by $\mathbf{1}_{\{\text{condition}\}}$, which is equal to 1 if the condition holds and 0 otherwise.

We study sequential outlier detection problem, where at each round $t \geq 1$, we observe a sample vector $\mathbf{x}_t \in \mathbb{R}^m$ and seek to determine whether it is anomalous or not. We label the sample vector \mathbf{x}_t by $\hat{d}_t = -1$ for normal samples and $\hat{d}_t = 1$ for anomalous ones, where d_t corresponds to the true label which may or may not be revealed. We make no assumption on the generating model of the labels d_t . They can be decided by the environment (or nature) in an arbitrary fashion which can be nonstationary, chaotic, adversarial and so on.

In general, the cost of making an error on normal and anomalous data may not be the same. Therefore, we define C_{d_t} as the cost of making an error while the true label is d_t . The objective is to minimize the accumulated cost in a series of rounds, i.e., $\sum_{t=1}^T C_{d_t} \mathbf{1}_{\{\hat{d}_t \neq d_t\}}$.

In our two step approach, we first introduce an algorithm for probability density estimation, which learns a multi-modal density function that fits “best” to the observations. This density function can be seen as a scoring function determining the normality of samples. Due to the online setting of our problem, at each round t , our density function estimate, denoted by $\hat{p}_t(\cdot)$, is a function of previously observed samples and their possibly

revealed labels, i.e.,

$$\hat{p}_t(\cdot) = f(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}, d_1, d_2, \dots, d_{t-1}). \quad (1)$$

Note that in general, even if the samples are not generated from a density function, e.g., deterministic framework [36], our estimate $\hat{p}_t(\cdot)$ can be seen as a scoring function determining the normality of the samples. As widely used in the literature [37], we measure the accuracy of our density function estimate \hat{p}_t by the log-loss function

$$l_P(\hat{p}_t(\mathbf{x}_t)) = -\log(\hat{p}_t(\mathbf{x}_t)). \quad (2)$$

In order to refrain from any statistical assumptions on the normal data, we work in a competitive framework [10]. In this framework we seek to achieve the performance of the best model in a class of models called the competition class. We use the notion of “regret” as our performance measure in both density estimation and anomaly detection steps. The regret of a density estimator producing the density function $\hat{p}_t(\cdot)$ against a density function $p(\cdot)$ at round t is defined as

$$r_{P,t}(\hat{p}_t(\mathbf{x}_t), p(\mathbf{x}_t)) = -\log(\hat{p}_t(\mathbf{x}_t)) + \log(p(\mathbf{x}_t)), \quad (3)$$

where selection of $p(\cdot)$ will be clarified later. We denote the accumulated density estimation regret up to time T by

$$R_{P,T} = \sum_{t=1}^T r_{P,t}(\hat{p}_t(\mathbf{x}_t), p(\mathbf{x}_t)). \quad (4)$$

Since the expected value of the per round regret in (3), with respect to the random variable \mathbf{x}_t , corresponds to the KL divergence between the estimate $\hat{p}_t(\cdot)$ and the true distribution $p(\cdot)$, the cumulative regret in (4) asymptotically ($T \rightarrow \infty$) converges to the sum of such KL divergences emerging at each time t which are always nonnegative and are only zero when the estimates $\hat{p}_t(\cdot)$ are equivalent to $p(\cdot)$. Hence, the optimal strategy is indeed to choose $\hat{p}_t(\cdot) = p(\cdot)$.

In order to produce our decision on the label of observations being “normal” or “anomalous”, at each round t , we observe the new sample \mathbf{x}_t and declare our decision by thresholding $\hat{p}_t(\mathbf{x}_t)$ as

$$\hat{d}_t = \text{sign}(\tau_t - \hat{p}_t(\mathbf{x}_t)), \quad (5)$$

where τ_t is the threshold level. After declaring our decision, we may or may not observe the true label d_t as a feedback. We use this information to optimize τ_t whenever we observe the correct decision d_t . We define the loss of thresholding $\hat{p}_t(\mathbf{x}_t)$ by τ_t as

$$l_A(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t) = C_{d_t} \mathbf{1}_{\{\text{sign}(\tau_t - \hat{p}_t(\mathbf{x}_t)) \neq d_t\}}. \quad (6)$$

We define the regret of choosing the threshold value τ_t against a specific threshold τ (which can even be the unknown “best” threshold that minimizes the cumulative error) at round t by

$$r_{A,t}(\tau_t, \tau) = l_A(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t) - l_A(\tau, \hat{p}_t(\mathbf{x}_t), d_t). \quad (7)$$

We denote the accumulated anomaly detection regret up to time T by

$$R_{A,T} = \sum_{t=1}^T r_{A,t}(\tau_t, \tau). \quad (8)$$

We emphasize that the main challenge in “two-step” approaches for anomaly detection is to construct a density function $\hat{p}_t(\cdot)$, which powerfully models the observations distribution. For this purpose, in Section III, we first introduce an algorithm, which achieves the performance of a wide range of single component density functions. Based on this algorithm, in Section IV, we use a nested tree structure to construct a multi-modal density estimation algorithm. In Section V, we introduce our adaptive thresholding scheme, which will be used on the top of the density estimator described in Section IV to form our complete anomaly detection algorithm.

III. SINGLE COMPONENT DENSITY ESTIMATION

In this section we introduce an algorithm, which sequentially achieves the performance of the best single component distribution in the exponential family of distributions [14]. At each round t , we observe a sample vector $\mathbf{x}_t \in \mathbb{R}^m$, drawn from an exponential-family distribution

$$f(\mathbf{x}_t) = h(\mathbf{x}_t) \exp(\langle \eta, \mathbf{s}_t \rangle - A(\eta)), \quad (9)$$

where

- $\eta \in \mathbf{F}$ is the unknown “natural parameter” of the exponential-family distribution. Here, $\mathbf{F} \subset \mathbb{R}^d$ is a bounded convex set.
- $h(\mathbf{x}_t)$ is the “base measure function” of the exponential-family distribution.
- $A(\eta)$ is the “log-partition function” of the distribution.
- $\mathbf{s}_t \in \mathbb{R}^d$ is the “sufficient statistics vector” of \mathbf{x}_t . Given the type of the exponential-family distribution, e.g., Gaussian, Bernoulli, Gamma, etc., \mathbf{s}_t is calculated as a function of \mathbf{x}_t , i.e., $\mathbf{s}_t = T(\mathbf{x}_t)$.

With an abuse of notation, we put the “base measure function” $h(\mathbf{x}_t)$ inside the exponential part by setting $\mathbf{s}_t = [\mathbf{s}_t; \log(h(\mathbf{x}_t))]$ and $\eta = [\eta; 1]$. Hence, from now on, we write

$$f(\mathbf{x}_t) = \exp(\langle \eta, \mathbf{s}_t \rangle - A(\eta)). \quad (10)$$

At each round t , we estimate the natural parameter η based on the previously observed sample vectors, i.e., $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{t-1}\}$, and denote our estimate by $\hat{\eta}_t$. The density estimate at time t is given by

$$\hat{f}_t(\mathbf{x}_t) = \exp(\langle \hat{\eta}_t, \mathbf{s}_t \rangle - A(\hat{\eta}_t)). \quad (11)$$

In order to produce our estimate $\hat{\eta}_t$, we seek to minimize the accumulated loss we would suffer following this $\hat{\eta}_t$ during all past rounds, i.e.,

$$\hat{\eta}_t = \underset{\eta}{\operatorname{argmin}} \sum_{\tau=1}^{t-1} l(\eta, \mathbf{x}_\tau), \quad (12)$$

where

$$l(\eta, \mathbf{x}_\tau) = -\langle \eta, \mathbf{s}_\tau \rangle + A(\eta). \quad (13)$$

This is a convex optimization problem. Finding the point in which the gradient is zero, it can be seen that it suffices to choose the $\hat{\eta}_t$ such that

$$m_{\hat{\eta}_t} = \frac{\sum_{\tau=1}^{t-1} \mathbf{s}_\tau}{t-1}, \quad (14)$$

Algorithm 1: Single Component Density Estimator.

- 1: Initialize $m_s^0 = 0$
 - 2: Select $\hat{\eta}_1 \in \mathbf{F}$ arbitrarily
 - 3: **for** $t = 1$ **to** T **do**
 - 4: Observe $\mathbf{x}_t \in \mathbb{R}^m$
 - 5: Calculate $\mathbf{s}_t = T(\mathbf{x}_t)$
 - 6: Suffer the loss $l(\hat{\eta}_t, \mathbf{x}_t)$ according to (13)
 - 7: Calculate $m_s^t = \frac{m_s^{t-1} \times (t-1) + \mathbf{s}_t}{t}$
 - 8: Calculate $\hat{\eta}_{t+1}$ s.t. $m_{\hat{\eta}_{t+1}} = m_s^t$
 - 9: **end for**
-

where $m_{\hat{\eta}_t}$ is the mean of \mathbf{s}_t when \mathbf{x}_t is distributed with the natural parameter $\hat{\eta}_t$.

Note that the memory demand of our single-component density estimator does not increase with time, as it suffices to keep the sample mean of the “sufficient statistic vectors”, i.e., \mathbf{s}_τ 's, in memory. The complete pseudo code of our single component density estimator is provided in Algorithm 1.

IV. MULTIMODAL DENSITY ESTIMATION

In this section, we extend our basic density estimation algorithm to model the observation vectors using multi-modal density functions of the form

$$p(\mathbf{x}_t) = \sum_{n=1}^N \alpha_n f_n(\mathbf{x}_t), \quad (15)$$

where each $f_n(\cdot)$ is an exponential-family density function as in (9) and $(\alpha_1, \dots, \alpha_N)$ is a probability simplex, i.e., $\forall n : \alpha_n \geq 0$, $\sum_{n=1}^N \alpha_n = 1$.

In order to construct such model, we split the space of sample vectors into several subspaces and run an independent copy of the Algorithm 1 in each subspace. Each one of these density estimators observe only the sample vectors, which fall into their corresponding subspace. We adaptively combine the aforementioned single component density estimators to produce our multi-modal density function. In the following, in Section IV-A, we first suppose that a set of subspaces is given and explain how we combine the density estimators running over the subspaces. Then, in Section IV-B, we explain how we construct our set of subspaces using an incremental decision tree.

A. Mixture of Single Component Density Estimators

Let $\mathcal{S} = \{S_1, \dots, S_N\}$ be a given set of N subspaces of the observation space. For instance, in Fig. 1(a) set of 11 subspaces in \mathbb{R}^2 is shown. We run N independent copies of the Algorithm 1 in these subspaces and denote the estimated density function corresponding to S_i at round t by $\tilde{f}_{t,i}(\cdot)$. We adaptively combine $\tilde{f}_{t,i}(\cdot)$, $i = 1, \dots, N$, in a mixture-of-experts setting using the well known Exponentiated Gradient (EG) algorithm [38]. At each round t , we declare our multi-modal density estimation as

$$\tilde{p}_t(\cdot) = \sum_{i=1}^N \tilde{\alpha}_{t,i} \tilde{f}_{t,i}(\cdot), \quad (16)$$

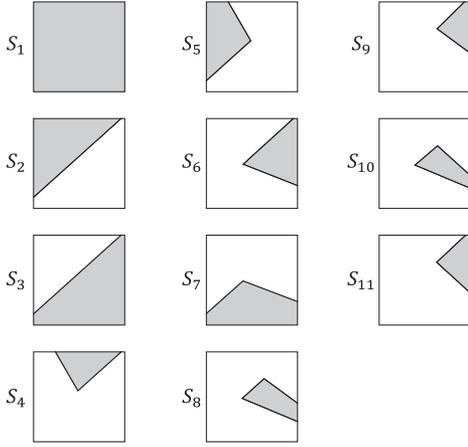


Fig. 1. An example of 11 subspaces of \mathbb{R}^2 . The square shapes represent the whole \mathbb{R}^2 space and the gray regions show subspaces.

where $\tilde{\alpha}_{1,i}$'s are initialized to be $1/N$ for $i = 1, \dots, N$. After observing \mathbf{x}_t , we suffer the loss $l(\tilde{p}_t, \mathbf{x}_t) = -\log(\tilde{p}_t(\mathbf{x}_t))$ and update the mixture coefficients as

$$\tilde{\alpha}_{t+1,i} = \tilde{\alpha}_{t,i} \exp\left(\theta \frac{\tilde{f}_{t,i}(\mathbf{x}_t)}{\tilde{p}_t(\mathbf{x}_t)}\right), \quad (17)$$

where θ is the learning rate parameter. The following proposition shows that in a T rounds trial, we achieve a regret bound of $O(\sqrt{T})$ against the multi-modal density estimator with the best $\tilde{\alpha}$ variables (in the log-loss sense), i.e., the best convex combination of our single component density functions.

Theorem 1: For a T round trial, let R be a bound such that $\max_{t,n} \{\tilde{f}_{t,n}\} \leq R$, for all t, n . Let $p_t^*(\cdot) = \sum_{n=1}^N \alpha_n^* \tilde{f}_{t,n}(\cdot)$ be the optimal (in the accumulated log-loss sense) convex combination of $\tilde{f}_{t,i}$'s with fixed coefficients $(\alpha_1^*, \dots, \alpha_N^*)$ selected in hindsight. If the accumulated log-loss of $p_t^*(\mathbf{x}_t)$ is upper bounded as

$$\sum_{t=1}^T l_P(p_t^*(\mathbf{x}_t)) \leq cT, \quad (18)$$

we achieve a regret bound as

$$R_{P,T}(\tilde{p}_t(\cdot), p_t^*(\cdot)) \leq \sqrt{2cT \ln N} + \frac{R^2 \ln N}{2}. \quad (19)$$

Proof: Denoting the relative entropy distance [39] between the best probability simplex $(\alpha_1^*, \dots, \alpha_N^*)$ and the initial point $(\tilde{\alpha}_{1,1}, \dots, \tilde{\alpha}_{1,N})$ by D , since $\tilde{\alpha}_{1,n} = 1/N, \forall n = 1, \dots, N$, we have

$$D \leq \ln N - H((\alpha_1^*, \dots, \alpha_N^*)), \quad (20)$$

where $H((\alpha_1^*, \dots, \alpha_N^*))$ is the entropy of the best probability simplex. Since the entropy is always positive, we have $D \leq \ln N$. Using Exponentiated Gradient [38] algorithm with the parameter

$$\theta = \frac{2\sqrt{\ln N}}{R\sqrt{2cT} + R^2\sqrt{\ln N}}, \quad (21)$$

we achieve the regret bound in (19).

Remark 1: We emphasize that one can use any arbitrary density estimator in the subspaces and achieve the performance of their best convex combination using the explained adaptive combination. However, since the exponential family distribution covers a wide set of parametric distributions and closely approximates a wide range of non-parametric real life distributions, we use the density estimator in Algorithm 1.

As shown in the theorem, no matter how the set of subspaces \mathcal{S} is constructed, our multi-modal density estimate in (16) is competitive against the best convex combination of the density functions defined over the subspaces in \mathcal{S} . However, the subspaces themselves play an important role in building a proper model for arbitrary multi-modal distributions. For instance, suppose that the true underlying model is a multi-modal PDF composed of several components, which are far away from each other. If we carefully construct subspaces, such that each subspace contains only the samples generated from one of the components (or these subspaces are included in \mathcal{S}), then the best convex combination of the subspaces will be a good model for the true underlying PDF. This scenario is further explained through an example in Section VI-A.

In the following subsection, we introduce a decision tree approach [17] to construct a growing set of proper subspaces and fit a model of the form (15) to the sample vectors. Using this tree, we start with a model with $N = 1$ and increase N as we observe more samples. Hence, while mitigating overfitting issues due to the $\ln N$ bound in (19), our modeling power increases with time.

B. Incremental Decision Tree

We use a decision tree to partition the space of sample vectors into several subspaces. Each node of this tree corresponds to a specific subspace of the observation space. The samples inside each subspace are used to train a single component PDF. These single component probability density functions are then combined to achieve the performance of their best convex combination.

As explained in Section IV-A, our adaptive combination of single component density functions will be competitive against their best convex combination, regardless of how we build the subspaces. However, in order to closely model arbitrary multi-modal density functions of the form (15), we seek to find subspaces that contain only the samples from one of the components. Clearly, this is not always straightforward (or may not be even possible), specially if the centroids of the component densities are close to each other. To this end, we use an incremental decision tree [17], [19] which generates a set of subspaces so that as we observe more samples, our tree adaptively grows and produces more subspaces tuned to the underlying data. Hence, using its carefully produced subspaces, we are able to generate a multi-modal PDF that can closely model the normal data even for complex multi-modal densities, which are hard to learn with classical approaches. We next explain how we construct this incremental tree. We emphasize that we use binary trees as an example and our construction can be extended to multi branch trees in a straightforward manner.

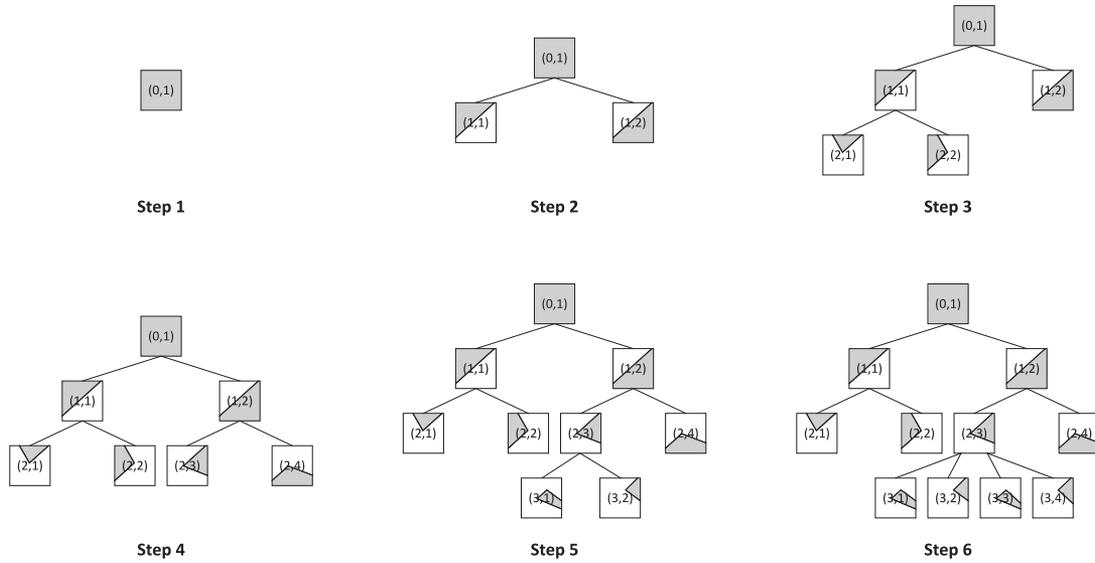


Fig. 2. An example structure of the binary tree introduced in Section IV-B, where the observation space is \mathbb{R} represented as squares here. The regions corresponding to the nodes are colored in gray. Each node is represented by a binary index of the form (i, j) , where i is the level of the node, and j is its order among the nodes in level i .

We start building our binary tree with a single node corresponding to the whole space of the sample vectors. As an example, consider step 1 in Fig. 2. We say that this node is the 1st node in level 0, and denote this node by a binary index of $(0, 1)$, where the first element is the node's level and the second element is the order of the node among its co-level nodes. We grow the tree by splitting the subspace corresponding to a specific node into two subspaces (corresponding to two new nodes) at rounds $t = \beta^k$ for $k = 1, 2, \dots$. Hence, at each round t , the tree will have $\lceil \log_{\beta} t \rceil$ nodes. We emphasize that, as shown in Theorem 1, selecting the splitting times as the powers of β , we achieve a regret bound of $O(\sqrt{T \log \log T})$ against the best convex combination of the single component PDFs (see (19)). Moreover, this selection of splitting times leads to a logarithmic in time computational complexity. However, again, we note that our algorithm is generic so that the splitting times can be selected in any desired manner.

To build subspaces (or sets), we use hyperplanes to avoid overfitting. In order to choose a proper splitting hyperplane, we run a sequential 2-means algorithm [40] over all the nodes as detailed in Algorithm 2. These 2-means algorithms are also used to select the nodes to split as follows. At each splitting time, we split the node that has the maximum ratio of “distance between 2 centroids” to “ $2^{\{\text{level of the node}\}}$ ”, where “level of the node” is the number of splits required to build the node's corresponding subspace as shown in Fig. 2. Note that as this ratio increases, it's implied that the node does not include samples from a single component PDF, which makes it a good choice to split. This motivation is illustrated using a realistic example in Section VI-A. We split the nodes using the hyperplane, which is perpendicular to the line connecting the two centroids of the 2-means algorithm running over the node and splits this line in half. The splitted node keeps a portion of its $\tilde{\alpha}$ value for itself and splits the remaining among its children. This portion, which is a

parameter of the algorithm is denoted by ξ . We emphasize that using the described procedure, each node may be splitted several times. Hence, if the splitting hyperplane is not proper due to lack of observations, the problem can be fixed later by splitting the node again with more accurate hyperplanes in the future rounds. As an example, consider Fig. 2. At the last step, node $(2, 3)$ is splitted again with a slightly shifted splitting line. This is illustrated in more detail using an example in Section VI-A. The algorithm pseudo code is provided in Algorithm 2.

Remark 2: We use linear separation hyperplanes to avoid overtraining while the modeling power is attained by using an incremental tree. However, our method can be directly used with different separation hyperplanes.

As detailed in Algorithm 2, at each round t , the tree nodes declare their single component PDFs, i.e., $\tilde{f}_{t,i}(\cdot)$, $i = 1, \dots, N$. We combine these density functions using (16) to produce our multi-modal density estimate $\tilde{p}_t(\cdot)$. Then, the new sample vector \mathbf{x}_t is observed and we suffer our loss as (2). Subsequently, we update the combination variables, i.e., $\tilde{\alpha}_{t,i}$, $i = 1, \dots, N$, using (17). The centroids of the 2-means algorithms running over nodes are also updated as detailed in Algorithm 2. Finally, the single component density estimates at the nodes are updated as detailed in Algorithm 1. At the end of the round, if $t = \beta^k$, we update the tree structure and construct new nodes as explained in Section IV-B.

In the following section, we explain our adaptive thresholding scheme, which will be used on top of described multi-modal density estimator to form our two-step anomaly detection algorithm.

V. ANOMALY DETECTION USING ADAPTIVE THRESHOLDING

We construct an algorithm, which thresholds the estimated density function $\hat{p}_t(\mathbf{x}_t)$ to label the sample vectors. To this end, we label the sample \mathbf{x}_t by comparing $\hat{p}_t(\mathbf{x}_t)$ with a threshold τ_t

Algorithm 2: IDT-based Multi-modal Density Estimator.

```

1: Select parameters  $\beta$  and  $\xi$ 
2: Initialize  $N = 1$ 
3: Initialize  $\Sigma x_{1,L} = \Sigma x_{1,R} = \mathbf{0}$  (zero vector)
4: Initialize  $\xi_{1,L} = \xi_{1,R} = 1$ 
5: Run Algorithm 1 over node 1.
6: for  $t = 1$  to  $T$  do
7:   Declare  $\hat{p}_t(\cdot)$  as (16)
8:   Observe  $\mathbf{x}_t$ 
9:   for  $n = 1$  to  $N$  do
10:    if  $\mathbf{x}_t$  belongs to the region assigned to node  $n$  then
11:     Update  $f_{t,n}$  using Algorithm 1.
12:     if  $\|\frac{\Sigma x_{n,L}}{\xi_{n,L}} - \mathbf{x}_t\| \leq \|\frac{\Sigma x_{n,R}}{\xi_{n,R}} - \mathbf{x}_t\|$  then
13:        $\Sigma x_{n,L} = \Sigma x_{n,L} + \mathbf{x}_t$ 
14:        $\xi_{n,L} = \xi_{n,L} + 1$ 
15:     else
16:        $\Sigma x_{n,R} = \Sigma x_{n,R} + \mathbf{x}_t$ 
17:        $\xi_{n,R} = \xi_{n,R} + 1$ 
18:     end if
19:   end if
20: end for
21: Update  $\tilde{\alpha}$  variables as (17)
22: if  $t = \beta^k$  then
23:   Select the node  $n$  as explained in Section IV-B
24:   Let  $L = \Sigma x_{n,L} / \xi_{n,L}$ ,  $R = \Sigma x_{n,R} / \xi_{n,R}$ 
25:   Split the node using the hyperplane with normal
   vector of  $a = D / \|D\|$  and  $b = \langle a, (L + R) / 2 \rangle$ ,
   where  $D = L - R$ . (Hyperplane:  $\langle a, x \rangle = b$ )
26:   Run copies of Algorithm 1 over new nodes.
27: end if
28: end for

```

as

$$\hat{d}_t = \begin{cases} +1, & \hat{p}_t(\mathbf{x}_t) < \tau_t \\ -1, & \hat{p}_t(\mathbf{x}_t) \geq \tau_t. \end{cases} \quad (22)$$

Suppose at some specific rounds $t \in T_f$, after we declared our decision \hat{d}_t the true label d_t is revealed. We seek to use this information to minimize the total regret defined in (8). However, since we observe the incurred loss only at rounds $t \in T_f$, we restrict ourselves to these rounds. Moreover, since the loss function used in (8) is based on the indicator function that is not differentiable, we substitute the loss function defined in (6) with the well known logistic loss function defined as

$$\tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t) = C_{d_t} \log(\exp((\hat{p}_t(\mathbf{x}_t) - \tau_t)d_t) + 1). \quad (23)$$

Our aim is to achieve the performance of the best constant τ in a convex feasible set \mathbf{G} . To this end, we define our regret as

$$\tilde{R}_{T_f} = \sum_{t \in T_f} \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t) - \min_{\tau \in \mathbf{G}} \sum_{t \in T_f} \tilde{l}(\tau, \hat{p}_t(\mathbf{x}_t), d_t), \quad (24)$$

We use the Online Gradient Descent algorithm [41] to produce our threshold level τ_t . To this end, we choose $\tau_1 \in \mathbf{G}$ arbitrarily.

Algorithm 3: IDT-based Anomaly Detector.

```

1: Select parameters  $C_1$  and  $C_{-1}$ 
2: Fix  $\alpha_t$  using (27) for  $t = 1, \dots, T$ 
3: Select  $\tau_1 \in \mathbf{G}$  arbitrarily
4: for  $t = 1$  to  $T$  do
5:   Observe  $\hat{p}_t(\mathbf{x}_t)$ 
6:   Calculate  $\hat{d}_t$  using (22)
7:   Observe  $d_t$ 
8:   Suffer the loss  $\tilde{l}(\hat{\eta}_t, \mathbf{x}_t)$  according to (23)
9:   Calculate  $\tau_{t+1} = \mathbb{P}_{\mathbf{G}} \left( \tau_t + \frac{\alpha_t d_t C_{d_t}}{1 + \exp((\tau_t - \hat{p}_t(\mathbf{x}_t))d_t)} \right)$ 
10: end for

```

At each round t , after declaring our decision \hat{d}_t , we construct

$$\tau_{t+1} = \begin{cases} \mathbb{P}_{\mathbf{G}} \left(\tau_t - \alpha_t \nabla_{\tau} \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t) \right), & \text{if } d_t \text{ is known} \\ \tau_t, & \text{otherwise,} \end{cases} \quad (25)$$

where α_t is the step size at time t and $\mathbb{P}_{\mathbf{G}}(\cdot)$ is a projection function defined as

$$\mathbb{P}_{\mathbf{G}}(a) = \operatorname{argmin}_{b \in \mathbf{G}} \|b - a\|. \quad (26)$$

The complete algorithm is provided in Algorithm 3.

For the sake of notational simplicity, from now on, we assume that d_t is revealed at all time steps. We emphasize that since the rounds with no feedback do not affect neither the threshold in (25), nor the regret in (24), we can simply ignore them in our analysis. The following theorem shows that using Algorithm 3, we achieve a regret upper bound of $O(\log T)$, against the best fixed threshold level selected in hindsight.

Theorem 2: Using Algorithm 3 with step size

$$\alpha_t = \frac{(1 + \exp(\mathcal{D}_{\mathbf{G}}))^2}{t C_{\min} \exp(\mathcal{D}_{\mathbf{G}})}, \quad (27)$$

our anomaly detection regret in (24) is upper bounded as

$$\tilde{R}_T \leq \frac{\exp(\mathcal{D}_{\mathbf{G}}) C_{\max}^2}{2 C_{\min}} (1 + \log T), \quad (28)$$

where $\mathcal{D}_{\mathbf{G}} = \max_{a, b \in \mathbf{G}} \|a - b\|$ is the diameter of the feasible set \mathbf{G} including τ_t and $\hat{p}_t(\mathbf{x}_t)$. C_{\max} and C_{\min} are the maximum and minimum of $\{C_1, C_{-1}\}$, respectively.

Proof: Considering the loss function in (23), we take the first derivatives of \tilde{l} as

$$\frac{\partial \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t)}{\partial \tau_t} = \frac{-d_t C_{d_t}}{1 + \exp((\tau_t - \hat{p}_t(\mathbf{x}_t))d_t)}, \quad (29)$$

and its second derivative as

$$\frac{\partial^2 \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t)}{\partial \tau_t^2} = \frac{C_{d_t} \exp((\tau_t - \hat{p}_t(\mathbf{x}_t))d_t)}{(1 + \exp((\tau_t - \hat{p}_t(\mathbf{x}_t))d_t))^2}. \quad (30)$$

The first derivative can be bounded as

$$\left| \frac{\partial \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t)}{\partial \tau_t} \right| \leq \frac{C_{\max}}{1 + \exp(-\mathcal{D}_{\mathbf{G}})}. \quad (31)$$

Similarly, the second derivative is bounded as

$$\left| \frac{\partial^2 \tilde{l}(\tau_t, \hat{p}_t(\mathbf{x}_t), d_t)}{\partial \tau_t^2} \right| \geq \frac{C_{\min} \exp(\mathcal{D}_{\mathbf{G}})}{(1 + \exp(\mathcal{D}_{\mathbf{G}}))^2}. \quad (32)$$

Using Online Gradient Descent [41], with step size given in (27) we achieve the regret upper bound in (28).

VI. EXPERIMENTS

In this section, we demonstrate the performance of our algorithm in different scenarios involving both real and synthetic data. In the first experiment, we have created a synthetic scenario to illustrate how our algorithm works. In this scenario, we sequentially observe samples drawn from a 4-component distribution, where the probability density function is a convex combination of 4 multivariate Gaussian distributions. The samples generated from one of the components are considered anomalous. The objective is to detect these anomalous samples. In the second experiment, we have shown the superior performance of our algorithm with respect to the state-of-the-art methods on a synthetic dataset, where the underlying PDF cannot be modeled as a multi-modal Gaussian distribution. The third experiment shows the performance of the algorithms on a real multi-class dataset. In this experiment, the objective is to detect the samples belonging to one specific class, which are considered anomalous.

We compare the density estimation performance of our algorithm *ITAN*, against a set of state-of-the-art competition composed of *wGMM* [42], *wKDE* [42], and *ML* algorithms. The *wGMM* [43] is an algorithm which uses a sliding window of the last $\log t$ normal samples to train a GMM using the well known Expectation-Maximization (EM) [43] method. The length of sliding window is set to $\log t$ in order to have a fair comparison against our algorithm in the sense of computational complexity. In favor of the *wGMM* algorithm, we provide to it the number of components that provides the best performance for that algorithm. The *wKDE* is the well-known KDE [42] algorithm that uses a sliding window of the last \sqrt{t} normal samples to produce its estimate on the density function. The length of sliding window is \sqrt{t} in favor of the *wKDE* algorithm to produce competitive results. The kernel bandwidth parameters are chosen based on Silverman's rule [42]. Finally, *ML* algorithm is the basic Maximum Likelihood algorithm which fits the best single-component density function to the normal samples. We use our algorithm *ITAN* with the parameters $\beta = 2$, $\xi = 0.8$ and with sufficient statistics of Gaussian in all three experiments. We emphasize that no optimization has been performed to tune parameters β and ξ to the datasets.

In order to compare the anomaly detection performance of the algorithms, we use the same thresholding scheme described in Algorithm 3 for all algorithms. We use the ROC curve as our performance metric. Given a pair of false negative and false positive costs, denoted by C_1 and C_{-1} , respectively, each algorithm achieves a pair of True Positive Rate (TPR) and False Positive Rate (FPR), which determines a single point on its corresponding ROC curve. In order to plot the ROC curves, we have repeated the experiments 100 times, where $C_1 = 1$ and

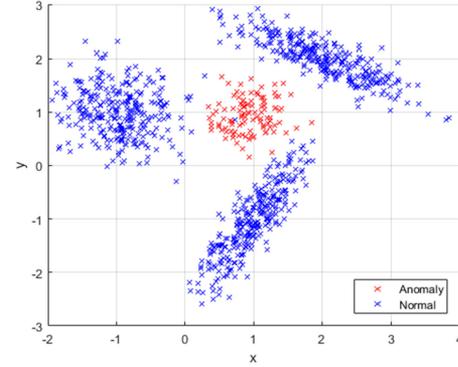


Fig. 3. Visualization of samples in one of the datasets used in Experiment VI-A.

C_2 is selected from the set of $\{\frac{i}{100} | i = 0, 1, \dots, 99\}$. The ROC curves are plotted using the resulting 100 samples. The Area Under Curve (AUC) of the algorithms are also calculated using these samples as another performance metric.

A. Synthetic Multimodal Distribution

In the first experiment, we have created 10 datasets of length 1000 and compared the performance of the algorithms in both density estimation and anomaly detection tasks. Each sample is labeled as “normal” or “anomalous” with probabilities of 0.9 and 0.1, respectively. The normal samples are randomly generated using the density function

$$f_{\text{normal}}(\mathbf{x}_t) = \frac{1}{3} \left(N \left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix} \right) + N \left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0.14 & 0.2 \\ 0.2 & 0.4 \end{bmatrix} \right) + N \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 0.4 & -0.2 \\ -0.2 & 0.14 \end{bmatrix} \right) \right), \quad (33)$$

while the anomalous samples are generated using

$$f_{\text{anomaly}}(\mathbf{x}_t) = N \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \right). \quad (34)$$

Fig. 3 shows the samples in one of the datasets used in this experiment to provide a clear visualization.

In order to show how our algorithm learns, we illustrate how the tree splits the observation space, how the density estimations train their single component PDFs and how the combination of single component PDFs models the normal data in the experiment over one of the 10 datasets. Fig. 4 shows five growth steps of the tree. In each subfigure, the observed samples are shown using black cross signs. The centroids of the 2-means algorithm running over the node that is going to split are shown using two blue and red points. The thicker green line is the new splitting line, while the thinner green lines show previous splitting lines. The splittings shown in this figure result in a tree structure that is shown in Fig. 2. Fig. 5 shows how the single component PDFs defined over nodes are combined to construct our

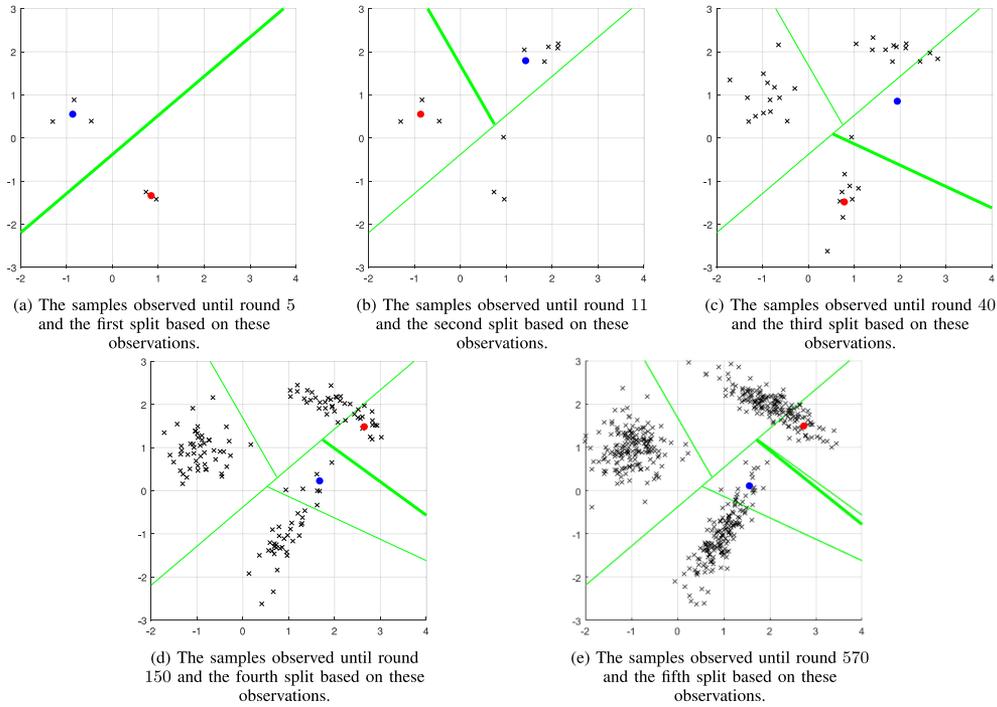


Fig. 4. An example on how the tree learns the underlying distribution of the samples. The normal samples are from a synthetic dataset generated using (33).

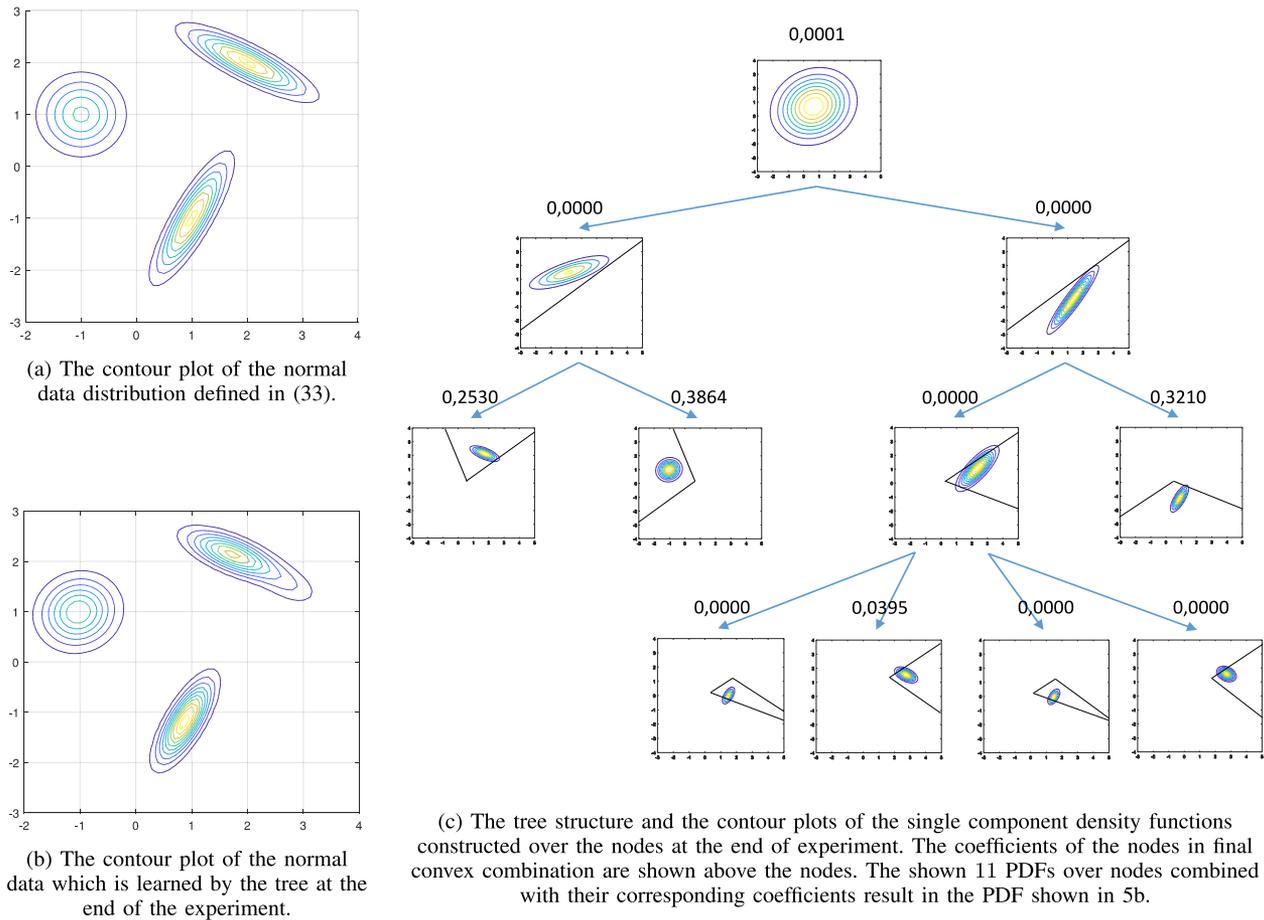


Fig. 5. The true underlying PDF, the tree structure and the single component PDFs defined over nodes, and the final PDF learned by the algorithm at the end of the experiment on one of the datasets of the first experiment described in Section VI-A.

TABLE I

“LOG-LOSS,” “AUC,” AND “RUNNING TIME” OF THE ALGORITHMS OVER THE DATASETS DESCRIBED IN SECTION VI-A. THE AUC AND RUNNING TIME VALUES ARE IN THE FORMAT OF “MEAN VALUE \pm STANDARD DEVIATION”

Algorithm	Log-Loss	Area Under Curve	Running Time (ms)
ITAN	2.174	0.8281 ± 0.1383	313.02 ± 5.82
wGMM	3.056	0.8394 ± 0.0210	2275.67 ± 123.23
wKDE	3.022	0.5678 ± 0.0228	273.16 ± 6.92
ML	3.087	0.2700 ± 0.0943	38.56 ± 1.33

TABLE II

“LOG-LOSS,” “AUC,” AND “RUNNING TIME” OF THE ALGORITHMS OVER THE DATASETS DESCRIBED IN SECTION VI-B. THE AUC AND RUNNING TIME VALUES ARE IN THE FORMAT OF “MEAN VALUE \pm STANDARD DEVIATION”

Algorithm	Log-Loss	Area Under Curve	Running Time (ms)
ITAN	0.833	0.7962 ± 0.0757	315.85 ± 13.67
wGMM	1.303	0.7381 ± 0.0285	7167.78 ± 140.99
wKDE	1.455	0.6863 ± 0.0321	283.21 ± 1.10
ML	1.337	0.6859 ± 0.0323	34.82 ± 0.43

multi-modal density function. In Fig. 5(a) the contour plot of the normal data distribution function is shown. Fig. 5(c) shows the structure of the tree at the end of the experiment, the contour plots of the single component PDFs learned over the nodes, and their coefficient in the convex combination which yields the final multi-modal density function. The contour plot of this final multi-modal PDF is shown in Fig. 5(b). As shown in these figures, the three components of the underlying PDF are almost captured by the three nodes generated in the second level of our tree.

In order to compare the density estimation performance of the algorithms, their averaged loss per round defined by $\text{Loss}(t) = \sum_{\tau=1}^t l_P(\hat{p}_\tau(\mathbf{x}_\tau))/t$, are shown in Fig. 7(a). The loss of all algorithms on the rounds with anomalous observations are considered as 0 in these plots. The anomaly performance of the algorithms are compared in Fig. 7(d). This figure shows the ROC curves of the algorithms averaged over 10 datasets. The time-averaged log-loss performance, AUC results and running time of the algorithms are provided in Table I. All results are obtained using an Intel(R) Core(TM) i5-4570 CPU with 3.20 GHz clock rate and 8 GB RAM.

As shown in Figs. 7(a), our algorithm achieves a significantly superior performance for the density estimation task. This superior performance was expected because in the dataset used for this experiment the components are far from each other. Hence, our tree can generate proper subspaces, which contain only the samples from one of the components of the underlying PDF, as shown in Fig. 5. For the anomaly detection task, as shown in Fig. 7(d), our algorithm and wGMM provide close performance, where ITAN performs better in low FPRs and wGMM provides superior performance in high FPRs. However, as shown in Table I, we achieve this performance with a significantly lower computational complexity. Comparing Fig. 7(a) and Fig. 7(d) shows that while satisfactory log-loss performance is required for successful anomaly detection, it is not sufficient in general. For instance, while ML algorithm performs as well as wKDE and wGMM in the log-loss sense, its anomaly detection performance is much worse than the others. In fact, labeling the samples

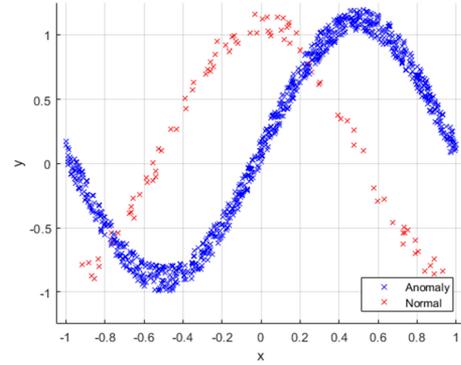


Fig. 6. Visualization of samples in one of the datasets used in Experiment VI-B.

exactly opposite of the suggestions of the ML algorithm provides way better anomaly detection performance. This is because of the weakness of the model assumed by the ML algorithm. This weak performance of the ML algorithm was expected due to the underlying PDF of the normal and anomalous data. It can be also seen from Fig. 3. If we fit a single component Gaussian PDF to the normal samples shown in blue, roughly speaking, the anomalous samples shown in red will get the highest normality score when evaluated using our PDF.

In the next experiment, we compare the algorithms in a scenario, where the data cannot be modeled as a convex combination of Gaussian density functions.

B. Synthetic Arbitrary Distribution

In this experiment, we have created 10 datasets of length 1000. In order to generate each sample, first its label is randomly determined to be “normal” or “anomalous” with probabilities of 0.9 and 0.1, respectively. The normal samples are 2-dimensional vectors $\mathbf{x}_t = [x_{t,1}, x_{t,2}]^T$ generated using the following distribution:

$$\begin{cases} f_{\text{normal}}(x_{t,1}) = \mathcal{U}(-1, 1), \\ f_{\text{normal}}(x_{t,2}) = \mathcal{U}(\sin(\pi x_{t,1}), \sin(\pi x_{t,1}) + 0.2), \end{cases} \quad (35)$$

where $\mathcal{U}(a, b)$ is the uniform distribution between a and b . The anomalous samples are generated using the following distribution:

$$\begin{cases} f_{\text{anomaly}}(x_{t,1}) = \mathcal{U}(-1, 1), \\ f_{\text{anomaly}}(x_{t,2}) = \mathcal{U}(\cos(\pi x_{t,1}), \cos(\pi x_{t,1}) + 0.2). \end{cases} \quad (36)$$

Fig. 6 shows the samples in one of the datasets used in this experiment to provide a clear visualization.

Fig. 7(b) shows the averaged accumulated loss of the algorithms averaged over 10 data sets. As shown in the figure, our algorithm outperforms the competitors for the density estimation task. This superior performance is due to the growing in time modeling power of our algorithm. The ROC curves of the algorithms for the anomaly detection task are shown in Fig. 7(e). This figure shows that our algorithm provides superior anomaly detection performance as well. This superior performance is due to the better approximation of the underlying PDF made

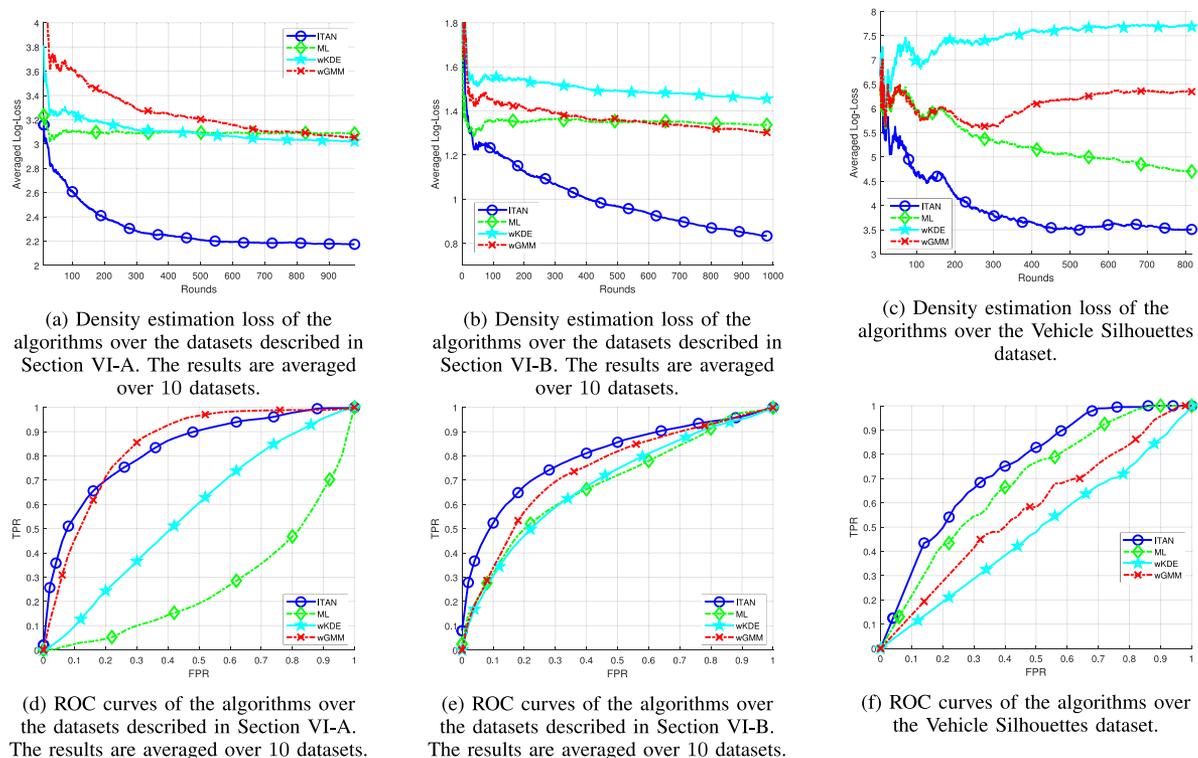


Fig. 7. The averaged density estimation loss and ROC curves of the algorithms over three experiments.

by our algorithm. The averaged log-loss performance, AUC results and running time of the algorithms in this experiment are summarized in Table II.

For brevity, tables for real experiments are excluded.

C. Real Multiclass Dataset

In this experiment, we use Vehicle Silhouettes [44] dataset. This dataset contains 846 samples. Each sample includes a 18-dimensional feature vector extracted from an image of a vehicle. The labels are the vehicle class among 4 possible classes of “Opel”, “Saab”, “Bus” and “Van”. Our objective in this experiment is to detect the vehicles with “Van” labels as our anomalies. Fig. 7(c), shows the density estimation loss of the opponents, based on the rounds in which they have observed “normal” samples. Fig. 7(f) shows the ROC curves of the algorithms. As shown in the figures, our algorithm achieves a significantly superior performance in both density estimation and anomaly detection tasks over this dataset.

Fig. 7(c) shows that the performance of *wGMM* highly depends on the stationarity of normal samples stream. The intrinsic abrupt change of the underlying model at around round 250 has caused a heavy degradation in its density estimation performance. However, our algorithm shows a robust log-loss performance even in the case of non-stationarity. Fig. 7(f) shows that our algorithm achieves the best anomaly detection performance among the competitors. Note that *ML* algorithm outperforms both *wGMM* and *wKDE* algorithms in both density estimation and anomaly detection tasks. This is because *wGMM* and *wKDE* suffer from overfitting due to the high dimensionality of the sample vectors and short time horizon of the experiment. However,

due to the growing tree structure used in our algorithm, we significantly outperform the *ML* algorithm and provide highly superior and more robust performance compared to the all other three algorithms.

D. Real Anomaly Detection Datasets

In this section, we will compete against more density estimators. In the previous experiments we have drawn the ROC curve using our thresholding scheme. This time, for variety to further the examination of how well these density estimators work in anomaly detection, we will draw the ROC curve by varying a fixed threshold instead.

We have included three new real dataset called Wisconsin-Breast Cancer Diagnostics dataset (WBC), Thyroid Disease dataset (Thyroid) and Japanese Vowels dataset (Vowels) [45].

We have renamed one of the competitors and included three new ones, denoted as “G-ROT”, “G-LCV”, “E-LSCV”, “E-HSJM” in the plots. These competitors are based on non-parametric density estimators. In the denotations, before the hyphen, “G” refers to Gaussian kernel and “E” refers to the Epanechnikov kernel, also called the optimal kernel [46]; after the hyphen refers to the bandwidth selection strategies. “ROT” is the Silverman’s rule of thumb method. “LCV” is the likelihood cross-validation method. “LSCV” is the least-squares cross-validation method and “HSJM” is the method proposed by Hall *et al.* in [47].

In these new experiments, we observe that only our algorithm ITAN performs well in both log-loss and ROC plots for all three datasets. This can be attributed to the fact that our algorithm combines best of parametric and non-parametric approaches by

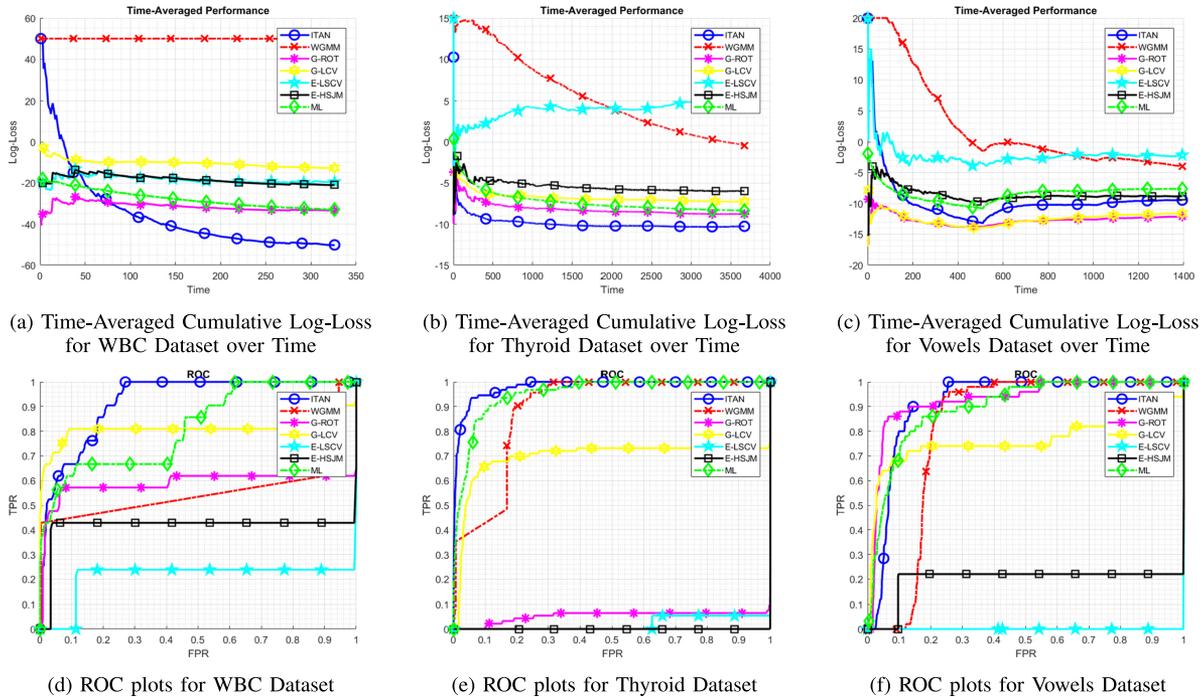


Fig. 8. (a) Time-averaged cumulative log-loss for WBC dataset over time. (b) Time-averaged cumulative log-loss for thyroid dataset over time. (c) Time-averaged cumulative log-loss for vowels dataset over time. (d) ROC plots for WBC dataset. (e) ROC plots for thyroid dataset. (f) ROC plots for vowels dataset.

creating coarser and finer estimators via an incremental tree which hierarchically separates the sample space. Coarser estimators towards the tree root, spanning relatively larger regions, behave like component learning in parametric approaches, while finer estimators towards the leaves, spanning relatively smaller regions, behave more like kernels in non-parametric approaches.

As seen in Figs. 8(a), 8(d), 8(b) and 8(e), ITAN outperforms the competition for both WBC and Thyroid datasets in terms of both log-loss and area under ROC plot. The competition “G-LCV” performs better on a small region of low FPR for WBC dataset as seen in Fig. 8(d), however it performs the second worst in log-loss for WBC dataset as seen in Fig. 8(a).

As seen in Fig. 8(c) and 8(f), for the Vowels dataset, our algorithm ITAN is outperformed by “G-LCV” in log-loss plots and by “G-ROT” in both the log-loss and area under ROC plot. However, “G-LCV” performs very poorly in area under ROC plot for Vowels dataset as in Fig. 8(f). Furthermore, “G-ROT” have performed very poorly in log-loss and area under ROC plot for WBC and Thyroid datasets as in Figs. 8(a), 8(b), and 8(d), 8(e), respectively.

Based on these new set of experiments, we have observed that our algorithm ITAN performs reliably well while performance of the competitors heavily depend on the dataset.

VII. CONCLUDING REMARKS

We studied the sequential outlier detection problem and introduced a highly efficient algorithm to detect outliers or anomalous samples in a series of observations. We use a two-stage method, where we learn a PDF that best describes the normal samples, and decide on the label of the new observations based

on their conformity to our model of normal samples. Our algorithm uses an incremental decision tree to split the observation space into subspaces whose number grow in time. A single component PDF is trained using the samples inside each subspace. These PDFs are adaptively combined to form our multi-modal density function. Using the aforementioned incremental decision tree, while avoiding overtraining issues, our modeling power increases as we observe more samples. We threshold our density function to decide on the label of new observations using an adaptive thresholding scheme. We prove performance upper bounds for both density estimation and thresholding stages of our algorithm. Due to our competitive algorithm framework, we refrain from any statistical assumptions on the underlying normal data and our performance bounds are guaranteed to hold in an individual sequence manner. Through extensive set of experiments involving synthetic and real datasets, we demonstrate the significant performance gains of our algorithm compared to the state-of-the-art methods.

REFERENCES

- [1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.
- [2] M. Thottan and C. Ji, “Anomaly detection in IP networks,” *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.
- [3] 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.
- [4] 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.
- [5] C. Phua, D. Alahakoon, and V. Lee, “Minority report in fraud detection: Classification of skewed data,” *SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 50–59, Jun. 2004.

- [6] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," *SIGMOD Rec.*, vol. 30, no. 2, pp. 37–46, May 2001.
- [7] R. Fujimaki, T. Yairi, and K. Machida, "An approach to spacecraft anomaly detection problem using kernel feature space," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2005, pp. 401–410.
- [8] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Adv. Neural Inf. Process. Syst. 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds., New York, NY, USA: Curran Associates, Inc., 2008, pp. 817–824.
- [9] 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.
- [10] A. C. Singer and S. S. Kozat, "A competitive algorithm approach to adaptive filtering," in *Proc. 7th Int. Symp. Wireless Commun. Syst.*, IEEE, 2010, pp. 350–354.
- [11] 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.
- [12] 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. 90, pp. 175–184, 2018.
- [13] K. Gokcesu and S. S. Kozat, "Online anomaly detection with minimax optimal density estimation in nonstationary environments," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1213–1227, Mar. 2018.
- [14] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Foundations Trends Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, 2008.
- [15] A. R. Barron and C.-H. Sheu, "Approximation of density functions by sequences of exponential families," *The Ann. Statist.*, vol. 19, pp. 1347–1369, 1991.
- [16] O. Arandjelovic and R. Cipolla, "Incremental learning of temporally-coherent gaussian mixture models," *Soc. Manuf. Eng. Tech. Papers*, 2006.
- [17] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 3, pp. 660–674, May/Jun. 1991.
- [18] 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, 1997.
- [19] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.
- [20] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*. Springer, 2002, pp. 170–180.
- [21] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 25–36.
- [22] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection," in *Applications of Data Mining in Computer Security*. New York, NY, USA: Springer, 2002, pp. 77–101, 2002.
- [23] A. M. Pires and C. Santos-Pereira, "Using clustering and robust estimators to detect outliers in multivariate data," in *Proc. Int. Conf. Robust Statist.*, 2005.
- [24] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, "Informal identification of outliers in medical data," in *Proc. 5th Int. Workshop Intell. Data Anal. Med. Pharmacol.*, vol. 1, 2000, pp. 20–24.
- [25] K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining Knowl. Discovery*, vol. 8, no. 3, pp. 275–300, 2004.
- [26] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An experimental evaluation of novelty detection methods," *Neurocomputing*, vol. 135, pp. 313–327, 2014.
- [27] M. Kristan, A. Leonardis, and D. Skočaj, "Multivariate online kernel density estimation with gaussian kernels," *Pattern Recognit.*, vol. 44, no. 10–11, pp. 2630–2642, 2011.
- [28] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inf. Theory*, vol. 35, no. 2, pp. 299–315, Mar. 1989.
- [29] D. P. Helmbold and R. E. Schapire, "Predicting nearly as well as the best pruning of a decision tree," *Mach. Learn.*, vol. 27, no. 1, pp. 51–68, 1997.
- [30] O. J. J. Michel, A. O. Hero, and A. E. Badel, "Tree-structured nonlinear signal modeling and prediction," *IEEE Trans. Signal Process.*, vol. 47, no. 11, pp. 3027–3041, Nov. 1999.
- [31] N. D. Vanli and S. S. Kozat, "A comprehensive approach to universal piecewise nonlinear regression based on trees," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5471–5486, Oct. 2014.
- [32] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote Sens. Environ.*, vol. 61, no. 3, pp. 399–409, 1997.
- [33] P. E. Utgoff, N. C. Berkman, and J. A. Clouse, "Decision tree induction based on efficient tree restructuring," *Mach. Learn.*, vol. 29, no. 1, pp. 5–44, 1997.
- [34] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Inf. Fusion*, vol. 37, pp. 132–156, 2017.
- [35] C. Scott and G. Blanchard, "Novelty detection: Unlabeled data definitely help," in *Proc. 12th Int. Conf. Artif. Intell. Statist.*, 2009, pp. 464–471.
- [36] M. Kloppenburg and P. Tavan, "Deterministic annealing for density estimation by multivariate normal mixtures," *Phys. Rev. E*, vol. 55, no. 3, 1997, Art. no. R2089.
- [37] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, Cambridge, MA, USA: MIT Press, 2012.
- [38] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," Univ. California Santa Cruz, Comput. Res. Lab., Santa Cruz, CA, USA, Tech. Rep. UCSC-CRL-94-16, 1994 (Revised Dec. 7, 1995. An extended abstract to appeared in the STOC 95, pp. 209–218).
- [39] J. N. Kapur and H. K. Kesavan, "Entropy optimization principles and their applications," in *Entropy and Energy Dissipation in Water Resources*. New York, NY, USA: Springer, pp. 3–20, 1992.
- [40] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
- [41] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, no. 2, pp. 169–192, Dec. 2007.
- [42] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, vol. 26, Boca Raton, FL, USA: CRC Press, 1986.
- [43] J. A. Bilmes *et al.*, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Tech. Rep. ICSI-TR-97-021, 1998.
- [44] J. P. Siebert, "Vehicle recognition using rule based methods," Turing Institute Research Memorandum TIRM-87-018, 1987.
- [45] C. C. Aggarwal and S. Sathe, "Theoretical foundations and algorithms for outlier ensembles," *ACM SIGKDD Explor. Newslett.*, vol. 17, no. 1, pp. 24–47, 2015.
- [46] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory Probab. Appl.*, vol. 14, no. 1, pp. 153–158, Jan. 1969.
- [47] P. Hall, S. J. Sheather, M. C. Jones, and J. S. Marron, "On optimal data-based bandwidth selection in Kernel density estimation," *Biometrika*, vol. 78, no. 2, pp. 263–269, 1991.

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

Mohammadreza Mohaghegh Neyshabouri, photograph and biography not available at the time of publication.

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

Suleyman Serdar Kozat, photograph and biography not available at the time of publication.