FITNESS: (Fine Tune on New and Similar Samples) to detect anomalies in streams with drift and outliers

Abishek Sankararaman [1] **Balakrishnan (Murali) Narayanaswamy** [1] **Vikramank Singh** [*1] **Zhao Song** [*1]

Abstract

Technology improvements have made it easier than ever to collect diverse telemetry at high resolution from any cyber or physical system, for both monitoring and control. In the domain of monitoring, anomaly detection has become an important problem in many research areas ranging from IoT and sensor networks to devOps. These systems operate in real, noisy and non-stationary environments. A fundamental question is then, '*How to quickly spot anomalies in a data-stream, and differentiate them from either sudden or gradual drifts in the normal behaviour?*' Although several heuristics have been proposed for detecting anomalies on streams, no known method has formalized the desiderata and rigorously proven that they can be achieved. We begin by formalizing the problem as a sequential estimation task. We propose FITNESS, (**Fi**ne **T**une on **Ne**w and **S**imilar **S**amples), a flexible framework for detecting anomalies on data streams. We show that in the case when the data stream has a gaussian distribution, FITNESS is provably both robust and adaptive. The core of our method is to fine-tune the anomaly detection system only on recent, similar examples, before predicting an anomaly score. We prove that this is sufficient for robustness and adaptivity. We further experimentally demonstrate that FITNESS is *flexible* in practice, i.e., it can convert existing offline AD algorithms in to robust and adaptive online ones.

1. Introduction

Anomaly detection (AD) on streaming high-dimensional data has become increasingly important to a variety of applications ranging from real-time monitoring of industrial and sensor systems (Hill and Minsker, 2010), or software security (Ahmed et al., 2016). Roughly speaking, the task of AD is to identify which if any of a given set of data points are surprising or inconsistent (Chandola et al., 2009). This is typically achieved by algorithms that output a real valued score for each input data point, where larger the score, the more anomalous the input point is.

We study the *online version* of AD, where the data arrives sequentially, and the AD algorithm must produce an anomaly score for a data point before it can see the next one. The online version has attracted study recently due to its wide applicability - including e.g. in infrastructure monitoring (Xu et al., 2018; Audibert et al., 2020) and security applications (Bhatia et al., 2021a;b). We define and study a theoretical framework that captures commonly encountered challenges in design of AD algorithms — drift and noise — that make anomalies hard to detect and model.

Drift typically manifests as a change in the distribution of observed data over time. A comprehensive classification of drifts is in (Gama et al., 2014). In an AD system, the drift can either be *gradual, but continuously occurring* - for example due to wear and tear of machines in an industrial monitoring system (Martí et al., 2015), or it can be *rare, but abrupt* for example due to seasonal variations or external interventions on the sensor (Hill and Minsker, 2010; Saurav et al., 2018). Thus the first desideratum of a practical anomaly detection system is that it be *adaptive* (Definition 3.5) to drifts in the data stream.

Being adaptive in an online setting however, requires us to trade off adaptivity with *robustness* (Gama et al., 2014). Roughly speaking, an online algorithm is said to be robust (Definition 3.6), if it can tolerate a small number of adversarially corrupted data points. The trade-off is inherent since drifts, when they occur, are hard to distinguish from data corruptions. This trade-off is exacerbated in anomaly detection systems, as anomalies are usually defined to be those data-points that differ from the baseline, i.e., appear like potential corruptions. On the other hand, being adaptive to changes is critical in practical anomaly detection systems, since otherwise they will raise a large number of false positives after a change. In practical AD systems, false

[*]Equal contribution [1]Amazon AWS AI Labs, Santa Clara, CA. Correspondence to: Abishek Sankararaman <abisanka@amazon.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

positives are the key performance drivers, since alert fatigue will cause users to turn off a system, and thus it is important to quickly learn the "new normal" when a drift occurs (Ruff et al., 2021).

1.1. Desiderata of online AD algorithms

Most online AD systems aim to incorporate and trade-off the following (Togbe et al., 2021).

1. **Limited Supervision:** No feedback on the detections are assumed to be available online
2. **Streaming:** Anomaly score for a given input to be produced before processing the next.
3. **Competitive in Stationary case:** If all non-anomalous points are sampled i.i.d., the performance improves over time.
4. **Adaptive to Shifts:** In the case the data-stream exhibits distribution shifts —either a sudden shift, or slowly varying drifts, the model must be adapt.
5. **Robust:** The algorithm can tolerate a small number of data corruptions in the stream.

1.2. Main Contributions

- A statistical formulation of the desiderata of practical streaming AD systems, with lower bounds on achievable performance that define the complexity parameters of data streams for AD.
- We show our desiderata to be reasonable and non-trivial target benchmarks that are not achieved with obvious algorithms.
- In the case of sub-Gaussian distributions and L_2 loss, we propose `FITNESS:GAUSSIAN` that provably achieves the desiderata. We empirically validate the superiority of `FITNESS:GAUSSIAN` thereby showing that the gains evidenced by theory are fundamental and not artefacts of mathematical bounding techniques.
- We propose `FITNESS:GENERAL` and demonstrate that it is *flexible* in practice, in that it can convert any batch AD algorithm into an adaptive and online one.

2. Related Work

Online AD Surveys and taxonomies of AD algorithms are in (Chandola et al., 2009; Ruff et al., 2021; Togbe et al., 2021). We focus on unsupervised online algorithms for AD here. We classify online AD algorithms as memory based or sliding window based. MEMSTREAM (Bhatia et al., 2021b), is made adaptive by keeping a dynamic memory buffer to estimate statistics. MEMSTREAM discards a sample if it is predicted as an anomaly *at the time of scoring the data point*. This form of adaptation was also proposed in SketchDetect (Huang and Kasiviswanathan, 2015). We show (Section 5.3), that such techniques can fail to distinguish between anomalies and abrupt changes. StreamIF (Ding and Fei, 2013), xStream (Manzoor et al., 2018) and (Raab et al., 2020; Zhang et al., 2016) propose sliding window algorithms to adapt to drifts. However, the window size is an input that must be supplied. We show in Section 5.2 that the optimal window size depends on the data statistics which may be unknown apriori, and thus these algorithms are not truly adaptive. DiLOF (Na et al., 2018) proposes a fast heuristic to trade-off not retraining on seemingly anomalous points, and being adaptive to abrupt changes. RSForest, (Wu et al., 2014), RSHash (Sathe and Aggarwal, 2016) LODA (Pevnỳ, 2016) Adaptive Trees (Heigl et al., 2021) HSTrees (Tan et al., 2011) and iLOF (Pokrajac et al., 2007) incrementally train on every input point, i.e., they neither keep a reference window, nor discard anomalous points We show empirically that our algorithms outperform these on real datasets.

Continual Learning Reviews of handling concept drifts in stream learning are in (Lu et al., 2018; Gupta et al., 2013). (Bifet and Gavalda, 2007; Bifet Figuerol and Gavaldà Mestre, 2009) study the problem of when to discard samples and when to fine-tune in the one-dimensional setting. Their algorithms have a change point detector interleaved with an estimator. Our setting generalizes theirs by considering multi-dimensional data with drifts and adversarial corruptions. (Miyaguchi and Kajino, 2019) proposed an adaptation technique to tune learning rates of SGD in a drifting stream. Time-series AD with drifts have been studied in (Liu et al., 2016; Laxhammar and Falkman, 2013; Saurav et al., 2018) and references therein.

Online supervised learning: The adaptivity robustness trade-off was studied in the supervised setting in (Chu et al., 2004). Streaming SVRG (Frostig et al., 2015) and its variants, such as SAGA (Defazio et al., 2014), DYNASAGA (Daneshmand et al., 2016) and DriftSurf (Tahmasbi et al., 2021) were proposed as improvements in the online supervised learning setting. In contrast to our unsupervised setting, these algorithms have unambiguous information as to how well they are performing, since they can explicitly compute the loss in the supervised setting.

Robust Learning in the presence of adversarially corrupted data has attracted renewed attention (Diakonikolas et al., 2017; 2018; Cheng et al., 2020; 2019; Diakonikolas and Kane, 2019a). These are limited to the offline setting and do not consider drift.

3. Problem Setting

3.1. Notations and Definitions

Definition 3.1 (Benign Data Stream with distribution $\boldsymbol{\mathcal{D}}$)**.** A sequence of T, d dimensional random vectors $\widetilde{\mathbb{X}} := \{\widetilde{X}_1, \cdots, \widetilde{X}_T\}$ is said to be *a benign data stream from dis-*

tribution $\boldsymbol{\mathcal{D}} := (\mathcal{D}_i)_{i=1}^T$, if $(X_i)_{i=1}^T$ are independent with $X_i \sim \mathcal{D}_i$, where every $\mathcal{D}_i \in \mathcal{F}$ belongs to a known class of distributions $\mathcal{F}$.

Definition 3.2 (Υ Corrupted data stream)**.** A data stream $\mathbb{X} := \{X_1, \cdots, X_T\}$ is said to be Υ corrupt, if there exists a set of T vectors $\mathbf{c} := (c_t)_{t=1}^T$, with each $c_t \in \mathbb{R}^d$ and cardinality $|\{t : c_t \neq 0\}| \leq \Upsilon$, such that for all $t \in [T]$, $X_t := \widetilde{X}_t + c_t$, where $\widetilde{\mathbb{X}} \sim \boldsymbol{\mathcal{D}}$ is a benign data-stream. In other words, any benign data-stream that is corrupted in at-most Υ time points is defined as an Υ corrupted data stream. We term the time points that are corrupted ($c_t \neq 0$) as *anomalous points* and the rest as *non-anomalous points*.

Definition 3.3 (AD Model Class $\{g(\theta, \cdot)\}_{\theta \in \Theta}$)**.** A family of functions $\{g(\theta, \cdot)\}_{\theta \in \Theta}$ is termed an *AD model class*, if for every $\theta \in \Theta$, $g(\theta, \cdot) : \mathbb{R}^d \to \mathbb{R}$, is a function mapping $\mathbb{R}^d$ to $\mathbb{R}$. For $\theta \in \Theta$, $g(\theta, \cdot) : \mathbb{R}^d \to \mathbb{R}$ is denoted as *AD model* θ and for $X \in \mathbb{R}^d$, $g(\theta, X) \in \mathbb{R}$ as the *anomaly score on input* X *by AD model* θ.

We give two examples of AD model class to help parse the definition. One is the auto-encoder framework (Kim et al., 2020), where θ denotes the weights of an auto-encoder with fixed architecture, and $g(\theta, X) \geq 0$ is the reconstruction loss. The second one is Extended Isolation Forests (Hariri et al., 2019), where Θ is the space of all forests, and for a given forest θ, $g(\theta, X) \geq 0$ is the score computed by forest θ, on input X.

Definition 3.4 (Loss function $\mathcal{L}$ for an AD model class $\{g(\theta, \cdot)\}_{\theta \in \Theta}$)**.** A function $\mathcal{L} : \Theta \times \Theta \to \mathbb{R}_+$ is said to be a *loss function for the AD model class* $\{g(\theta, \cdot)\}_{\theta \in \Theta}$, if for every $\theta_1, \theta_2 \in \Theta$, $\mathcal{L}(\theta_1, \theta_2)$ measures the difference in performance between AD models θ_1 and θ_2.

The value $\mathcal{L}(\theta_1, \theta_2)$ measures the distance between the two functions $g(\theta_1, \cdot) : \mathbb{R}^d \to \mathbb{R}$ and $g(\theta_2, \cdot) : \mathbb{R}^d \to \mathbb{R}$. For appropriately chosen $\mathcal{L}$, the value $\mathcal{L}(\theta_1, \theta_2)$ is a proxy for the difference in performance of the AD system with models θ_1 and θ_2. Throughout this paper, we focus on the case where $\mathcal{L}(\theta_1, \theta_2) := \|\theta_1 - \theta_2\|$ is the Euclidean norm between the two parameter vectors. This measure is known to be indicative of the performance difference between two Lipschitz continuous AD models under the assumption of *realizability* (Kim et al., 2020). Realizability is a common assumption that posits $\exists \theta^* \in \Theta$ not necessarily unique, and a model class $\{g(\theta, \cdot)\}_{\theta \in \Theta}$, such that if each point X_i of a dataset $\mathbb{X}$ was given a score $S_i := g(\theta^*, X_i) \in \mathbb{R}$, then all non-anomalous points in $\mathbb{X}$ will receive a lower score than any anomalous point. Thus, for any input $X \in \mathbb{R}^d$ the binary label of whether it is an anomaly or not is declared by thresholding the anomaly scores $g(\theta, X)$. So two AD models θ_1 and θ_2 will yield similar performance after thresholding, if the difference $|g(\theta_1, X) - g(\theta_2, X)|$ is small for all $X \in \mathbb{R}^d$. In the case when the AD model class is Lipschitz continuous, $\|\theta_1 - \theta_2\|$ gives a proxy for how dis-similar AD models θ_1 and θ_2 are (Kim et al., 2020).

3.2. A sequential interaction protocol

The AD algorithm is instantiated with *(i)* an AD model class $\{g(\theta, \cdot)\}_{\theta \in \Theta}$, and *(ii)* a corresponding loss function $\mathcal{L}$. At each time $t = 1, ...,$ the AD algorithm

1. First observes $X_t \in \mathbb{R}^d$, the t-th entry of a Υ corrupted stream from a known family $\mathcal{F}$.
2. Estimates $\theta_t \in \Theta$, using all observations $X_1, \cdots, X_t$ thus far and outputs anomaly score $S_t := g(\theta_t, X_t)$.
3. Incurs loss $r_t := \inf\{\mathcal{L}(\theta_t, \theta_t^*) \text{ s.t } \theta_t^* \in \arg\min_{\theta \in \Theta} \mathbb{E}_{X \sim \mathcal{D}_t} \mathbb{E}[g(\theta, X)]\}$.

However, the loss r_t is not observed, as the AD algorithm does not know $\mathcal{D}_t$ and thus cannot compute θ_t^*. This is different from classical online learning (Cesa-Bianchi and Lugosi, 2006), where at-least for some of the actions, some form of feedback — complete, partial, or noisy — is observed at some point of time.

3.3. Performance Measure of Online AD

The sequence of functions $\mathcal{A}$ is an *online AD algorithm*, if for every $t \in [T]$, the output estimate is $\theta_t := \mathcal{A}(X_1, \cdots, X_t) \in \Theta$. The performance of an algorithm $\mathcal{A}$ on a Υ corrupted data-stream with distribution $\boldsymbol{\mathcal{D}}$ is measured by regret defined

$$R_T(\boldsymbol{\mathcal{D}}, \Upsilon, \mathcal{A}) := \sup_{\mathbf{c} \text{ s.t.} \|\mathbf{c}\|_0 \leq \Upsilon} \sum_{t=1}^{T} \mathbf{1}(c_t = 0) r_t.$$

For every algorithm and data stream the regret is a random variable, measuring the cumulative loss on the uncorrupted time points, over all valid choices of the adversary. The regret does not measure loss on the adversarially corrupted data points, and we do not constrain the underlying distribution generating the sample before corruption. Minimizing loss only on non-anomalous points has been used in the past to optimize AD model classes such as One-Class SVMs (Schölkopf et al., 2001) and is in line with practical considerations of controlling false positives (Ruff et al., 2021).

3.4. Formalizing the desiderata

We seek AD algorithms with sub-linear regret, if the 'complexity' of the problem as defined below, is small. This complexity definitions are justified based on lower bounds presented in Sec 4.

The **Distribution Change Complexity** of a sequence of distributions $\boldsymbol{\mathcal{D}}$, denoted by $\Phi(\boldsymbol{\mathcal{D}}) := \sum_{t=2}^{T} \text{TV}(\mathcal{D}_t, \mathcal{D}_{t-1})$ - denotes the cumulative distribution changes over time. Here TV is the total variation distance between the two probability measures. This measure of distribution shift complexity is analogous to the path length metric used for measuring

complexity in online optimization (Zinkevich, 2003). We drop the argument $\boldsymbol{\mathcal{D}}$, and refer to the distribution change complexity as Φ, whenever it is clear from the context.

The **Corruption Complexity** $\Upsilon := \sum_{t=1}^{T} \mathbf{1}(c_t \neq 0)$.

Definition 3.5 (Adaptivity)**.** An algorithm $\mathcal{A}$ is said to be *adaptive*, if for every $\rho < 1$, there exists a $\beta < 1$, such that

$$\limsup_{T\to\infty} \sup_{\substack{\boldsymbol{\mathcal{D}} \text{ s.t.} \\ \mathcal{D}_i \in \mathcal{F}, \forall i \in [T], \\ \Phi \leq T^{\rho}}} \frac{\mathbb{E}[R_T(\boldsymbol{\mathcal{D}}, 0, \mathcal{A})]}{T^{\beta}} = 0.$$

In words, $\mathcal{A}$ is adaptive if, when $\boldsymbol{\mathcal{D}}$ has *sub-linear* distribution shift and no adversarial corruptions, $\mathcal{A}$ achieves sub-linear expected regret.

Definition 3.6 (Adaptive and Robust Algorithms)**.** An algorithm $\mathcal{A}$ is said to be both adaptive and robust, if for every $\rho < 1$ and $c < 1$, there exists a $\beta < 1$, such that

$$\limsup_{T\to\infty} \sup_{\substack{\boldsymbol{\mathcal{D}} \text{ s.t.} \\ \mathcal{D}_i \in \mathcal{F}, \forall i \in [T], \\ \Phi \leq T^{\rho}, \\ \Upsilon \leq T^{c}}} \frac{\mathbb{E}[R_T(\boldsymbol{\mathcal{D}}, \Upsilon, \mathcal{A})]}{T^{\beta}} = 0.$$

This definition stipulates that, under *any* choice of the adversary, constrained to inserting "few" anomalies and the system exhibiting "mild" distribution shifts, an Adaptive and Robust algorithm has sub-linear expected regret.

Definition 3.7 (Competitive in the Normal Stationary Setting)**.** An algorithm $\mathcal{A}$ is said to be competitive in the normal stationary setting, if when $X_t \sim \mathcal{D}$ are i.i.d. from the standard normal[1] , i.e., both $\Phi = \Upsilon = 0$, then for all $\varepsilon > 0$

$$\limsup_{T\to\infty} \sup_{\substack{\boldsymbol{\mathcal{D}} \text{ s.t.} \\ \mathcal{D}_i \in \mathcal{F}, \forall i \in [T]}} \frac{\mathbb{E}[R_T(\boldsymbol{\mathcal{D}}, 0, \mathcal{A})]}{T^{\frac{1}{2}+\varepsilon}} = 0.$$

We propose FITNESS and show that it is simultaneously adaptive, robust and competitive in the normal stationary setting.

4. Lower Bounds

We show that sub-linear total regret can only be achieved when the corruption complexity and the distribution shift complexity are sub-linear. Thus, the Definitions 3.5 and 3.6 are good benchmarks for practical algorithms to achieve.

Proposition 4.1. *Let $\mathcal{F}$ be the class of Gaussian distributions on the real line with unit variance. Suppose, $X_1, \cdots, X_T$ are from a Υ corrupted stream with all $\mathcal{D}_i := \mathcal{D} \in \mathcal{F}$, i.e., all time points having the same distribution, then there exists an universal constant c, such that*

$$\inf_{\mathcal{A}} R_T \geq c\frac{\Upsilon}{T}(T - \Upsilon),$$

holds[2] *with probability (over the random draws in the data-stream) at-least $2/3$.*

This result shows that sub-linear in T regret is achievable only if the total corruptions Γ is sub-linear in T.

Proposition 4.2. *For every $\zeta > 0$ and $T > 0$, there exists a family of distributions $\mathcal{F}$ with $\sup_{\mathcal{D}_i \in \mathcal{F}, \forall i \in [T]} \Phi(\mathcal{D}) \leq \zeta$, such that for any algorithm $\mathcal{A}$,*

$$\sup_{\mathcal{D}_i \in \mathcal{F}, \forall i \in [T]} \mathbb{E}[R_T(\boldsymbol{\mathcal{D}}, \mathbf{0}, \mathcal{A})] \geq \frac{1}{24} T^{2/3} \zeta^{1/3}$$

.

In words, the above proposition states that even in the absence of corruptions, there exists a family of problem instances where the expected regret scales at-least as $\Omega(T^{2/3}\Phi^{1/3})$. Thus, in order to expect sub-linear regret, we require the total distribution shift Φ to be sub-linear in T.

Proofs are given in Section 13, in the Supplementary Material. The two propositions reveal that in order to hope for sub-linear regret, the number of anomalies, and the distribution shift must be sub-linear in time.

5. Natural Approaches That Fail

We show that two natural approaches to estimating model parameters θ^* — *(i)* using all the data from a fixed sliding window, and *(ii)* using a dynamic window consisting of only points initially marked as non-anomalous, both fail to achieve the desiderata.

5.1. A Simplified Problem Setting

To glean intuition, we instantiate the problem with Gaussian distributions and L_2 loss. In the rest of this section, we will assume that the class $\mathcal{F}$ refers to the set of all Gaussian distributions with a fixed covariance matrix with matrix norm bounded by σ. For this family of distributions, the distribution complexity can be represented by $\Phi := \sum_{t=2}^{T} ||\mu_t - \mu_{t-1}||$ without loss of generality (Devroye et al., 2018). The set of Gaussian distributions are also closed under addition, i.e., the entire class $\mathcal{F}$ can be represented by a single Gaussian measure $\mathcal{D}$, such that for all $\mathcal{D}' \in \mathcal{F}$, there exists a deterministic vector $v \in \mathbb{R}^d$, such that if $X \sim \mathcal{D}$, then $X + v \sim \mathcal{D}'$.

In this simplified setting, the action set $\Theta = \mathbb{R}^d$ and the AD model class is given by $g(\theta, X) = \|\theta - X\|$. For

[1] One can generalize this to hold for any Gaussian measure

[2] The arguments from R_T are dropped whenever clear from context.

the L2 loss, it is well known that for any distribution $\mathcal{D}$ with finite mean, the mean vector satisfies $\mathbb{E}_{X\sim\mathcal{D}}[X] = \arg\min_{\theta\in\Theta}\mathbb{E}_{X\sim\mathcal{D}}[||\theta - X||]$. Thus, for this setting, $\theta_t^* = \mu_t$, for all $t \in \{1, \cdots, T\}$. As this model class is Lipschitz, the AD loss function $\mathcal{L}(\theta_1, \theta_2) = \|\theta_1 - \theta_2\|$.

We record here an useful property of the Gaussian distribution.

Lemma 5.1 (Theorem 3.1.1 (Vershynin, 2018))**.** *There exists an absolute positive constant $C > 0$ such that for every $\sigma > 0$, and for all $0 < \delta \leq 1$, we have*

$$\mathbb{P}_{X\sim\mathcal{D}}\left[\|X - \mathbb{E}X\| \geq C\sqrt{\sigma d}\log\left(\frac{1}{\delta}\right)\right] \leq \delta,$$

where $\mathcal{D}$ is any Gaussian measure on $\mathbb{R}^d$ with covariance matrix norm bounded by σ.

5.2. Sliding Window Algorithms with fixed windows

This class of algorithms has two hyper-parameters, $B \in \mathbb{N}$ the size of the window which determines the number of samples over which to compute the empirical mean, and $\lambda > 1$ is a parameter that controls when to declare a data point as anomalous. Concretely, the output at any time t is $\widehat{\mu}_t$, defined as

$$\widehat{\mu}_t = \begin{cases} \widetilde{\mu}_t^{(B)} & \text{if } \|\widetilde{\mu}_t^{(B)} - X_t\| \leq \lambda\sqrt{\frac{d\sigma}{B}}\log(T/\delta), \\ X_t & \text{otherwise,} \end{cases}$$

where $\widetilde{\mu}_t^{(B)} = \left(\frac{1}{\min(B,t)}\sum_{s=0}^{\min(t-1,B-1)} X_{t-s}\right)$. Pseudo code is provided in Algorithm 6, in the Supplementary Material.

Theorem 5.2. *If Algorithm 6 is run with parameters $B \in \mathbb{N}$, $\lambda > 1$ and $\delta \in (0, 1)$, then with probability at-least $1 - \delta$, the regret after T time-steps satisfies*

$$R_T \leq C\frac{T\sqrt{\sigma d}\log(T/\delta)}{\sqrt{B}} + B\Phi + \frac{(\lambda+1)B\Phi}{\left(\lambda - \frac{1}{\sqrt{B}} - 1\right)} + B\Upsilon(\lambda+1)C\sqrt{\sigma d}\log(T/\delta),$$

where C is the absolute constant in Definition 5.1.

5.2.1. Why is this bound unsatisfactory ?

The bound above yields sub-linear regret only when the window B is *small enough* such that $B\max(\Phi, \Gamma)$ is sub-linear in time.

Remark 5.3. Suppose $\max(\Phi, \Upsilon) = O(T^\alpha)$, where $\alpha < 1$. Then if Algorithm 6 is run with window $B = O(T^\gamma)$ where $\gamma = \frac{2}{3}(1 - \alpha)$, then the regret scales as $\widetilde{O}(\sqrt{d}T^{\frac{1}{3}(2+\alpha)})$. In other words, if an upper bound to the maximum drift or corruptions (the parameter α) is known apriori, then sub-linear regret is achievable, which matches upto logarithmic factors, the lower bound in Proposition 4.2.

However, if the parameter B is sub-linear in T, then the sliding window algorithm cannot be competitive with respect to stationary as shown in this proposition.

Proposition 5.4. *For any fixed window B, the regret of the sliding window algorithm on a sequence of i.i.d. 0 mean, unit variance Gaussians satisfies $\mathbb{E}[R_T] \geq \sqrt{\frac{2}{\pi}}\left(\frac{T-B}{\sqrt{B}} + \frac{\sqrt{B}}{2}\right)$.*

Thus, this algorithm cannot simultaneously achieve all the three desiderata of being robust to corruptions, drifts and competitive in a stationary environment.

5.3. Dynamically Growing Window

This class of algorithms produce as output, the empirical mean of a set of points, whose cardinality is non-decreasing with time. At any time t, we denote by $\mathcal{B}_t \subseteq \{X_1, \cdots, X_{t-1}\}$ as the **buffer** maintained by this algorithm at time t. This algorithm has one hyper-parameter $\lambda \geq 0$ which determines when a data point is not anomalous and thus can be added to the buffer. Concretely, at every time t, the algorithm outputs $\widehat{\mu}_t := \frac{1}{|\mathcal{B}_t|+1}\sum_{X\in\mathcal{B}_t\cup\{X_t\}} X$, based on the buffer contents at time t, $\mathcal{B}_t$. The buffer at time 1 is assumed to be empty-set. For the next time-step $t + 1$, a new buffer $\mathcal{B}_{t+1}$ is computed as follows

$$\mathcal{B}_{t+1} = \begin{cases} \mathcal{B}_t \cup \{X_t\} & \text{if } \|\widehat{\mu}_t - X_t\| \leq \lambda, \\ \mathcal{B}_t & \text{otherwise,} \end{cases}$$

In short, only points are not marked as anomalies at their time of arrival are added into the buffer. Thus, once a point is added to the buffer, it is never discarded. Conversely, once discarded, a point is never added back into the buffer.

Proposition 5.5. *For every $\lambda \geq 0$, there exists a distribution $\boldsymbol{\mathcal{D}} \in \mathcal{F}$ with $\Phi(\boldsymbol{\mathcal{D}}) = 4\lambda\log(4T^2)$, such that on the benign data stream with distribution $\boldsymbol{\mathcal{D}}$, $\inf_{\mathcal{A}}\mathbb{E}[R_T] \geq T/100$.*

Proof is in Section 19 of the Supplementary Material.

Remark 5.6. The above proposition shows that even in the absence of corruptions, it is important to discard past samples, similar to what the sliding window algorithm does.

6. The `FITNESS` Algorithm

From the previous discussions, we see that it is critical to both enlarge the window size in the case of stationary sequence to obtain higher statistical certainty, and discard old samples when they no longer represent the current distribution to improve adaptation. We present `FITNESS` in Algorithm 1 which is based on the principle to *estimate the*

mean from the largest set of recent samples that are relevant. This algorithm takes three inputs - σ the sub-Gaussian parameter, T the time horizon, and $\delta \in (0,1)$ denoting the *slack* parameter which controls the high probability guarantee with which our performance bounds hold.

Algorithm 1 FITNESS: GAUSSIAN

1: **input**: $\sigma \geq 0$, Slack parameter $\delta \in (0,1)$, Time horizon T, C as given in Definition 5.1
2: **for** each time $t \geq 1$ **do**
3: Receive Input $X_t \in \mathbb{R}^d$
4: $j \leftarrow 1$
5: **while** $\left\| \frac{1}{j} \sum_{s=0}^{j-1} X_{t-s} - X_t \right\| \leq C\left(1 + \frac{2}{\sqrt{j}}\right)\sqrt{d\sigma}\log\left(\frac{T^2}{\delta}\right)$ **do**
6: $j \leftarrow j+1$
7: **end while**
8: **Output** $\widehat{\mu}_t := \frac{1}{j}\sum_{s=0}^{j-1} X_{t-s}$
9: **end for**

6.1. Regret Guarantee for FITNESS

Definition 6.1. For every $t \in \{1,2,\cdots,T\}$ that is non-anomalous (i.e., $c_t = 0$), define $J^*(t)$ as

$$J^*(t) := \inf\Bigg\{ j \in \{1,2,\cdots,t\}, \text{s.t.} \left\| \mu_t - \frac{1}{j}\sum_{s=0}^{j-1}(\mu_{t-s}+c_{t-s}) \right\| > C\sqrt{\frac{d\sigma}{j}}\log\left(\frac{T^2}{\delta}\right) \Bigg\},$$

where inf of an empty set is defined as $J^*(t) := t+1$.

The time quantities are random, since the corruptions $\mathbf{c}$ chosen by the adversary can depend on the realization of the random vectors of the benign data stream $\widetilde{\mathbb{X}}$. In words, for every time t that is not corrupted by the adversary, $J^*(t)$ is the *first time instant while scanning backwards from* t, when the mean of the distribution at time t significantly deviates from the average of the means in the time-window $[J^*(t), t]$.

Theorem 6.2. *If Algorithm 1 is run with slack parameter $\delta \in (0,1)$, then with probability at-least $1-\delta$, the following regret bound holds*

$$R_T \leq \sum_{t=1}^{T} 2C\sqrt{\frac{d\sigma}{J^*(t)-1}}\log\left(\frac{T^2}{\delta}\right).$$

Proof is in Section 14 of the Supplementary Material. We now interpret the above result in certain special cases to demonstrate that it is adaptive and robust. Proofs of these corollaries are in Section 15 of the Supplementary Material.

6.2. FITNESS is Competitive in the Normal Stationary setting

Remark 6.3. Suppose, if all the T samples are i.i.d., i.e., $\Phi = 0$, then Algorithm 1 when run with slack parameter $\delta \in (0,1)$, achieves regret bounded by $R_T \leq 8C\sqrt{Td\sigma}\log(T/\delta)$, i.e., R_T is $\widetilde{O}(\sqrt{Td\sigma}\log(1/\delta))$[3], with probability at-least $1-\delta$. This can be seen since $J^*(t) := t+1$, for all t in this setting and substituting that in the formula in Theorem 6.2. Furthermore, in the Appendix in Lemma 14.5, we show that with probability at-least $1-\delta$ in this setting, at all times t, Algorithm 1 outputs the average of all samples till time t, i.e., $X_1, \cdots, X_t$.

In particular, when Algorithm 1 is run with slack parameter $\delta = 1/T$, we get $\mathbb{E}[R_T] = \widetilde{O}(\sqrt{Td\sigma})$ and thus satisfies the criterion of Definition 3.7.

6.3. FITNESS is Adaptive in the case of steady drift and absence of corruptions

Corollary 6.4. *Suppose, for all $t \in \{1,\cdots,T-1\}$, $\|\mu_t - \mu_{t+1}\| \leq \frac{\Phi}{T}$ for some $\Phi > 0$, i.e., the sequence exhibits "smooth" variation. Then Algorithm 1 achieves regret satisfying $R_T \leq \left(C\sqrt{d\sigma}T\log\left(\frac{T^2}{\delta}\right)\right)^{2/3}\Phi^{1/3}$, i.e., $R_T = \widetilde{O}((\sqrt{d}T)^{2/3}\Phi^{1/3}\log(1/\delta))$, with probability at-least $1-\delta$. This regret matches (upto logarithmic factors), the minimax lower bound given in Proposition 4.2, without knowing the drift parameter.*

6.4. FITNESS is Robust to Anomalies

Corollary 6.5. *Suppose there no drifts, i.e., $\Phi = 0$, and the corruptions are such that $\Upsilon = T^\alpha$ for some $0 \leq \alpha < 1$. Suppose further that, if time t is corrupted, then $\|c_t\| > \sqrt{d\sigma}\log\left(\frac{T^2}{\delta}\right)$, i.e., the corruptions if they occur are 'large'. Then, with probability at-least $1-\delta$, the regret satisfies $R_T \leq 16CT^{\frac{1+\alpha}{2}}\sqrt{d\sigma}\log\left(\frac{T^2}{\delta}\right)$. In particular, the algorithm is robust.*

6.5. The key idea in FITNESS

The key novelty in FITNESS is to introduce j in the RHS of Line 5 in Algorithm 1, which captures the intuition that as more samples from the past are averaged, the concentration around the test sample X_t is sharper. On the one hand, estimating the mean from a larger history is essential for the estimate to concentrate around the true mean. However, samples farther past in time could be from a distribution far from the one generating the current sample. The condition in Line 5 of Algorithm 1 balances these competing objectives.

[3] $\widetilde{O}(\cdot)$ suppresses logarithmic dependence on T

6.6. Comparison with offline robust mean estimation

The offline robust mean estimation problem consists of estimating the mean of a distribution from n i.i.d., samples, of which at-most εn of them can be adversarially corrupted. Even in this simpler setup, understanding of computationally efficient estimators is an active area of current research (c.f. survey (Diakonikolas and Kane, 2019a)). One class of algorithms for the offline robust mean estimation involve a careful pruning of potentially corrupted samples based on the spectrum of the empirical covariance matrix, and computing the empirical mean on the remaining samples.

The design of FITNESS bears resemblance to this class of methods, where the samples on which the mean is estimated are chosen in Line 5 of Algorithm 1. However, the relevant samples in FITNESS are chosen using a simple empirical mean test, without resorting to computing spectrum of the empirical covariance matrix. The algorithm in the online case is seemingly easier than the offline case because the online problem estimates the mean *at a given time point* with a reference sample X_t. Thus, the online algorithm leverages this given sample X_t to identify other non-corrupted samples that come from a 'similar' distribution and compute the empirical mean. The computational complexity at time t is $O(td)$ is linear in t and matches the best known offline algorithms (Diakonikolas and Kane, 2019a).

6.7. Known σ assumption in FITNESS

FITNESS depends on the knowledge of σ. In practice and our experiments, we modify the algorithm to estimate σ from the data (Algorithm 2). From a theoretical perspective, the problem of adaptive algorithms in the presence of known σ is already non-trivial. Indeed, the offline robust-mean estimation algorithms rely on the fact that either σ is known, or that all the non-corrupted samples are from the same distribution, i.e., there is no drift (c.f. (Diakonikolas and Kane, 2019a)). Developing provably low-regret adaptive algorithms with unknown σ in the presence of drifts and corruptions is a challenging open problem left to future work.

6.8. Known time horizon T assumption

This assumption is standard and can be relaxed by incurring an additional log(T) regret by the doubling-trick Sec 2.3 (Cesa-Bianchi and Lugosi, 2006). We include this for completeness in Algorithm 5 in Appendix Section 15.3.

7. FITNESS: GENERAL Algorithm

FITNESS: GAUSSIAN has two steps - *(i)* identifying recent *similar* points, and *(ii)* estimating θ_t^*, based on the identified samples. We generalize these two steps to create FITNESS GENERAL in Algorithm 2. FITNESS: GENERAL has two main sub-routines given in Algorithms 3 and 4. The main result we will show through experiments, is that FITNESS: GENERAL can convert a base anomaly scoring algorithm into a streaming one which is adaptive and robust.

7.1. Identifying recent similar samples (Algorithm 3)

In the general case, we replace the known variance bound, by the robust empirical covariance (Rousseeuw, 1984) of the window in consideration (Line 3 of Algorithm 3). The intuition is that the estimated covariance from samples is close to the true covariance, even when there are a few anomalous/corrupted points.

7.2. Fine-tuning on the identified similar samples (Algorithm 4)

In the general case, we replace the empirical mean of the samples with one step of an optimizer OPT or model parameter update mechanism. An effect of FITNESS: GENERAL not imposing any restrictions on the anomaly scoring models given as input is that it requires an appropriate incremental optimization procedure OPT as an input. In the experiments, we consider two instantiations of FITNESS GENERAL; *(i)* is with the class of anomaly scoring models $(g(\theta, \cdot))_{\theta \in \Theta}$ being the Lipschitz auto-encoder of (Kim et al., 2020), for which the optimizer OPT is the Adam (Kingma and Ba, 2014) optimizer with learning rate $2e-4$, and *(ii)* is with the class of anomaly scoring models being the Extended Isolation Forest (Hariri et al., 2019), with the optimizer OPT as the iForest algorithm from (Hariri et al., 2019).

7.3. Heuristics for Scalability

In the worst-case, it takes $O(t)$ time to complete the scoring of the sample at time t, which is expensive in some settings. Our heuristic is to 'search' for the correct window sizes from among a fixed set of choices $\{W_1, \cdots, W_k\}$, say powers of 2. For each window, we compute the test (Line 5 of Algorithm 4) and select the largest window passing the test. Details in Section 12.3 in Supplementary Material. Unlike the sliding window method however, FITNESS: GENERAL uses all historical data - as the estimate θ_t at time t is fine-tuned by using the estimate θ_{t-1} at time $t-1$ as a warm-start. Line 4 in Algorithm 2 sends θ_{t-1} as input to the update. Although we propose heuristics for scalability, provably achieving $O(1)$ complexity per-time step even in the Gaussian case is an open problem.

8. Synthetic Experiments

The main result of this Section is in Figure 1, where we observe that FITNESS: GAUSSIAN is adaptive and robust

Algorithm 2 FITNESS: GENERAL

Input: $(g(\theta,\cdot))_{\theta\in\Theta}$, a family of anomaly scoring models, OPT an optimizer
for each time $t \geq 1$, receive $X_t \in \mathbb{R}^d$ and **do**
  $\mathcal{B}_t \leftarrow$RELEVANT-HISTORY$(\{Y_s\}_{s\leq t})$ {Algo 3}
  $\theta_t \leftarrow$ UPDATE-MODEL$(\mathcal{B}_t, \theta_{t-1}, \texttt{OPT})$ {Algo 4}
  Output Anomaly Score: $S_t \leftarrow g(\theta_t, Z_t)$
end for

Algorithm 3 GET-RELEVANT-HISTORY$(\{Y_s\}_{s\leq t})$

Input $\{Y_s\}_{s\leq t}$ - Input data points ordered by time
Hyper-parameters $C_1 > 0$
$j \leftarrow 1$
$\widehat{\mu}_t^{(j)} \leftarrow \frac{1}{j}\sum_{s=0}^{j-1} Y_{t-s}$
$\widehat{\Sigma}_t^{(j)} \leftarrow$COVARIANCE$((Y_{t-s})_{s=0}^{j-1})$ {The robust covariance algorithm of (Rousseeuw, 1984)}
while $\|\widehat{\Sigma}^{-\frac{1}{2}}(\widehat{\mu}_t^{(j)} - Y_t)\| \leq C_1\sqrt{p}\left(1+\frac{1}{\sqrt{j}}\right)$ **do**
  $j \leftarrow j+1$
  Recompute $\widehat{\mu}_t^{(j)}$ and $\widehat{\Sigma}_t^{(j)}$ from Lines 3 and 4 above.
end while
Return $\{Y_{t-s}\}_{s=0}^{j-1}$

and outperforms any fixed sliding window algorithm. This complements our regret upper bounds we derived theoretically to demonstrate that the benefits are not just artefacts of mathematical bounds, but also hold in practice. In Figure 1, the true mean at time t was set equal to $\sqrt{t}$, i.e., the drift complexity over a time horizon T is $\sqrt{T}$. For corruptions we randomly selected 2% of the time points and perturbed them with random values drawn from a 0 mean Gaussian with variance $\sqrt{T}$, i.e., the perturbations are typically large positive or negative samples. The plot in Figure 1 plots the cumulative regret versus the time-horizon. For this fixed set of means, each different algorithms was run 50 times, and

Algorithm 4 UPDATE-MODEL

Input: Dataset $\mathcal{B} := (Y_i)_{i=1}^n$, Θ solution space, loss function $\mathcal{L} : \mathbb{R}^p \times \Theta \to \mathbb{R}_+$, $\theta_0 \in \Theta$ an initial guess, OPT an optimization algorithm
Hyper-parameters maxiter $\in \mathbb{N}$
$\theta^{(0)} \leftarrow \theta_0$,
$iter \leftarrow 1$
Initialize OPT
while $iter <$ maxiter **do**
  $\theta^{(iter+1)} \leftarrow$OPT.STEP$(\theta^{(iter)})$ {Optimize starting at prev model}
  $iter \leftarrow iter + 1$
end while
Return $\theta^{(\text{maxiter})}$

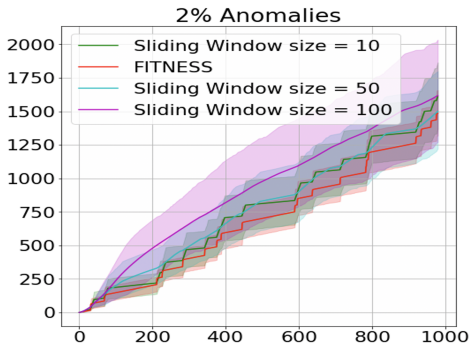


Figure 1: Comparision of cumulative regret of Fitness with sliding windows.

Dataset	dim	# samples	anomalies
Thyroid	6	3772	2.5%
Satellite	36	6435	20.6%
Kitsune	115	764136	16%
Telemetry	300	108400	1.2%

Table 1: Description of benchmark Datasets

the average regret and the resulting 95% confidence interval are plotted in the solid and shaded regions respectively. The different runs average over the random samples X_t drawn from the distribution at time t.

We observe in Figure 1 that FITNESS: GAUSSIAN outperforms any fixed sliding window algorithm. We also conduct synthetic experiments on FITNESS: GENERAL in the Appendix in Section 11, where we show across a variety of situations, a fixed, common value for C_1 (used in Algorithm 3) works well. We then use this in the real data experiments.

9. Real Data Experiments

The main result of this section is to show that FITNESS is **flexible** in practice. In particular, we show that FITNESS: GENERAL can be used both with deep-learning based Lipschitz Autoencoder AD model (Kim et al., 2020), as well as non deep-learning based Extended Isolation Forest (Hariri et al., 2019) as the family of anomaly scoring models fed as input to in Algorithm 2. This demonstrates that FITNESS is a *flexible framework*, that can take as input any existing anomaly scoring models, and produce an online AD algorithm. Moreover, we observe on several real datasets that, the best performance is consistently achieved by either FITNESS with the Autoencoder as the anomaly scoring model, or FITNESS with EIF as the anomaly scoring class. This indicates that FITNESS can convert any anomaly scoring class into an online AD algorithm that is robust to distribution shifts.

Algo/Dataset	Thyroid	Satellite (rand)	Satellite (sort)	IoT Attack	Telemetry
FITNESS ADAE	0.13 ± 0.05	0.24 ± 0.01	0.29 ± 0.02	0.86 ± 0.01	$\mathbf{0.25 \pm 0.05}$
FITNESS EIF	$\mathbf{0.28 \pm 0.01}$	$\mathbf{0.53 \pm 0.03}$	$\mathbf{0.40 \pm 0.02}$	$\mathbf{0.96 \pm 0.02}$	0.01 ± 0.006
LODA	0.02 ± 0.001	0.21 ± 0.001	0.21 ± 0.001	0.84 ± 0.001	0.0
xSTream	0.09 ± 0.04	0.32 ± 0.02	0.30 ± 0.04	0.73 ± 0.03	0.07 ± 0.01
HS Trees	0.05 ± 0.02	0.342 ± 0.007	0.33 ± 0.001	0.74 ± 0.06	0.05 ± 0.01
MStream	0.23 ± 0.03	0.34 ± 0.02	0.25 ± 0.003	0.93 ± 0.001	0.02 ± 0.002
RSHash	0.01 ± 0.001	0.22 ± 0.04	0.17 ± 0.03	$\mathbf{0.98 \pm 0.004}$	$\mathbf{0.2 \pm 0.04}$
RCF [Offline]	0.26	0.29	0.29	0.82	0.09 ± 0.01
Extended IF [Offline]	0.31 ± 0.08	0.56 ± 0.05	0.56 ± 0.05	0.69	0.02 ± 0.002

Table 2: Table comparing the APS score. Higher is better. In each column, the best online score is marked in bold.

9.1. Benchmark Datasets

Public Datasets: We consider 3 public datasets, covering the spectrum in dimensions and number of samples. Satellite and Thyroid is from (Rayana, 2016) and IoT data is the mirai attack from (Mirsky et al., 2018).

Multi-server Telemetry Data: We also compared performance on multi-variate telemetry dataset from a multi-server cloud service. Each data-point contains categorical and numerical features. The categorical features were processed by StreamHash (Manzoor et al., 2018) and each data-point converted into a vector of 300 dimensions. We insert anomalies into this data and evaluate different algorithms.

Data characteristics are in Table 1 and the pre-processing details in the Appendix in Section 12.

9.2. Performance Metric

All algorithms we consider are unsupervised, and we only use the labels for evaluation. Note that every algorithm outputs a score for each data point, which must be thresholded to obtain the final list of anomalies. Different choices of thresholds lead to different precision and recall. In order to measure performance of anomaly scoring algorithms, we use the *Averaged-Precision-Score* (APS), which measures the area under the curve of Precision versus Recall obtained by sweeping over all possible thresholds (Pedregosa et al., 2011). Consistent with recent work in anomaly detection, we use APS and not the alternative AUROC since *(i)* the area under the ROC is not necessarily equivalent to optimizing area under the PR curve (Davis and Goadrich, 2006) and *(ii)* performance of practical anomaly detection systems are measured using precision and recall values.

9.3. Benchmark Algorithms

Models used in FITNESS: We instantiate FITNESS with two AD model classes $(g(\theta, \cdot))_{\theta \in \Theta}$ fed as input to Algorithm 2 - the Lipschitz Auto-encoder of (Kim et al., 2020) termed as FITNESS ADAE, and isolation forest (Hariri et al., 2019) termed as FITNESS EIF. In both cases, we choose the same set of hyper-parameters and do not tune per data-set. This value is chosen by experimenting on synthetic data (Section 11 in the Supplementary Material).

Competing Algorithms: We compare against state-of-art, unsupervised, online AD algorithms such as MSTREAM (Bhatia et al., 2021a), xSTream (Manzoor et al., 2018), LODA (Pevnỳ, 2016), RSHash (Sathe and Aggarwal, 2016) and HSTrees (Tan et al., 2011). The implementations of all of these except MSTREAM was taken from (Yilmaz and Kozat, 2020). In addition, we also compare with two offline, unsupervised AD algorithms, RCF (Guha et al., 2016) from (rrc, 2019) and ExtendedIF (Hariri et al., 2019).

9.4. Results from Experiments

We observe from Table 2, that FITNESS obtained superior performance compared to other competing online algorithms. In the IoT dataset, FITNESS exceeds even the offline benchmark's performance. The reason for this is that the fraction of anomalies in that dataset is large, thereby skewing the performance of the offline benchmark.

10. Conclusions and Future Work

In our theoretical analysis we identify the complexity parameters and formalize desiderata that are expected of online AD algorithms. We prove that FITNESS is able to achieve the desiderata in the case when the underlying distributions are sub-gaussian, and the loss function is L_2. For the general case we propose a heuristic motivated by our analysis, and empirically show that FITNESS is useful on real datasets. In practical applications of AD systems, computational and memory complexity are as important as statistical accuracy (Mirsky et al., 2018; Bhatia et al., 2021a). We propose heuristics in Sections 7.3 to allow experimentation on large datasets in Section 9. A formal study of the impact of and trade-offs between compute and memory on statistical performance is an important direction of future research.

Acknowledgements We thank Sriram Manohar and Jeremiah Wilton for useful discussions in formulating the problem. We thank the reviewers for their positive and constructive feedback which substantially improved the paper.

References

(2019). RRCF: Implementation of the Robust Random Cut Forest algorithm for anomaly detection on streams. *The Journal of Open Source Software*, 4(35):1336.

Ahmed, M., Mahmood, A. N., and Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. (2020). USAD: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404.

Bhatia, S., Jain, A., Li, P., Kumar, R., and Hooi, B. (2021a). MStream: Fast anomaly detection in multi-aspect streams. In *Proceedings of the Web Conference 2021*, pages 3371–3382.

Bhatia, S., Jain, A., Srivastava, S., Kawaguchi, K., and Hooi, B. (2021b). MEMSTREAM: Memory-based anomaly detection in multi-aspect streams with concept drift. *arXiv preprint arXiv:2106.03837*.

Bifet, A. and Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM.

Bifet Figuerol, A. C. and Gavaldà Mestre, R. (2009). Adaptive parameter-free learning from evolving data streams.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58.

Cheng, Y., Diakonikolas, I., Ge, R., and Soltanolkotabi, M. (2020). High-dimensional robust mean estimation via gradient descent. In *International Conference on Machine Learning*, pages 1768–1778. PMLR.

Cheng, Y., Diakonikolas, I., Ge, R., and Woodruff, D. P. (2019). Faster algorithms for high-dimensional robust covariance estimation. In *Conference on Learning Theory*, pages 727–757. PMLR.

Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2019). Learning to optimize under non-stationarity. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1079–1087. PMLR.

Chu, F., Wang, Y., and Zaniolo, C. (2004). An adaptive learning approach for noisy data streams. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 351–354. IEEE.

Daneshmand, H., Lucchi, A., and Hofmann, T. (2016). Starting small-learning with adaptive sample sizes. In *International conference on machine learning*, pages 1463–1471. PMLR.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654.

Devroye, L., Mehrabian, A., and Reddad, T. (2018). The total variation distance between high-dimensional gaussians. *arXiv preprint arXiv:1810.08693*.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2017). Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008. PMLR.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. (2018). Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2683–2702. SIAM.

Diakonikolas, I. and Kane, D. M. (2019a). Recent advances in algorithmic high-dimensional robust statistics. *arXiv preprint arXiv:1911.05911*.

Diakonikolas, I. and Kane, D. M. (2019b). Recent advances in algorithmic high-dimensional robust statistics. *arXiv preprint arXiv:1911.05911*.

Ding, Z. and Fei, M. (2013). An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes*, 46(20):12–17.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. (2015). Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763. PMLR.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.

Guha, S., Mishra, N., Roy, G., and Schrijvers, O. (2016). Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pages 2712–2721. PMLR.

Gupta, M., Gao, J., Aggarwal, C. C., and Han, J. (2013). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering*, 26(9):2250–2267.

Hariri, S., Kind, M. C., and Brunner, R. J. (2019). Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*.

Heigl, M., Anand, K. A., Urmann, A., Fiala, D., Schramm, M., and Hable, R. (2021). On the improvement of the isolation forest algorithm for outlier detection with streaming data. *Electronics*, 10(13):1534.

Hill, D. J. and Minsker, B. S. (2010). Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9):1014–1022.

Huang, H. and Kasiviswanathan, S. P. (2015). Streaming anomaly detection using randomized matrix sketching. *Proceedings of the VLDB Endowment*, 9(3):192–203.

Kim, Y.-g., Kwon, Y., Chang, H., and Paik, M. C. (2020). Lipschitz continuous autoencoders in application to anomaly detection. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2507–2517. PMLR.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Laxhammar, R. and Falkman, G. (2013). Online learning and sequential anomaly detection in trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 36(6):1158–1173.

Liu, C., Hoi, S. C., Zhao, P., and Sun, J. (2016). Online arima algorithms for time series prediction. In *Thirtieth AAAI conference on artificial intelligence*.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

Manzoor, E., Lamba, H., and Akoglu, L. (2018). XSTREAM: Outlier detection in feature-evolving data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1963–1972.

Martí, L., Sanchez-Pi, N., Molina, J. M., and Garcia, A. C. B. (2015). Anomaly detection based on sensor data in petroleum industry applications. *Sensors*, 15(2):2774–2797.

Mirsky, Y., Doitshman, T., Elovici, Y., and Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

Miyaguchi, K. and Kajino, H. (2019). Cogra: Concept-drift-aware stochastic gradient descent for time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4594–4601.

Na, G. S., Kim, D., and Yu, H. (2018). Dilof: Effective and memory efficient local outlier detection in data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1993–2002.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pevnỳ, T. (2016). LODA: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304.

Pokrajac, D., Lazarevic, A., and Latecki, L. J. (2007). Incremental local outlier detection for data streams. In *2007 IEEE symposium on computational intelligence and data mining*, pages 504–515. IEEE.

Raab, C., Heusinger, M., and Schleif, F.-M. (2020). Reactive soft prototype computing for concept drift streams. *Neurocomputing*, 416:340–351.

Rayana, S. (2016). Odds library.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*.

Sathe, S. and Aggarwal, C. C. (2016). Subspace outlier detection in linear time with randomized hashing. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 459–468. IEEE.

Saurav, S., Malhotra, P., TV, V., Gugulothu, N., Vig, L., Agarwal, P., and Shroff, G. (2018). Online anomaly detection with concept drift adaptation using recurrent neural networks. In *Proceedings of the acm india joint international conference on data science and management of data*, pages 78–87.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Shekhar, S., Shah, N., and Akoglu, L. (2021). FairOD: fairness-aware outlier detection. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 210–220.

Tahmasbi, A., Jothimurugesan, E., Tirthapura, S., and Gibbons, P. B. (2021). Driftsurf: Stable-state/reactive-state learning under concept drift. In *International Conference on Machine Learning*, pages 10054–10064. PMLR.

Tan, S. C., Ting, K. M., and Liu, T. F. (2011). Fast anomaly detection for streaming data. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Togbe, M. U., Chabchoub, Y., Boly, A., Barry, M., Chiky, R., and Bahri, M. (2021). Anomalies detection using isolation in concept-drifting data streams. *Computers*, 10(1):13.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Wu, K., Zhang, K., Fan, W., Edwards, A., and Philip, S. Y. (2014). RS-forest: A rapid density estimator for streaming anomaly detection. In *2014 IEEE International Conference on Data Mining*, pages 600–609. IEEE.

Xu, H., Chen, W., Zhao, N., Li, Z., Bu, J., Li, Z., Liu, Y., Zhao, Y., Pei, D., Feng, Y., et al. (2018). Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196.

Yilmaz, S. F. and Kozat, S. S. (2020). PySAD: A streaming anomaly detection framework in python. *arXiv preprint arXiv:2009.02572*.

Zhang, L., Lin, J., and Karim, R. (2016). Sliding window-based fault detection from high-dimensional data streams. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(2):289–303.

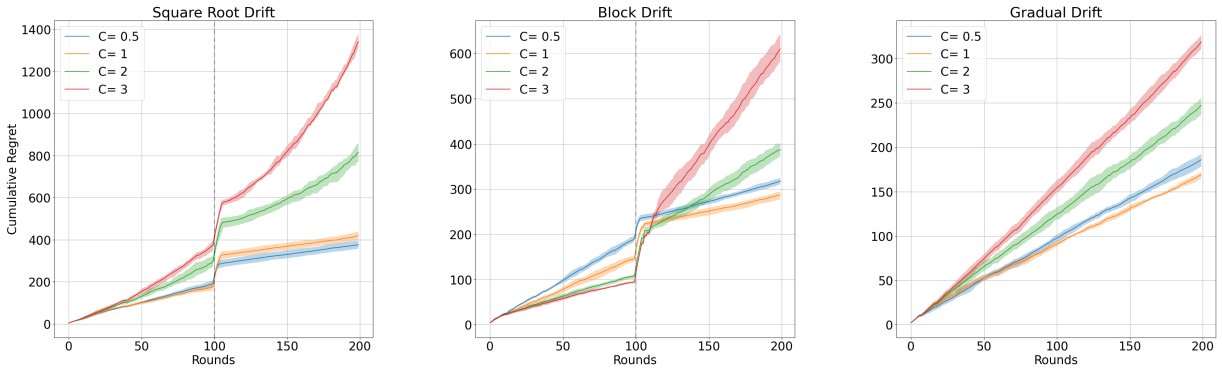
Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

APPENDIX

11. Synthetic Data Experiments

11.1. Hyper-parameter sensitivity in FITNESS

In this section, we show that FITNESS GENERAL has good performance in multiple situations, with a single universal value of the hyper-parameter $C_1 = 1$ of Algo 3 (used in Line 3 of Algo 2). The hyper-parameter C_1 plays the role of σ, which was assumed as an input to Algorithm 1. Observe that if C_1 is large, then the cardinality of the dataset $\mathcal{B}$ in Line 4 of Algo 2 will be large running the risk of containing old samples with distributions different than that producing the current sample X_t. However, if C_1 is small, then the dataset $\mathcal{B}$ used in Line 4 of Algo 2 will be small leading to large variance in estimation.



(a) (b) (c)

Figure 2: Evolution of Cumulative regret, where the changepoint occurs in the middle with change of mean and variance.

Impact of C_1 under drift In Figure 2, we considered a data-stream of 5 dimensional vectors with only drift, but no outliers. In Figure 2 (a), the sample X_t, for $t \leq 100$ was sampled from $\mathcal{N}(\sqrt{t}\mathbf{1}, 3 * \mathbb{I})$, while for $t > 100$, $X_t \sim \mathcal{N}(-\sqrt{t}\mathbf{1}, \mathbb{I})$. Thus, this situation consists of both a gradual drift, as well as an abrupt mean and variance change at time 100. In Figure (b), consisted of two i.i.d. segments, for $t \leq 100$, $X_t \sim \mathcal{N}(0, \mathbb{I})$ and for $t > 100$, $X_t \sim \mathcal{N}(0, 2 * \mathbb{I})$. In Figure 2 (c), there was no abrupt shift, but a slowly varying change where $X_t \sim \mathcal{N}(\log(1 + t), \mathbb{I})$. In all the three cases, we observe that setting $C_1 = 1$, obtains the smallest regret scaling.

Impact of C_1 under drift and outliers - In Figure 3, the top plot shows the signed mean of the vector X_t at time t, and the bottom plot shows the instantaneous regret achieved by algorithms with different C_1. Concretely, the sample X_t at time $t < 100$ was sampled from $\mathcal{N}(-\sqrt{t}\mathbf{1}, 2\mathbb{I})$, for $100 \leq t < 200$, $X_t \sim \mathcal{N}(\sqrt{t}\mathbf{1}, 3 * \mathbb{I})$, for $200 \leq t < 300$, $X_t \sim \mathcal{N}(\sqrt{t}, \mathbb{I})$ and for $t \geq 300$, $X_t \sim \mathcal{N}(\sqrt{t}, 2\mathbb{I})$. In addition, at each time, we tossed a coin with probability 0.05, and set it as an outlier. Any point that was set as an outlier was set as $X_t = 20\mathbf{1}$. The times of an outlier are indicated by the red dashed vertical lines in Figure 3. The black vertical lines represent the abrupt change points. From the Figure in 3, we can observe that for $C_1 = 1$, the trade-off between adaptivity and reduction of estimation variance is optimal. A higher C_1 leads to a lot more fluctuations due to the outliers while a lower value of C_1 suffers from high variance in estimation.

Figure 3: Variation of Instantaneous Regret with different parameter C.

12. Additional Details on Real Data Experiments

12.1. Public Datasets

- Thyroid (Dua and Graff, 2017) - This low-dimensional AD dataset with 6 real valued attributes and three classes. The minority class is treated as anomalous. More details on this dataset is in (Rayana, 2016).
- Kitsune Network Attack Dataset (Mirsky et al., 2018). This is a dataset collected from packet capture on the network interface of real IoT devices. The devices initially see a benchmark benign traffic, and a mirai attack is carried out at some point during the data collection. The captured packets are then processed into 115 real valued features. All packet during the attack duration are marked anomalous.
- Satellite (Rayana, 2016) is a multi-class (6 classes) classification dataset, where classes 2 and 4 are defined as anomalies.

12.2. Data Pre-processing

Thyroid (Rayana, 2016) is an open-source anomaly detection dataset, where the order in which the points were streamed was randomly shuffled. The reported results in Table 2 are the average and standard deviation from 10 runs. *Satellite* (Rayana, 2016) is a multi-class classification dataset with 6 target classes. We set class 2 and 4 as the anomalous ones, and the rest as non-anomalous. We use this dataset to conduct two experiments. In Satellite (rand) (3rd col in Table 2), the entries of the dataset are streamed one at a time in a random order. The results reported in col 3 of Table 2 are averaged over 10 different random ordering of the dataset. In Satellite (sort) (4th col in Table 2), the entries were streamed in order of class labels - first was all points corresponding to class 1, followed by 2 and so on. The result reported in col 4 of Table 2 were average after 3 runs in this sorted order. The offline benchmark, by definition for both these are the same. The difference in online performance on these two settings indicate the degree to which the algorithms are adaptive and robust. This methodology is commonly used in evaluating streaming AD algorithms (Wu et al., 2014).

IoT Network data is the mirai attack data from (Mirsky et al., 2018). This dataset consists of processed packet features that are captured sequentially in time. At some point of time, an attack gets executed and all points in the attack duration are

marked as anomalous. The *telemetry data* is also an attack data, ordered by time. As both datasets are ordered by time, we use this order for the streaming experiments, and report the results after 3 runs.

12.3. Hyper-parameter choices in Table 2

Each algorithm (both ours and benchmark) was given one hyper-parameter choice across all datasets. For the benchmark algorithms, we used the default hyper-parameters given in the implementations (Yilmaz and Kozat, 2020) and (Bhatia et al., 2021a). For `FITNESS` we project data down to 36 dimensions (in-case the input data has dimension larger than 36), by choosing R to be a matrix where each row consists of i.i.d. Gaussian random variables drawn from 0 mean, and equal variance such that the variance of the norm is 1. We do this, as this is known to reduce dimension while preserving pair-wise distances (Bingham and Mannila, 2001). For Algorithm 3, we use $C_1 = 1$, and search over window sizes ranging from 0 to 2000, skipping over 100 at a time, i.e., test at $0, 100, 200, \cdots, 2000$. In case there are not enough history (i.e., less than 2000), we take the full history as one of the window sizes to test. For `FITNESS ADAE`, the network architecture we use the KDD-Model defined in (Kim et al., 2020) with code for implementation in *https://github.com/kyg0910/Lipschitz-Continuous-Autoencoders-in-Application-to-Anomaly-Detection/blob/master/src/configuration.py*. We use the default parameters given there for the batch size, learning rate and optimization algorithm and fine-tune for 1 step, i.e, `max-iter` of Algorithm 4 was set to 1. For Extended IF, we use `n-trees` as 200 and sample size of 256.

13. Lower Bound Proofs from Section 4

We will prove the results for the simplified problem setting defined in Section 5.1.

Proof of Proposition 4.1. The lower bound follows from the key fact that two unit variance Gaussian that have means with distance ε, has a total variation distance of order ε.

$$
\begin{aligned}
&\inf_{\mathcal{A}} \sup_{\substack{\boldsymbol{\mathcal{D}} \text{ s.t.} \\ \min(\Psi,\Phi)\leq T^{\rho}, \\ \Upsilon=\varepsilon T}} R_T \\
&\geq \inf_{\mathcal{A}} \sup_{\substack{\boldsymbol{\mathcal{D}} \text{ s.t.} \\ \max(\Psi,\Phi)\leq 0, \\ \Upsilon=\varepsilon T}} R_T, \\
&\geq \inf_{\mathcal{A}} \sup_{(\mathcal{D}_i)_{i=1}^T \sim \mathcal{N}(\mu,1), \atop \Upsilon=\varepsilon T} \sum_{t=1, t\notin\Lambda_T}^{T} ||\widehat{\mu}_t - \mu||, \\
&\geq \inf_{\mathcal{A}} \sup_{(\mathcal{D}_i)_{i=1}^T \sim \mathcal{N}(\mu,1), \atop \Upsilon=\varepsilon T} (T-\varepsilon T) \min_{t\notin\Lambda_T} ||\widehat{\mu}_t - \mu||, \\
&\overset{(a)}{\geq} \inf_{\mathcal{A}} \sup_{(\mathcal{D}_i)_{i=1}^T \sim \mathcal{N}(\mu,1), \atop \Upsilon=\varepsilon T} (T-\varepsilon T) ||\widehat{\mu}_T - \mu||, \\
&\overset{(b)}{\geq} c\varepsilon(T-\varepsilon T), \text{ with probability at-least } 2/3.
\end{aligned}
$$

Step (a) follows from the fact that the regret can be no smaller than if the AD waited to see all samples before predicting an anomaly score. Step (b) follows from Fact 1.2 of (Diakonikolas and Kane, 2019b). □

Proof of Proposition 4.2. The proof strategy follows that of the lower bound (Cheung et al., 2019). We will construct a family of one dimensional problem instances $\mathcal{F}$, such that the regret is lower-bounded for any estimation algorithm.

In order to do so, fix a $H \in [0, T]$ and divide the time interval into $\lceil \frac{T}{H} \rceil$ blocks with each block lasting for H time-slots, except for the last one. We consider the family $\mathcal{F}$ to be the set of one dimensional Gaussian distributions with unit variance, and the means at any point of time taking values in the binary set $\left\{\pm\frac{1}{4\sqrt{H}}\right\}$. Thus the family $\mathcal{F}$ contains two distributions. Moreover, the means are unchanging in each block, i.e., for all $i \in \left[\lceil \frac{T}{H} \rceil\right]$ and $t_1, t_2 \in [(i-1)H + 1, \min(iH, T)]$, $\mu_{t_1} = \mu_{t_2} \in \left\{\pm\frac{1}{4\sqrt{H}}\right\}$.

Moreover, the algorithm $\mathcal{A}$ is privy to this information. In any block $i \in \left[\lceil \frac{T}{H} \rceil\right]$, the estimation error at the end of the block is lower bounded by $\frac{1}{12\sqrt{H}}$ (Example 15.4 in (Wainwright, 2019)), i.e.,

$$\mathbb{E}[r_{\min(iH,T)}] \geq \frac{1}{12\sqrt{H}}.$$

The instantaneous regret inside any block, is trivially lower bounded by the instantaneous regret at the end of the block, i.e., for any $i \in \left[\lceil \frac{T}{H} \rceil - 1\right]$ and $t \in [(i-1)H+1, \min(iH,T)]$, $r_t \geq \frac{1}{12\sqrt{H}}$. Moreover as the blocks are independent, there is no information across blocks, i.e., the minimax error across blocks are independent. Thus, the total cumulative regret is lower bounded by

$$\begin{aligned}
\mathbb{E}[R_T] &\geq \sum_{i=0}^{\lceil \frac{T}{H} \rceil - 1} \sum_{t \in [(i-1)H+1, \min(iH,T)]} \frac{1}{12\sqrt{H}}, \\
&\geq \frac{1}{12}\sqrt{H}\left(\frac{T}{H} - 1\right), \\
&= \frac{T}{12\sqrt{H}} - \frac{\sqrt{H}}{12}. \qquad (1)
\end{aligned}$$

Now, we need to see what is the most number of blocks H one can have that adheres to the given variation budget. Observe that the total variation between two unit variance Gaussians with means μ_1 and μ_2 is equal to $\frac{1}{2}\|\mu_1 - \mu_2\|$. Note that the maximum difference between the means between two blocks is $\frac{1}{2\sqrt{H}}$. Thus, the maximum possible variation in the family $\mathcal{F}$ is $\frac{1}{2\sqrt{H}}\lceil \frac{T}{H} \rceil \leq \frac{T}{H^{3/2}}$. Thus, if

$$\frac{T}{H^{3/2}} \leq \frac{1}{2}\zeta,$$

then the maximum variation in the family $\mathcal{F}$ is within ζ. The $1/2$ in the RHS of the previous display is because the total variation between two unit variance Gaussians is $\frac{1}{2}\|\mu_1 - \mu_2\|$.

Re-arranging the preceding display, we get that we need $H \geq 2^{2/3}T^{2/3}\zeta^{-2/3}$. Substituting this in Equation (1), we get that

$$\begin{aligned}
\mathbb{E}[R_T] &\geq \frac{1}{12}T^{2/3}\zeta^{1/3} - \frac{1}{12}T^{1/3}\zeta^{-1/3}, \\
&\geq \frac{1}{24}T^{2/3}\zeta^{1/3}.
\end{aligned}$$

□

14. Proof of Theorem 6.2

We denote by the t^{th} input $X_t := Z_t + \mu_t + c_t$, where $(Z_t)_{t\geq 1}$ are i.i.d., 0 mean, sub-gaussian distribution with covariance bounded above by $\sigma^2\mathbb{I}$. The deterministic sequence $(\mu_t)_{t=1}^T$ are vectors characterizing the distribution shifts and $\mathbf{c} := (c_t)_{t=1}^T$ are the corruptions. Recall that $\|\mathbf{c}\|_0 \leq \Upsilon$ by definition. Moreover, $\mathbf{c}$ is a random-variable, as it could depend on the realization of $(Z_t)_{t\geq 1}$. In the rest of the proof, denote the set of indices that the adversary modified as Λ_T (which can also be the null set), i.e., $\Lambda_T := \{t \in [T], c_t \neq 0\}$.

14.1. Notations and Definitions

Define by the *Good Event* $\mathcal{E}$ as

$$\mathcal{E} := \left\{\forall t \in \{1, \cdots, T\}, \forall j \in \{1, \cdots, t-1\}, \left\|\frac{1}{j}\sum_{s=0}^{j-1} Z_{t-s}\right\| \leq C\sqrt{\frac{d}{j}}\log\left(\frac{T^2}{\delta}\right)\right\}.$$

Definition 14.1. For every $t \in \{2, \cdots, T\}$, and $j \in \{1, \cdots, t-1\}$, denote by the vector $\mathcal{T}(j,t) \in \mathbb{R}^d$ as

$$\mathcal{T}(j,t) = \frac{1}{j}\sum_{s=0}^{j-1}(\mu_t - \mu_{t-s} - c_{t-s}).$$

For every $t \in \{2, \cdots, T\}$, denote by $B_t \leq t$, to be the index of j, at which the `While` loop in Line 4 of Algorithm 1 breaks. Denote by $\widehat{\mu}_t \in \mathbb{R}^d$ as the output of Algorithm 1 at time t. Thus, according to our notations, $\widehat{\mu}_t := \frac{1}{B_t}\sum_{s=0}^{B_t-1} X_{t-s}$. Denote by r_t as the instantaneous regret, i.e., $r_t := \|\widehat{\mu}_t - \mu_t\|$. The total regret is thus $R_T := \sum_{t=1}^{T} r_t$.

14.2. Supporting Lemmas

Proposition 14.2 (A simple concentration inequality)**.**

$$\mathbb{P}[\mathcal{E}] \geq 1 - \delta.$$

Proof. We use the classical fact that if $(Z_s)_{s=1}^T$ are i.i.d. σ sub-Gaussian random vectors, then $\frac{1}{B}\sum_{s=t-B}^{t} Z_s$ is sub-gaussian with parameter $\frac{\sigma}{B}$. Thus, we have from Definition 5.1 that, for any given $t \in \{1, \cdots, T\}$, $\left\|\frac{1}{B}\sum_{s=t-B}^{t} Z_s\right\| \leq C\sqrt{\frac{d\sigma}{B}}\log(T^2/\delta)$, holds with probability at-least $1 - \delta/T^2$. Now, applying an union bound over all t, B, of which there are at-most $\binom{T}{2} \leq T^2$ we obtain the stated result. □

Lemma 14.3. *Under the event $\mathcal{E}$, if $c_t = 0$, i.e., time t is not corrupted,*

$$r_t \leq C\sqrt{\frac{d\sigma}{|B_t|}}\log\left(\frac{T^2}{\delta}\right) + \|\mathcal{T}(|B_t|, t)\|.$$

Proof. This follows from the following chain of triangle inequalities,

$$\begin{aligned}
r_t &= \|\widehat{\mu}_t - \mu_t\|, \\
&= \left\|\frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1}(Z_{t-j} + \mu_{t-j} + c_{t-j}) - \mu_t\right\|, \\
&= \left\|\frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1} Z_{t-j} + \frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1}(\mu_{t-j} + c_{t-j} - \mu_t)\right\|, \\
&\leq \left\|\frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1} Z_{t-j}\right\| + \left\|\frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1}(\mu_t - \mu_{t-j} - c_{t-j})\right\|, \\
&= \left\|\frac{1}{|B_t|}\sum_{j=0}^{|B_t|-1} Z_{t-j}\right\| + \|\mathcal{T}(|B_t|, t)\|, \\
&\overset{(a)}{\leq} C\sqrt{\frac{d\sigma}{|B_t|}}\log\left(\frac{T^2}{\delta}\right) + \|\mathcal{T}(|B_t|, t)\|,
\end{aligned}$$

where step (a) follows as we are on event $\mathcal{E}$. □

Lemma 14.4. *Under event $\mathcal{E}$, for every $t \in \{2, \cdots, T\}$, if $B_t < t$, then,*

$$\|\mathcal{T}(B_t + 1, t)\| > C\sqrt{\frac{d\sigma}{B_t + 1}}\log\left(\frac{T^2}{\delta}\right).$$

Proof. Let t be a time index such that $B_t < t$. Denote by $j = B_t + 1$. We know that j is the index that caused the `While` loop of Algorithm 1 to break. Thus, we have

$$\left\| \frac{1}{j} \sum_{s=0}^{j-1} X_{t-s} - X_t \right\| \geq C \left(1 + \frac{2}{\sqrt{j}}\right) \sqrt{d\sigma} \log\left(\frac{T^2}{\delta}\right) \tag{2}$$

On the other hand, triangle inequality gives us the following chain

$$\begin{aligned}
\left\| \frac{1}{j} \sum_{s=0}^{j-1} X_{t-s} - X_t \right\| &= \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} - Z_t + \frac{1}{j} \sum_{s=0}^{j-1} (\mu_{t-s} + c_{t-s}) - \mu_t \right\|, \\
&\leq \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} \right\| + \|Z_t\| + \left\| \frac{1}{j} \sum_{s=0}^{j-1} \mu_{t-s} + c_{t-s} - \mu_t \right\|, \\
&= \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} \right\| + \|Z_t\| + \|\mathcal{T}(j,t)\|, \\
&\overset{(a)}{\leq} C \left(1 + \frac{1}{\sqrt{j}}\right) \sqrt{d\sigma} \log\left(\frac{T^2}{\delta}\right) + \|\mathcal{T}(j,t)\|.
\end{aligned} \tag{3}$$

Step (a) follows from triangle inequality. Thus from inequalities (2) and (3), we get that

$$\|\mathcal{T}(j,t)\| > C \sqrt{\frac{d\sigma}{j}} \log\left(\frac{T^2}{\delta}\right).$$

□

Lemma 14.5. *Under event $\mathcal{E}$, for every $t \in \{2, \cdots, T\}$, $J^*(t) - 1 \leq B_t$, where $J^*(t)$ is defined in Definition 6.1.*

Proof. The proof follows by analyzing the test-statistic in Line 4 of Algorithm 1 and arguing that for all $j \leq J^*(t) - 1$, $j \leq B_t$.

$$\begin{aligned}
\left\| \frac{1}{j} \sum_{s=0}^{j-1} X_{t-s} - X_t \right\| &= \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} - Z_t + \frac{1}{j} \sum_{s=0}^{j-1} (\mu_{t-s} + c_{t-s}) - \mu_t \right\|, \\
&\leq \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} \right\| + \|Z_t\| + \left\| \frac{1}{j} \sum_{s=0}^{j-1} \mu_{t-s} + c_{t-s} - \mu_t \right\|, \\
&= \left\| \frac{1}{j} \sum_{s=0}^{j-1} Z_{t-s} \right\| + \|Z_t\| + \|\mathcal{T}(j,t)\|, \\
&\overset{(a)}{\leq} C \left(1 + \frac{1}{\sqrt{j}}\right) \sqrt{d\sigma} \log\left(\frac{T^2}{\delta}\right) + \|\mathcal{T}(j,t)\|.
\end{aligned} \tag{4}$$

Thus, if $\|\mathcal{T}(j,t)\| \leq C\sqrt{\frac{d\sigma}{j}} \log\left(\frac{T^2}{\delta}\right)$, then $\left\| \frac{1}{j} \sum_{s=0}^{j-1} X_{t-s} - X_t \right\| \leq C \left(1 + \frac{2}{\sqrt{j}}\right) \sqrt{d\sigma} \log\left(\frac{T^2}{\delta}\right)$, i.e., the index $j \leq B_t$. □

Corollary 14.6. *Under event $\mathcal{E}$, for all $t \in \{1, \cdots, T\}$, if $j \leq B_t$, implies $\|\mathcal{T}(j,t)\| \leq C\sqrt{\frac{d\sigma}{j}} \log\left(\frac{T^2}{\delta}\right)$.*

Proof. Lemmas 14.4 and 14.5 gives that the first j such that $\|\mathcal{T}(j,T)\| > C\sqrt{\frac{d\sigma}{j}} \log\left(\frac{T^2}{\delta}\right)$, then the `While` loop in Algorithm 1 breaks. □

Concluding the proof of Theorem 6.2. With probability at-least $1 - 2\delta$, we have

$$\begin{aligned} R_T &= \sum_{t=1}^{T} r_t, \\ &\overset{(a)}{\leq} \sum_{t=1}^{T} C\sqrt{\frac{d\sigma}{|B_t|}}\log\left(\frac{T^2}{\delta}\right) + \|\mathcal{T}(|B_t|, t)\|, \\ &\overset{(b)}{\leq} \sum_{t=1}^{T} 2C\sqrt{\frac{d\sigma}{|B_t|}}\log\left(\frac{T^2}{\delta}\right), \end{aligned}$$

Step (a) follows from Lemma 14.3 and step (b) follows from Corollary 14.6. □

15. Proofs of Coroallries of Theorem 6.2

15.1. Proof of Corollary 6.4

Proof. Fix any time $t \in \{2, \cdots, T\}$ such that $J^*(t) < t$. By definition of $J^*(t)$ (Definition 6.1), we have that

$$\left\|\mu_t - \frac{1}{J^*(t)}\sum_{s=0}^{J^*(t)-1}\mu_{t-s}\right\| > C\sqrt{\frac{d\sigma}{J^*(t)}}\log\left(\frac{T^2}{\delta}\right). \tag{5}$$

On the other hand from triangle inequality, we get

$$\begin{aligned} \left\|\mu_t - \frac{1}{J^*(t)}\sum_{s=0}^{J^*(t)-1}\mu_{t-s}\right\| &= \left\|\frac{1}{J^*(t)}\sum_{s=0}^{J^*(t)-1}(\mu_t - \mu_{t-s})\right\|, \\ &\leq \frac{1}{J^*(t)}\sum_{s=0}^{J^*(t)-1}\|\mu_t - \mu_{t-s}\|, \\ &\leq \frac{1}{J^*(t)}\sum_{s=0}^{J^*(t)-1}\sum_{l=t}^{t-s-1}\|\mu_l - \mu_{l+1}\|, \\ &\leq J^*(t)\frac{\Phi}{T}. \end{aligned} \tag{6}$$

Thus, from inequalities (5) and (6), we get that

$$C\sqrt{\frac{d\sigma}{J^*(t)}}\log\left(\frac{T^2}{\delta}\right) \leq J^*(t)\frac{\Phi}{T}.$$

Rearranging the above display yields

$$\frac{1}{\sqrt{J^*(t)}} \leq \left(\frac{\Phi}{T}\right)^{1/3}\left(C\sqrt{d\sigma}\log\left(\frac{T^2}{\delta}\right)\right)^{-1/3}.$$

Plugging the last display into Theorem 6.2 gives that

$$R_T \leq \left(C\sqrt{d\sigma}T\log\left(\frac{T^2}{\delta}\right)\right)^{2/3}\Phi^{1/3}.$$

□

15.2. Proof of Corollary 6.5

Proof. Notice from the definition of $J^*(t)$ that, under the hypothesis of the proposition, $J^*(t)$ is the *distance to the nearest anomalous time instant in the past*. As there are at-most T^α number of corrupted time points, for any index $j \in \mathbb{N}$, there can be at-most $2T^\alpha$ non-corrupted time points that are at a distance of j from a corrupted point. Thus, the regret scales as

$$\begin{aligned} R_T &\leq 2C \sum_{t=1, t\in\{s:c_s=0\}}^{T} \sqrt{\frac{d\sigma}{J^*(t)-1}} \log\left(\frac{T^2}{\delta}\right), \\ &\leq 4CT^\alpha \sum_{j=1}^{T/T^\alpha} \sqrt{\frac{d\sigma}{j}} \log\left(\frac{T^2}{\delta}\right), \\ &\leq 16CT^{\frac{1+\alpha}{2}} \sqrt{d\sigma} \log\left(\frac{T^2}{\delta}\right). \end{aligned}$$

□

15.3. Doubling Trick to remove the known T assumption

Algorithm 5 `FITNESS :UNKNOWN TIME HORIZON`

1: **input**: $\sigma \geq 0$, Slack $\delta \in (0,1)$, C from Defn 5.1
2: **for** each phase $j \geq 0$ **do**
3: Run `FITNESS:SUB-GAUSSIAN` with inputs (σ, time-horizon T_j=2^j, C, slack δ_j=$\delta/2^j$) at times $2^j \cdots, 2^{j+1}-1$.
4: **end for**

We show here how Corollary 6.4 that relies on known time horizon can be relaxed. Equivalent results for Remarks 6.3 and Corollary 6.5 can be obtained by using the doubling trick.

Lemma 15.1 (Extending Corollary 6.4 to unknown T)**.** *Suppose, for all $t \in \{1, \cdots, T-1\}$, $\|\mu_t - \mu_{t+1}\| \leq \frac{\Phi}{T}$ for some $\Phi > 0$. Then with prob at-least $1-2\delta$, the doubling trick Algorithm 5 achieves regret $R_T \leq 2^{2/3}\log_2(T)\left(C\sqrt{d\sigma}T\log\left(\frac{T^2}{\delta}\right)\right)^{2/3}\Phi^{1/3}$. This is off by a factor of $(2\log(16/\delta))^{\frac{2}{3}}\log_2(T)$ from `FITNESS:SUB-GAUSSIAN`.*

Proof. As, over a time horizon of T, there are at-most $\log_2(T)$ phases, and Corollary 6.4 gives the regret in any phase j satisfies with probability at-least $1-\delta/2^j$, $R_j \leq \left(C\sqrt{d\sigma}(S_j)\log(T_j^2 2^j/\delta)\right)^{2/3}\Phi^{1/3}$, where $T_j = 2^j$ the time horizon, the regret of Algorithm 5 satisfies with probability at-least $1-\sum_{j=0}^{\log_2(T)}\delta/2^j \geq 1-2\delta$,

$$\begin{aligned} R_T &\leq \sum_{j=0}^{\log_2 T}\left(C\sqrt{d\sigma}(2^j)\log(4^j 2^j/\delta)\right)^{2/3}\Phi^{1/3}, \\ &\leq \Phi^{1/3}(2C\sqrt{d\sigma}\log(16/\delta))^{2/3}\log_2(T)T^{2/3}. \end{aligned}$$

The last step follows by using $\log(4^j 2^j/\delta) \leq j\log(16/\delta)$, for all $j \geq 0$ and $\delta < 1$ and $j \leq \log_2(T)$ in the summation. □

16. A Simple Static Sliding Window Estimator

17. Proof of Theorem 5.2

17.0.1. Notations and Definitions

We denote by $(Z_s)_{s=1}^T$ to be i.i.d. samples from $\mathcal{D}$, and by $\widetilde{X}_s := Z_s + \mu_s$, for all $s \in [T]$ as the samples sampled by nature. All expectations in the proof are with respect to the random samples $(Z_s)_{s=1}^T$. For any time point $t \in [T]$ denote by $\widetilde{\mu}_t := \frac{1}{\min(t,B)}\sum_{s=1}^t (Z_s + \mu_s)$.

Algorithm 6 Sliding Window Empirical Mean

1: **Input** Memory buffer $B \in \mathbb{N}$, $\sigma \geq 0$, Slack parameter $\delta \in (0,1)$, Time horizon T, $\lambda > 1$
2: **for** each time $t \geq 1$ **do**
3: Receive Input $X_t \in \mathbb{R}^d$
4: $\widehat{\mu}_t \leftarrow \frac{1}{\min(t-B,0)} \sum_{s=1}^{t} X_s$
5: **if** $||\widehat{\mu}_t - X_t|| \geq \lambda C\sqrt{\sigma d}\log(T/\delta)\{C$ is an absolute constant given in Definition 5.1$\}$ **then**
6: $\widehat{\mu}_t \leftarrow X_t$
7: **end if**
8: **Return** $\widehat{\mu}_t$
9: **end for**

In the rest of the proof, we assume we are on the event

$$\mathcal{E} := \left\{ \left|\left| \frac{1}{\min(t,B)} \sum_{s=\max(t-B,1)}^{t} Z_s \right|\right| \leq C\sqrt{\frac{d\sigma}{\min(t,B)}}\log(T/\delta), \forall t \in \{1,\cdots,T\} \right\}.$$

Denote the set of indices that the adversary modified as Λ_T (which can also be the null set), i.e., $\Lambda_T := \{t \in [T], X_t \neq \widetilde{X}_t\}$. In order to simplify exposition, we make some definitions.

Definition 17.1 (Clean B-history)**.** A time point $t \in [T]$ is said to have a *clean B-history* if there are no anomalies/outliers in $X_{\max(t-B,0)} \cdots, X_t$. Denote by the set $\mathcal{H}_B$ as

$$\mathcal{H}_B := \{t \in \{1,\cdots,T\} \text{ s.t. t has a clean } B \text{ history}\}.$$

Definition 17.2 (Good times)**.** A time $t \in \mathcal{H}_B$ is called *Good*, if Line 5 of Algorithm 6 is **not** executed.

17.0.2. SUPPORTING LEMMAS

Lemma 17.3. *A simple concentration inequality*

$$\mathbb{P}[\mathcal{E}] \geq 1 - \delta.$$

Proof. We use the classical fact that if $(Z_s)_{s=1}^T$ are i.i.d. σ sub-Gaussian random vectors, then $\frac{1}{B}\sum_{s=t-B}^{t} Z_s$ is sub-gaussian with parameter $\frac{\sigma}{B}$. Thus, we have from Definition 5.1 that, for any given $t \in \{1,\cdots,T\}$, $\left|\left|\frac{1}{B}\sum_{s=t-B}^{t} Z_s\right|\right| \leq C\sqrt{\frac{d\sigma}{B}}\log(T/\delta)$, holds with probability at-least $1-\delta/T$. Now, applying an union bound over all $t \in \mathcal{H}_B$, we see that with probability at-least $1-\delta$, for all $t \in \mathcal{H}_B$, $\left|\left|\frac{1}{B}\sum_{s=t-B}^{t} Z_s\right|\right| \leq C\sqrt{\frac{d\sigma}{B}}\log(T/\delta)$ holds. □

Lemma 17.4. *The cardinality* $|\{t \in \{1,\cdots,T\}$ *s.t.* $t \notin \mathcal{H}_B\}| \leq BK$.

Proof. There are K blocks of anomalies, each of which can appear in at-most B different time windows. □

Lemma 17.5. *If event $\mathcal{E}$ holds, then for all time point $t \in \mathcal{H}_B$,* $\|\mu_t - \widetilde{\mu}_t\| \leq C\frac{\sqrt{\sigma d}\log(T/\delta)}{\sqrt{B}} + \sum_{s=t-B}^{t-1}\|\mu_s - \mu_{s+1}\|$.

Proof. Fix a time $t \in \mathcal{H}_B$. The following chain holds from triangle inequalities.

$$\begin{aligned}
\|\widetilde{\mu}_t - \mu_t\| &\leq \left\| \frac{1}{B} \sum_{s=t-B}^{t} (Z_s + \mu_s) - \mu_t \right\|, \\
&= \left\| \frac{1}{B} \sum_{s=t-B}^{t} Z_s + \frac{1}{B} \sum_{s=t-B}^{t} (\mu_s - \mu_t) \right\|, \\
&\leq \left\| \frac{1}{B} \sum_{s=t-B}^{t} Z_s \right\| + \left\| \frac{1}{B} \sum_{s=t-B}^{t} (\mu_s - \mu_t) \right\|, \\
&\leq \left\| \frac{1}{B} \sum_{s=t-B}^{t} Z_s \right\| + \frac{1}{B} \sum_{s=t-B}^{t} \|\mu_s - \mu_t\|, \\
&\leq \left\| \frac{1}{B} \sum_{s=t-B}^{t} Z_s \right\| + \sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\|, \\
&\leq C\sqrt{\frac{d\sigma}{B}} \log(T/\delta) + \sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\|.
\end{aligned}$$

The last inequality follows from the fact that we are on event $\mathcal{E}$. □

Lemma 17.6. *If event $\mathcal{E}$ holds, then for all $t \in \mathcal{H}_B$, such that t is not Good,*

$$\sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\| \geq \left(\lambda - \frac{1}{\sqrt{B}} - 1\right) C\sqrt{d\sigma} \log(T/\delta).$$

Proof. A time $t \in \mathcal{H}_B$ is not good if $\|\widetilde{\mu}_t - X_t\| \geq \lambda C\sqrt{d\sigma} \log(T/\delta)$. Claim 17.5 gives that if event $\mathcal{E}$ holds, then for all $t \in \mathcal{H}_B$, $\|\mu_t - \widetilde{\mu}_t\| \leq C\frac{\sqrt{\sigma d} \log(T/\delta)}{\sqrt{B}} + \sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\|$. Thus,

$$\begin{aligned}
\lambda C\sqrt{d\sigma} \log(T/\delta) \leq \|\widetilde{\mu}_t - X_t\| &= \left\| \frac{1}{B} \sum_{s=t-B}^{t} (Z_s + \mu_s) - X_t \right\|, \\
&= \left\| \frac{1}{B} \sum_{s=t-B}^{t} (Z_s + \mu_s) - \mu_t + \mu_t X_t \right\|, \\
&\leq \left\| \frac{1}{B} \sum_{s=t-B}^{t} Z_s \right\| + \left\| \frac{1}{B} \sum_{s=t-B}^{t} (\mu_s - \mu_t) \right\| + \|Z_t\|, \\
&\leq C\frac{\sqrt{\sigma d} \log(T/\delta)}{\sqrt{B}} + \sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\| + C\sqrt{\sigma d} \log(T/\delta).
\end{aligned}$$

Rearanging the last display, we see that

$$\sum_{s=t-B}^{t-1} \|\mu_s - \mu_{s+1}\| \geq \left(\lambda - \frac{1}{\sqrt{B}} - 1\right) C\sqrt{d\sigma} \log(T/\delta).$$

□

Lemma 17.7. *If event $\mathcal{E}$ holds, then the cardinality*

$$|\{t \in \mathcal{H}_B, t \textit{ is not Good}\}| \leq \frac{B\Phi}{\left(\lambda - \frac{1}{\sqrt{B}} - 1\right) C\sqrt{d\sigma} \log(T/\delta)}.$$

Proof. From Claim 17.6, we get that if $t \in \mathcal{H}_B$, but t is not good, then $\sum_{s=\max(t-B,1)}^{t-1} \|\mu_s - \mu_{s+1}\| \geq \left(\lambda - \frac{1}{\sqrt{B}} - 1\right) C\sqrt{d\sigma}\log(T/\delta)$, whenever event $\mathcal{E}$ holds. We now recursively construct the following sequence of times. Let

$$t_0 := \inf\{t \geq 1, t \in \mathcal{H}_B, t \text{ is not Good}\}.$$

For any $k > 1$, denote by

$$t_k := \inf\{t > t_{k-1} + B : t \in \mathcal{H}_B, t \text{ is not Good}\}.$$

Observe that $k \leq \frac{\Phi}{\left(\lambda - \frac{1}{\sqrt{B}} - 1\right) C\sqrt{d\sigma}\log(T/\delta)}$. This is easy to see by contradiction. For if not, then the cumulative complexity parameter must strictly exceed Φ. Secondly from the construction, the number of points that are unaccounted for by this sequence is at-most B in between any two intervals. □

Lemma 17.8. *With probability at-least $1-\delta$, for all t such that it is not corrupted ,i.e., $c_t = 0$, $\|\mu_t - \widehat{\mu}_t\| \leq (\lambda + 1)C\sqrt{\sigma d}\log(T/\delta)$.*

Proof. From Line 6 of Algorithm 6, we know that for all $t \in \{1, \cdots, T\}$, $\|\widehat{\mu}_t - X_t\| \leq \lambda C\sqrt{\sigma d}\log(T/\delta)$ holds with probability 1. Moreover, from the sub-gaussian bounds in Definition 5.1, we get that for any fixed $t \in \{1, \cdots, T\}$, $\|X_t - \mu_t\| \leq C\sqrt{\sigma d}\log(T/\delta)$. Thus, taking an union bound over t, we get that with probability at-least $1-\delta$, for any $t \in \{1, \cdots, T\} \setminus \Lambda_T$,

$$\begin{aligned}\|\mu_t - \widehat{\mu}_t\| &\leq \|\widehat{\mu}_t - X_t\| + \|X_t - \mu_t\|, \\ &\leq \lambda C\sqrt{\sigma d}\log(T/\delta) + C\sqrt{\sigma d}\log(T/\delta).\end{aligned}$$

□

17.0.3. CONCLUDING THE PROOF OF THEOREM 5.2

Proof of Theorem 5.2. We now use these definitions to decompose the regret as follows. Under event $\mathcal{E}$,

$$\begin{aligned}R_T &= \sum_{t=1}^{T} \mathbf{1}(t \notin \Lambda_t)\|\widehat{\mu}_t - \mu_t\|, \\ &= \sum_{t=1, t\notin\Lambda_T}^{T} \|\widehat{\mu}_t - \mu_t\|, \\ &\leq \sum_{t=1, t\in\mathcal{H}_B}^{T} \|\widehat{\mu}_t - \mu_t\| + \sum_{t=1, t\notin\mathcal{H}_B}^{T} \|\widehat{\mu}_t - \mu_t\|, \\ &\overset{(a)}{\leq} \sum_{t=1, t\in\mathcal{H}_B}^{T} \|\widehat{\mu}_t - \mu_t\| + |\{t \notin \mathcal{H}_B\}|(\lambda+1)C\sqrt{\sigma d}\log(T/\delta), \\ &\overset{(b)}{\leq} \underbrace{\sum_{t=1, t\in\mathcal{H}_B}^{T} \|\widehat{\mu}_t - \mu_t\|}_{\text{Term 1}} + BK(\lambda+1)C\sqrt{\sigma d}\log(T/\delta).\end{aligned}$$

Step (a) follows from Lemma 17.8 and step (b) from Lemma 17.4. We now expand the first term as

$$\begin{aligned}
\sum_{t=1,t\in\mathcal{H}_B}^{T}\|\widehat{\mu}_t-\mu_t\| &= \sum_{t=1,t\in\mathcal{H}_B,t\text{ is Good}}^{T}\|\widehat{\mu}_t-\mu_t\| + \sum_{t=1,t\in\mathcal{H}_B,\text{ is not Good}}^{T}\|\widehat{\mu}_t-\mu_t\|,\\
&\overset{(c)}{\leq} \sum_{t=1,t\in\mathcal{H}_B,t\text{ is Good}}^{T}\|\widetilde{\mu}_t-\mu_t\| + \sum_{t=1,t\in\mathcal{H}_B,\text{ is not Good}}^{T}(\lambda+1)C\sqrt{d\sigma}\log(T/\delta),\\
&\overset{(d)}{\leq} \sum_{t=1,t\in\mathcal{H}_B,t\text{ is Good}}^{T}\|\widetilde{\mu}_t-\mu_t\| + \frac{(\lambda+1)B\Phi}{\left(\lambda-\frac{1}{\sqrt{B}}-1\right)},\\
&\leq \sum_{t=1}^{T}\|\widetilde{\mu}_t-\mu_t\| + \frac{(\lambda+1)B\Phi}{\left(\lambda-\frac{1}{\sqrt{B}}-1\right)},\\
&\overset{(e)}{\leq} C\frac{T\sqrt{\sigma d}\log(T/\delta)}{\sqrt{B}} + \sum_{t=1}^{T}\sum_{s=\max(1,t-B)}^{t-1}\|\mu_s-\mu_{s+1}\| + \frac{(\lambda+1)B\Phi}{\left(\lambda-\frac{1}{\sqrt{B}}-1\right)},\\
&\overset{(f)}{=} C\frac{T\sqrt{\sigma d}\log(T/\delta)}{\sqrt{B}} + B\Phi + \frac{(\lambda+1)B\Phi}{\left(\lambda-\frac{1}{\sqrt{B}}-1\right)}.
\end{aligned}$$

Here step (c) follows Lemma 17.8. Step (d) follows from Lemma 17.7, step (e) follows from Lemma 17.5 and step (f) follows by swapping the order of summations and the definition of the complexity parameter. □

18. Proof of Proposition 5.4

The proof follows from standard inequalities of Central Limit Theorem. At any time t, the instantaneous regret is given by

$$r_t = \left|\frac{1}{\min(B,t)}\sum_{s=\max(t-B,1)}^{t} Y_s\right|.$$

The above follows since the true mean parameter is 0. Moreover, since $(Y_s)_{s=1}^{T}$ are i.i.d., 0 mean, unit variance random variables, $\frac{1}{\min(B,t)}\sum_{s=\min(t-B,1)}^{t} Y_s$ is a 0 mean random variable with variance $\frac{1}{\min(B,t)}$. From standard results on Gaussian random variables, if $X\sim\mathcal{N}(0,\sigma)$, then $\mathbb{E}[|X|]=\sigma\sqrt{\frac{2}{\pi}}$. Thus,

$$\mathbb{E}[r_t] = \sqrt{\frac{2}{\pi\min(B,t)}}.$$

The cumulative regret is then

$$\begin{aligned}
\mathbb{E}[R_T] &= \sum_{t=1}^{T}\mathbb{E}[r_t],\\
&= \sum_{t=1}^{T}\sqrt{\frac{2}{\pi\min(B,t)}},\\
&= \sum_{t=1}^{B}\sqrt{\frac{2}{\pi t}} + \sum_{t=B+1}^{T}\sqrt{\frac{2}{\pi B}},\\
&\geq \sqrt{\frac{2}{\pi}}\left(\frac{T-B}{\sqrt{B}} + \frac{\sqrt{B}}{2}\right).
\end{aligned}$$

19. Proof of Proposition 5.5

Consider the distribution $\boldsymbol{\mathcal{D}}$ such that for all $t \leq T/2$, $X_t \sim \mathcal{N}(0,1)$, and for all $t > T/2$, $X_t \sim \mathcal{N}(4C\lambda \log(4T^2), 1)$. This sequence has one jump point and thus by construction $\Phi(\boldsymbol{\mathcal{D}}) = 4C\lambda \log(4T^2)$. Denote by the events

$$\mathcal{E}_1 = \{X_t \leq \lambda C \log(4T^2), t \in \{1, \cdots, T/2\}\},$$
$$\mathcal{E}_2 = \{X_t \geq 3\lambda C \log(4T^2), t > T/2\}.$$

Standard concentration inequalities (Lemma 14.2) gives that $\mathbb{P}[\mathcal{E}_1 \cap \mathcal{E}_2] \geq 1 - 1/2T \geq 1/2$. However, if both events $\mathcal{E}_1$ and $\mathcal{E}_2$ hold, then no sample $(X_t)_{t>T/2}$ will ever enter the dynamic window as $\|\widehat{\mu}_t - X_t\| \geq 2\lambda C \log(T^2)$, for all $t > T/2$. Furthermore, $\mu_t \leq \lambda C \log(4T^2)$, for all $t > T/2$. Thus, the regret under the event $\mathcal{E}_1 \cap \mathcal{E}_2$ is at-least $3\lambda C \frac{T}{2} \log(T^2)$, and the expected regret is at-least $3\lambda C \frac{T}{4} \log(T^2)$.

20. Potential Negative Societal Impacts

Anomaly Detection by definition is to identify 'outliers'. When used in a human context, minority groups naturally appear as outliers and thus the definition of anomalies must be more nuanced (Shekhar et al., 2021). In environments where outcomes of AD bear upon decisions made regarding humans, we recommend using the batch AD algorithm in (Shekhar et al., 2021) as the AD model class in `FITNESS` .