Differentially Private Online Active Learning with Applications to Anomaly Detection

Mohsen Ghassemi
Department of ECE
Rutgers University
Piscataway, New Jersey
m.ghassemi@rutgers.edu

Anand D. Sarwate
Department of ECE
Rutgers University
Piscataway, New Jersey
anand.sarwate@rutgers.edu

Rebecca N. Wright
Department of CS
Rutgers University
Piscataway, New Jersey
rebecca.wright@rutgers.edu

ABSTRACT

In settings where data instances are generated sequentially or in streaming fashion, online learning algorithms can learn predictors using incremental training algorithms such as stochastic gradient descent. In some security applications such as training anomaly detectors, the data streams may consist of private information or transactions and the output of the learning algorithms may reveal information about the training data. Differential privacy is a framework for quantifying the privacy risk in such settings. This paper proposes two differentially private strategies to mitigate privacy risk when training a classifier for anomaly detection in an online setting. The first is to use a randomized active learning heuristic to screen out uninformative data points in the stream. The second is to use mini-batching to improve classifier performance. Experimental results show how these two strategies can trade off privacy, label complexity, and generalization performance.

Keywords

differential privacy; online learning; active learning; anomaly detection; stochastic gradient descent

1. INTRODUCTION

Anomaly detection is an important application of statistical testing. Classical hypothesis testing approaches for detecting anomalous data samples often rely on known or partially known statistical models of anomalous vs. non-anomalous instances. However, in many real-world applications, these models may not be known and so must be learned from data before deployment or may be refined in an online manner after deployment. For example, spam filters for email may be trained on an existing corpus and then tuned based on user feedback [1]. Data-driven machine learning approaches can yield effective anomaly detectors in a wide range of applications [2].

Privacy can be a significant concern in several emerging domains for anomaly detection. For example, auditing of financial transactions, fraud investigation in medical billing, and various national security applications entail sifting through sensitive information about individuals in order to find anomalous behaviors or entities. Differential privacy was proposed a decade ago by Dwork et al. [3] to address the fundamental question of the privacy risk incurred by publishing functions of private data. Since its introduction, differentially private algorithms have been proposed for a wide range of applications by several different research communities that work with information processing systems.

In this paper, we look at two additional challenges that appear in the context of anomaly detection methods and design differentially private learning algorithms that can address these challenges. The first challenge is in the online adaptation of the anomaly detector in the face of new data. We use the private stochastic gradient framework (SGD) proposed recently [4–7] to perform stream-based online learning.

The second, more important, challenge we address is label efficiency; in anomaly detection problems where anomalies are relatively rare or when the labeling of anomalies requires an expert, the cost of acquiring a large labeled training set may be high. We use selective screening and active learning to limit the number of labels used by the algorithm. The algorithm publishes updates to its detection rule as well as the time of the updates; the key privacy challenge here is that the selection of points and timing of updates can reveal information about the training stream.

1.1 Our Contributions

We adapt general-purpose machine learning approaches to create differentially private anomaly detectors. The general strategy of our algorithm is shown in Figure 1. The input data consists of a stream of unlabeled feature vectors whose labels (anomalous or non-anomalous) are known to an oracle (an expert who can label the points). At each time, the algorithm chooses whether to request a label from the oracle. Labeled points are collected over time and processed in batches, possibly at a slower rate. When a batch of labeled points is processed, the classification rule is updated and the time and the current classification rule are published.

This abstract structure encompasses many fraud or anomaly detection systems of interest. The privacy challenge enters because the classification rule is revealed when it is updated. Both the selection rule and the update rule may reveal information about the training data. We apply standard differentially privacy techniques to control the privacy risk incurred by each of these steps.

To select points for screening, the algorithm can use ran-

AISec'16, October 28 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4573-6/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2996758.2996766

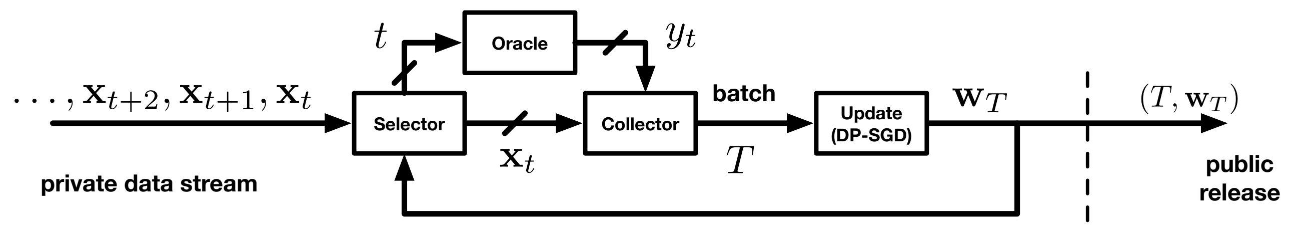


Figure 1: Block diagram of stream-based active screening and private updates. The dashed line indicates what is published by the algorithm.

domized response [8] or the exponential mechanism [9] to decide whether to select a given training sample. Our contribution here is to use a modified version of the active learning heuristic of Tong and Koller [10] within the exponential mechanism [9] to improve the usefulness of the selected points for the learning task.

With randomized selection, the classifier update step could immediately use differentially private stochastic gradient descent (SGD) [4–7]. However, due to the noise needed for differential privacy, the algorithm might potentially converge much more slowly in spite of the improved point selection process. Instead, to obtain better performance while still achieving privacy, we propose two mini-batching strategies, which we call fixed-length selection windows (FLSW) and fixed-length mini-batch (FLMB). The FLSW policy updates after fixed time intervals: the update uses a variable-sized mini-batch update of points selected within the previous window. The FLMB strategy updates after a pre-specified number of points has been selected. FLSW uses variable memory and batch size, whereas FLMB uses fixed memory and batch size.

We provide privacy proofs for these algorithms, but utility bounds are difficult to prove under the active learning strategy for point selection. An analysis of the Tong-Koller method [10] is not available even in the non-private case, and it is likely more challenging to prove such a result in the differentially private case. We demonstrate empirically that our algorithms allow the designer or other decision makers to trade off label complexity, privacy, and error during online classifier training for anomaly detection.

1.2 Related Work

Anomaly detection is a key application area for machine learning techniques. In this paper, in contrast to studying a particular application for anomaly detection, we focus on the privacy issues that may arise in learning an anomaly detector using sensitive streaming data. A 2009 survey by Chandola et al. [2] discusses several approaches for anomaly detection as well as applications. Hodge and Austin [11] survey many outlier detection techniques, some using machine learning methods. We evaluate our method on the `KDD Cup '99` dataset [12], which is a network intrusion detection problem widely studied by the machine learning community (for example, Tang and Cao [13]).

The online learning framework we use has been studied extensively in the machine learning community, starting with the work of Zinkevich [14]. Shalev-Shwartz's 2011 survey [15] gives a comprehensive introduction to the online learning problem; our algorithm is an instance of online learning in that framework. McMahan's 2014 survey [16] covers more recent work on adaptive regularizers. We do not incorporate this type of adaptivity in our work, but these ideas combined with our approach could be used to yield anomaly detection methods for scenarios in which the data distribution changes over time.

Active learning has received significant attention in the recent years. In active learning, the learning algorithm can adaptively select training examples based on previous data; this adaptivity can lead to significant improvements in sample complexity over "passive learning" approaches that use a random sample of the data [17, 18]. There is a rich body of work on active learning; we refer the reader to the survey by Settles [17], Fu et al. [19], and the monograph by Hanneke [20] for more details.

Our approach uses an active learning heuristic proposed by Tong and Koller [10], which is a simple "pool-based" active learning algorithm using a support vector machine classifier. In pool-based active learning [20], the learning algorithm selects samples to be labeled from a pool of unlabeled data. We propose a "stream-based" version of this heuristic. In stream-based active learning, in each iteration t the algorithm is given point $\mathbf{x}_t$ and has to decide whether or not to request the label y_t from an oracle. Sabato and Hess [21] recently identified conditions under which stream-based and pool-based active learning can simulate each other. A different line of work in active learning considers the case where the algorithm can choose $\mathbf{x}$ as well. Such active query algorithms [22] are related to probabilistic bisection methods [23].

Differential privacy has been widely studied since its introduction in 2006 [3]; Dwork and Roth's recent book [24] provides an inclusive review. Duchi et al. [25] study statistical learning problems under local privacy conditions measured by mutual information between private observations and non-private original data, as well as by differential privacy. Dwork et al. formalize the notion of pan privacy [26, 27] for streaming algorithms. Roughly speaking, pan-private algorithms guarantee differential privacy even when their internal memory state becomes available to an adversary. We partially expose the current state of the algorithm by publishing the learned classifier over time. However, we do assume the buffer of labeled points is kept secret and so do not guarantee pan privacy.

Online learning has also been studied with differential privacy [28]. Balcan and Feldman [29] describe a framework for designing offline active learning algorithms that are tolerant to classification noise and show that these algorithms are also differentially private. Our algorithms operate in an online manner and use a specific differentially private online active learning algorithm.

2. PRELIMINARIES

We consider a model, shown in Figure 1, in which a collection $\mathcal{D}_x = \{\mathbf{x}_t \in \mathbb{R}^d : t = 1, 2, \ldots\}$ of samples is presented to a learning algorithm in an online manner. We assume all data points are bounded, so

$$\|\mathbf{x}_t\| \leq M. \tag{1}$$

Associated with each sample point $\mathbf{x}_t$ is a (hidden) label y_t, which is known to an oracle $\mathcal{O}$. In the anomaly detection context, the vector $\mathbf{x}_t$ specifies the value of measured features about an event or individual observed at time t, and the label $y_t \in \{-1, +1\}$ indicates whether the sample is anomalous. Let $\mathcal{D} = \{(\mathbf{x}_t, y_t)\}$ be the labeled dataset. The feature vectors, but not the labels, are given to the learning algorithm. The online algorithm must decide at each time t whether to query the oracle for the label of $\mathbf{x}_t$ or discard it before observing the next sample. The goal of the learning algorithm is to learn a classifier $\mathbf{w}$ that assigns a correct label $\hat{y}$ to a feature vector $\mathbf{x}$. The labels are assigned by applying the sign function to the inner product of $\mathbf{w}$ and $\mathbf{x}$, i.e. $\hat{y} = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle)$.

The state of the algorithm is given by the current classifier $\mathbf{w}_t$. At each time $t = 1, 2, \ldots, N$, the state is $\mathbf{w}_t$ and the algorithm makes two choices. First, it selects whether or not to request the label y_t from the oracle. Second, it can choose to update the classifier/state $\mathbf{w}_t$ and publish the resulting update. We consider models in which $\mathbf{w}_t$ is updated based on new labeled data; updates are either based on a fixed schedule (if there is no new labeled data the algorithm publishes the previous state) or based on the number of new labeled points. We call the time between updates a *selection window*.

Thus the overall output of the algorithm over n iterations is the sequence of classifier estimates $\{\mathbf{w}_t : t = 1, 2, \ldots, n\}$ or alternatively the pairs of update times and updates $\{(T_i, \mathbf{w}_{T_i}) : i = 1, 2, \ldots, T\}$, where the algorithm chooses to update the classifier at times $\{T_i\}$. The algorithm requires a buffer that keeps the labeled points during each selection window until the end of the window.

Support vector machines.

We use support vector machine training to find a classifier by approximately solving the following regularized risk minimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(w) = R(\mathbf{w}) + \mathbb{E}_{\mathcal{P}}\left[\ell_h(\mathbf{w}; (\mathbf{x}, y))\right], \tag{2}$$

where $(\mathbf{x}, y)$ is a sample-label pair such that $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathcal{Y} = \{-1, 1\}$, $\mathcal{P}$ is the underlying joint distribution of the pair $(\mathbf{x}, y)$, and $R(w) = \frac{\lambda}{2}\|\mathbf{w}\|^2$ is a regularizer. The loss function ℓ_h is defined as the hinge loss $\ell_h(w; (x, y)) = [1 - y\langle \mathbf{w}, \mathbf{x} \rangle]_+$, which is convex. Throughout this paper, we use words sample, data point, and instance interchangeably to refer to $\mathbf{x}$.

For a training set of labeled points $\mathcal{S}$ (which in our case will be the subset of the data points whose points have been labeled), the SVM algorithm minimizes the regularized empirical risk as a proxy for (2):

$$\min_{\mathbf{w} \in \mathbb{R}^d} f(w) = R(\mathbf{w}) + \frac{1}{|\mathcal{S}|} \sum_{((\mathbf{x}, y) \in \mathcal{S}} \ell_h(\mathbf{w}; (\mathbf{x}, y)). \tag{3}$$

In the online setting, we assume that the true distribution $\mathcal{P}$ is unknown to the learning algorithm. Instead, the learner is presented sequentially with samples $\{\mathbf{x}_i : i = 1, 2, \ldots\}$ drawn from the marginal distribution $\mathcal{P}_X$. The regularized empirical risk minimization framework includes many classification and regression problems; we use SVM for simplicity of exposition.

Differential privacy.

Differential privacy [3] is a method of measuring the privacy risk from publishing functions of private data. In our setting the private data is the set of pairs $\{(\mathbf{x}_t, y_t) : t = 1, 2, \ldots\}$. For a finite time horizon n, consider two data streams that differ in a single point, say at time τ:

$$\mathcal{D} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_\tau, y_\tau), (\mathbf{x}_{\tau+1}, y_{\tau+1}), (\mathbf{x}_n, y_n)$$
$$\mathcal{D} = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}'_\tau, y'_\tau), (\mathbf{x}_{\tau+1}, y_{\tau+1}), (\mathbf{x}_n, y_n),$$

where all data vectors satisfy (1). We call such databases neighboring. We call a randomized algorithm $\mathcal{A}(\mathcal{D})$ operating on $\mathcal{D}$ ϵ-differentially private if the presence of any individual data point does not change the probability of $\mathcal{A}(\mathcal{D}) \in \mathcal{C}$ by much, for any set $\mathcal{C}$. Formally, we say algorithm $\mathcal{A}$ provides ϵ-differential privacy if for any two neighboring databases $\mathcal{D}$ and $\mathcal{D}$' and any set of outputs $\mathcal{C}$,

$$\left|\log \frac{\mathbb{P}(\mathcal{A}(\mathcal{D}) \in \mathcal{C}}{\mathbb{P}(\mathcal{A}(\mathcal{D}') \in \mathcal{C}}\right| \leq \epsilon, \tag{4}$$

where $\mathcal{A}(\mathcal{D})$ is the output of $\mathcal{A}$ on dataset $\mathcal{D}$.

For the algorithms described above, the output set would consist of n-tuples of classifiers $\{\mathbf{w}_t : t = 1, 2, \ldots, n\}$. Thus we are interested in controlling the total amount of privacy risk ϵ due to both the selection of points as well as the classifier update rule. In the differential privacy threat model, an adversary observes the output of the algorithm and attempts to infer whether the outcome came from input $\mathcal{D}$ or $\mathcal{D}'$; successful inference would mean that the adversary would learn whether $(\mathbf{x}_\tau, y_t)$ or $(\mathbf{x}'_\tau, y'_t)$ was in the data stream. The parameter ϵ controls how difficult this hypothesis test is, bounding the tradeoff between false alarm and missed detection (Type I and Type II) errors [30, 31].

Privacy-preserving stream-based learning.

Because our procedure reveals the classifiers over time, an adversary observes both the updates and the timing of updates. This means that potentially, the selection of points whose labels we query as well as the classifier updates must guarantee a total ϵ-differential privacy. We can consider two extremes for some intuition.

Algorithm 1 Batch SVM

```
Input: $\epsilon$, stream $\{\mathbf{x}_t\}$.
Initialize: $\mathbf{w}_1 = \mathbf{0}$, $\mathbf{S}_0 = \emptyset$.
for $t = 1$ to $n$ do
    Ask oracle $\mathcal{O}$ for label $y_t$ of $\mathbf{x}_t$.
    $\mathcal{S}_t \leftarrow \mathcal{S}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.
    Set $\mathbf{w}_{t+1} = \mathbf{w}_t$.
    Output $\mathbf{w}_t$.
end for
Output DPERM$(\mathcal{S}_n, \epsilon)$.
```

The first example in Algorithm 1 simply asks for all labels and then performs one update at time n using a batch ϵ-

differentially private SVM training method such as objective perturbation [32]; we call this DPERM in Algorithm 1. Since the algorithm selects all points, the point selection process trivially guarantees differential privacy, and the SVM training is differentially private from previous results [32]. This approach has high label complexity and high computation cost from training an SVM on the entire dataset at time n.

Algorithm 2 Private Stochastic Gradient Descent

```
Input: $\epsilon$, stream $\{\mathbf{x}_t\}$, step sizes $\{\eta_t\}$
Initialize: $\mathbf{w}_1 = \mathbf{0}$
for $t = 1$ to $n$ do
    Ask oracle $\mathcal{O}$ for label $y_t$ of $\mathbf{x}_t$.
    Update $\mathbf{w}_{t+1}$ using $\mathsf{DPSGD}(\mathbf{w}_t, \mathbf{x}_t, y_t, \eta_t, \epsilon)$.
    Output $\mathbf{w}_t$.
end for
```

On the other end of the update-frequency spectrum, we have Algorithm 2, which simply performs noisy stochastic gradient updates that guarantee differential privacy [7]. Again, this algorithm asks for the label of every data point and the updates are performed in a differentially private way, so the overall algorithm guarantees ϵ-differential privacy. This approach has low per-iteration complexity.

In settings where the learning algorithm is processing private or sensitive data, we would like to limit the privacy risk by not querying the labels of many points. In addition, it is also beneficial to limit the number of labels that must be queried in cases where training labels for anomaly detection must be generated by costly experts. In the next section, we describe how to use ideas from active learning to design algorithms that trade off label complexity, privacy, and classifier accuracy in this setting.

3. SVM PRIVATE ACTIVE LEARNING

In our system model, both the sample selection and classifier update must be made differentially private. In anomaly detection problems we are concerned with label complexity. In selecting samples for labeling, we would like to select points that are informative. We use a heuristic introduced by Tong and Koller [10] for training a support vector machine (SVM) using active learning. Their approach uses an informativeness measure based on the closeness to a hyperplane estimate $\mathbf{w}$. In our model, the informativeness of point $\mathbf{x}$ with respect to a hyperplane $\mathbf{w}$ is measured by its closeness to the hyperplane. Let

$$d(\mathbf{x}, \mathbf{w}) \triangleq \frac{|\langle \mathbf{w}, \mathbf{x} \rangle|}{\|\mathbf{w}\|} \tag{5}$$

The informativeness is

$$c(\mathbf{x}, \mathbf{w}) = \exp\left(-d(\mathbf{x}, \mathbf{w})\right) \in [0,\ 1].$$

Tong and Koller originally suggested this method for active learning in the pool-based setting: at iteration t the learner would select the point $\mathbf{x}_i$ with the largest informativeness $c(\mathbf{x}_i, \mathbf{w})$ and ask the oracle for the label. The algorithm then estimates a new $\mathbf{w}$ by retraining an SVM on the entire data set.

In our model, samples come in sequentially, so we define

$$c(t) = c(\mathbf{x}_t, \mathbf{w}_t) \tag{6}$$

at time t. In a non-private selection strategy, if the informativeness of sample $\mathbf{x}_t$ meets a certain threshold τ, then the learner queries the oracle to obtain y_i and updates the classifier. Otherwise, it discards $\mathbf{x}_t$ and waits for the next instance. Note that the condition $c(t) > \tau$ is equivalent to $d(\mathbf{x}_t, \mathbf{w}_t) < \log \frac{1}{\tau}$, which means that the sample $\mathbf{x}_t$ is within a *selection slab* of width $2 \log \frac{1}{\tau}$ around $\mathbf{w}_t$. We call this an *online active learning* strategy.

3.1 Differentially Private Point Selection

As mentioned earlier, the active learning algorithms we propose here make two decisions at each time t: first, whether to request the label y_t and add $(\mathbf{x}_t, y_t)$ to the training set, and secondly, whether to update the classifier so that $\mathbf{w}_{t+1} \neq \mathbf{w}_t$. We refer to our selection strategy as an online T-K (Tong-Koller) strategy. The original T-K heuristic's decision to query was deterministic, so to guarantee privacy we must randomize both the decision to query and the gradient step. At time t, based on the current state $\mathbf{w}_t$, the algorithm selects $\mathbf{x}_t$ to be labeled based on whether it passes the informativeness threshold. We can therefore use randomized mechanisms to select the whether or not to label the point. When the algorithm decides to update $\mathbf{w}_t$ we can use private stochastic gradient descent (SGD) [4, 7].

Bernoulli selection.

The simplest approach is to compare the informativeness $c(t)$ to a threshold and then use randomized response [8] to select the point. More formally, for parameters $p < 1/2$ and τ, the selection variable $s_t \sim \mathsf{Bern}(1-p)$ if $c(t) \geq \tau$ and $s_t \sim \mathsf{Bern}(p)$ if $c(t) < \tau$. Standard arguments imply that this provides $\epsilon_{\mathsf{Bern}} = \log \frac{1-p}{p}$ differential privacy (Lemma 1).

Exponential selection.

A strategy that is more in the spirit of the Tong-Koller method uses the exponential mechanism [9]. Consider a threshold on $d(\cdot, \cdot)$, so that we consider $d(\cdot, \cdot) \leq b$ and $d(\cdot, \cdot) > b$ as separate cases. Within the selection slab defined by b, the algorithm selects points with constant probability. Outside the slab it selects with probability that decays exponentially with the distance.

Define

$$\begin{aligned} q(t) &= \begin{cases} e^{-b\epsilon/\Delta} & d(\mathbf{x}_i, \mathbf{w}_t) \leq b \\ e^{-d(\mathbf{x}_i, \mathbf{w}_t)\epsilon/\Delta} & d(\mathbf{x}_i, \mathbf{w}_t) > b \end{cases} \\ &= \exp\left(-\max\{b, d(\mathbf{x}_i, \mathbf{w}_t)\} \cdot \epsilon/\Delta\right) \end{aligned} \tag{7}$$

where $\Delta = (1 - \frac{b}{M})M$ and $\epsilon > 0$. For this strategy, $s_t \sim \mathsf{Bern}(q(t))$. In this way, every point has a chance of being selected, so the adversary cannot infer whether an observed point is inside the selection slab or not. However, more informative points are still more likely to be selected. The privacy guarantee of this method is given in Lemma 2.

3.2 Differentially Private Update

In addition to making the point selection private, we must also make the update step private. We consider two strategies based on mini-batching: the fixed-length selection window (FLSW) policy, which updates after a fixed number of time steps, and the fixed-length mini-batch (FLMB) policy, which updates after a fixed number of selected points. Both strategies use a mini-batch update rule. Mini-batching is a well-known method in stochastic optimization for machine learning to control the variance of the subgradient estimates.

In a privacy preserving algorithm, it also provides ambiguity in the contribution of each member of the mini-batch to the approximate gradient [6]. Moreover, in our model, this technique can also provide ambiguity in sample selection.

To keep the privacy of the users whom data are present in $\mathcal{D}$, during gradient update step, one method is to follow the differentially private SGD update [7]. In this method, a controlled noise term $\mathbf{z}_t$ is added to each update. More specifically, in order to make updates ϵ_g-differentially private, we add a random noise vector $\mathbf{z}_t \in \mathbb{R}^d$ drawn i.i.d. from the following probability distribution:

$$\mathbb{P}(\mathbf{z}) = \gamma e^{-(\epsilon_g/2M)\|\mathbf{z}\|} \tag{8}$$

where M is an upper bound on every $\mathbf{x}_i \in \mathcal{D}_X$ and γ is a normalizing constant.

For a labeled point $(\mathbf{x}, y)$ let

$$u = \mathbf{1}(\ell_h(\mathbf{w}; (\mathbf{x}, y)) > 0) \tag{9}$$

be the indicator that the hinge loss is positive. For a batch $\mathcal{B} = \{(\mathbf{x}_t, y_t)\}$ of $B = |\mathcal{B}|$ points, the differentially-private mini-batch update rule is given by

$$\mathbf{w}' = \mathbf{w} - \eta\left(\lambda\mathbf{w} - \frac{1}{B}\sum_{t\in B} y_t\mathbf{x}_t u_t + \frac{1}{B}\mathbf{z}\right) \tag{10}$$

where $\mathbf{z} \sim \mathbb{P}(\mathbf{z})$ in (8) and u_t is given by (9).

We consider two variants of the mini-batch update. In the first method, the algorithm updates the classifier after every N observations, resulting in a variable batch size due to the randomized label queries. The second method suggests waiting until L samples are selected to be labeled, resulting in a fixed batch size.

Fixed-Length Selection Windows (FLSW).

Algorithm 3 FLSW update with randomized screening

```
Input: $\epsilon$, stream $\{\mathbf{x}_t\}$, step sizes $\{\eta_t\}$, batch $\mathcal{B}$.
Initialize: $\mathbf{w}_1 = \mathbf{0}$
for $t = 1$ to $n/N$ do
  Use $\epsilon_s$-DP method to decide whether to ask oracle for label $y_t$ of $\mathbf{x}_t$.
  If $y_t$ known, then add $(\mathbf{x}_t, y_t)$ to $\mathcal{B}$
  if $n = 0 \mod N$ then
    Update $\mathbf{w}_{t+1}$ using (10) with batch $\mathcal{B}$.
    Set $\mathcal{B} = \emptyset$.
    Output $\mathbf{w}_t$.
  end if
end for
```

In this update method, the learner collects labeled samples into a batch $\mathcal{B}$ during an interval of length N. It updates the classifier at a rate slower than it observes incoming samples: once every N observations. Since the adversary can only see the updates in $w(k)$, we can take advantage of the ambiguity in the number of the samples selected during the kth interval, which samples are selected, and the individual contribution of selected ones to the update. An extreme version of the FLSW rule is the *immediate update rule*, which takes $N = 1$.

Fixed-Length Mini-Batches (FLMB).

Here, before updating the classifier, the algorithm waits until a mini-batch of L labeled instances are collected. Again, the algorithm needs to decide whether to select a given sample or not for labeling immediately after it observes the sample. Thus the batch $\mathcal{B}$ has size $B = L$ in update rule (10). Unlike the fixed-length selection windows method, the number of labeled samples in a window is fixed. However, the adversary can observe the number of samples that are observed by the algorithm in each interval before L samples are selected. This, generally, adds ambiguity to the selection process since the selection result for each observation, in most cases, is not readily available to the adversary. However, when number of these observations equals L, the adversary can deduce that all points in the interval are selected.

Algorithm 4 FLMB update with randomized screening

```
Input: $\epsilon$, stream $\{\mathbf{x}_t\}$, step sizes $\{\eta_t\}$, batch $\mathcal{B}$.
Initialize: $\mathbf{w}_1 = \mathbf{0}$
for $t = 1$ to $n/N$ do
  Use $\epsilon_s$-DP method to decide whether to ask oracle for label $y_t$ of $\mathbf{x}_t$.
  If $y_t$ known, then add $(\mathbf{x}_t, y_t)$ to $\mathcal{B}$
  if $|\mathcal{B}| = L$ then
    Update $\mathbf{w}_{t+1}$ using (10) with batch $\mathcal{B}$.
    Set $\mathcal{B} = \emptyset$.
    Output $\mathbf{w}_t$.
  end if
end for
```

4. PRIVACY ANALYSIS

We now quantify the differential privacy guarantees under our models and update procedures. We compare the utility performance of the different methods empirically.

4.1 Privacy analysis for individual steps

In the proofs of the theorems in this section, let $s_t \in \{0, 1\}$ be the indicator whether the algorithm queries the label of the tth point. We begin with privacy guarantees for the selection steps and mini-batch steps that make up the algorithm.

Bernoulli selection strategy.

Lemma 1 bounds the privacy risk of the Bernoulli selection strategy.

LEMMA 1. *Consider an online active learning algorithm with the Bernoulli selection strategy as described in Section 3.1 with $p > \frac{1}{2}$. Assuming that the most recent value of $\mathbf{w}_t$ is public, this query strategy is ϵ_{ber}-differentially private, where $\epsilon_{\mathsf{ber}} = \log\left(\frac{p}{1-p}\right)$.*

PROOF. Consider two databases $\mathcal{D}_x$ and $\mathcal{D}'_x$ that differ in a single point: for some j, the jth point of $\mathcal{D}_x$ is $\mathbf{x}_p$ and the jth point of $\mathcal{D}'_x$ is $\mathbf{x}_q$. The selection procedure reveals $\mathbf{w}_t$ at each iteration t. Since the samples are presented online, until time τ the iterations have the same distribution.

Then,

$$\left|\log\frac{\mathbb{P}\left(s_t=1|\mathcal{D},\mathbf{w}_t\right)}{\mathbb{P}\left(s_t=1|\mathcal{D}',\mathbf{w}_t\right)}\right| \leq \log\frac{\mathbb{P}\left(s_t=1|i_t=p,\mathbf{w}_t\right)}{\mathbb{P}\left(s_t=1|i_t=q,\mathbf{w}_t\right)}$$
$$\leq \log\frac{\mathbb{P}\left(s_t=1|c_p(t)>\tau\right)}{\mathbb{P}\left(s_t=1|c_q(t)<\tau\right)}$$
$$\leq \log\frac{p}{1-p}. \qquad (11)$$

Similarly, the same result can be shown for the case $s_t = 0$. This concludes the proof. □

Exponential selection strategy.

Lemma 2 presents the privacy guarantee provided by the exponential mechanism.

Lemma 2. *Consider an online active learning algorithm with the exponential selection strategy as described in Section 3.1. Suppose that b in (7) satisfies $\exp(-b\epsilon_{\mathsf{exp}}/\Delta) \leq 1/2$. Assuming that the most recent value of $\mathbf{w}_t$ is public, this query strategy is ϵ_{exp}-differentially private.*

Proof. Let $\mathbf{x}_p$ and $\mathbf{x}_q$ be the jth entries of two neighboring datasets $\mathcal{D}_x$ and $\mathcal{D}'_x$, respectively. Then,

$$\left|\log\frac{\mathbb{P}\left(s_t=1|\mathcal{D},\mathbf{w}_t\right)}{\mathbb{P}\left(s_t=1|\mathcal{D}',\mathbf{w}_t\right)}\right|$$
$$\leq \left|\log\frac{\exp\left(-\max\{b,d(\mathbf{x}_p,\mathbf{w}_t)\}\epsilon_s/\Delta\right)}{\exp\left(-\max\{b,d(\mathbf{x}_q,\mathbf{w}_t)\}\epsilon_s/\Delta\right)}\right|$$
$$\leq \left|\max\{b,d(\mathbf{x}_q,\mathbf{w}_t)\}-\max\{b,d(\mathbf{x}_p,\mathbf{w}_t)\}\right|\frac{\epsilon_{\mathsf{exp}}}{\Delta}$$
$$\leq (M-b)\frac{\epsilon_{\mathsf{exp}}}{\Delta}$$
$$= \epsilon_{\mathsf{exp}}. \qquad (12)$$

Similarly, considering the assumption $\exp(-b\epsilon_{\mathsf{exp}}/\Delta) \leq 1/2$, we have

$$\left|\log\frac{\mathbb{P}\left(s_t=0|\mathcal{D},\mathbf{w}_t\right)}{\mathbb{P}\left(s_t=0|\mathcal{D}',\mathbf{w}_t\right)}\right| \leq \log\frac{1-\exp(-M\epsilon_{\mathsf{exp}}/\Delta)}{1-\exp(-b\epsilon_{\mathsf{exp}}/\Delta)}$$
$$\leq \epsilon_{\mathsf{exp}}. \qquad (13)$$

This concludes the proof. □

Mini-batch update.

Lemma 3. *The gradient step in (10) with batch $\mathcal{B}$ is ϵ_g-differentially private.*

Proof. let $\mathbf{x}_p$ is the jth entry of $\mathcal{D}_x$ and $\mathbf{x}_q$ is the jth entry of $\mathcal{D}_x$. Furthermore, assume without loss of generality that $|\langle\mathbf{w}_t,\mathbf{x}_p\rangle| \leq |\langle\mathbf{w}_t,\mathbf{x}_q\rangle|$.

$$\frac{\mathbb{P}\left(\mathbf{w}'|\mathcal{D},\mathbf{w}\right)}{\mathbb{P}\left(\mathbf{w}'|\mathcal{D}',\mathbf{w}\right)}$$
$$\leq \frac{\gamma e^{-\frac{\epsilon_g B}{2\eta M}\|\mathbf{w}'-\mathbf{w}+\eta\lambda\mathbf{w}-\frac{1}{B}\sum_{i\in\mathcal{B}:i\neq p}\eta y_i\mathbf{x}_i u_i-\frac{\eta}{B}y_p\mathbf{x}_p u_p\|}}{\gamma e^{-\frac{\epsilon_g B}{2\eta M}\|\mathbf{w}'-\mathbf{w}+\eta\lambda\mathbf{w}-\frac{1}{B}\sum_{i\in\mathcal{B}:i\neq q}\eta y_i\mathbf{x}_i u_i-\frac{\eta}{B}y_q\mathbf{x}_q u_q\|}}$$
$$\leq e^{(\epsilon_g/2M)\left\|\mathbf{x}_p-\mathbf{x}_q\right\|}$$
$$\leq e^{\epsilon_g}. \qquad (14)$$

Similarly, we can show that if $B=\emptyset$,

$$\mathbb{P}\left(\mathbf{w}'|\mathcal{D},\mathbf{w}\right)=\mathbb{P}\left(\mathbf{w}'|\mathcal{D}',\mathbf{w}\right)$$

and thus

$$\left|\log\frac{\mathbb{P}\left(\mathbf{w}'|\mathcal{D},\mathbf{w}\right)}{\mathbb{P}\left(\mathbf{w}'|\mathcal{D}',\mathbf{w}\right)}\right| = 0 \leq \epsilon_g. \qquad (15)$$

so the result is differentially private for two neighboring dataset. This concludes the proof. □

4.2 Immediate Update

As a baseline, we analyze the FLSW scheme with window size $N = 1$, which corresponds to immediate updates. Standard composition theorems [31, 33] allows us to find the overall differential privacy guarantee under each scenario under either Bernoulli or exponential sampling.

Corollary 1. *Each update in Algorithm 3 with $N = 1$ is ϵ_g-differentially private.*

Proof. The proof follows from Lemma 3 with batch size $B = 1$. □

Theorem 1. *The FLSW algorithm in Algorithm 3 with $N = 1$ is $(\epsilon_s + \epsilon_g)$-differentially private.*

Proof. Each selection step incurs ϵ_s privacy risk. If we show that each update incurs at most ϵ_g privacy risk, so the risk per sample (by composition) is at most $\epsilon_s + \epsilon_g$.

Suppose the algorithm runs for T iterations and define the matrix $W_T = [\mathbf{w}_0^\top, \mathbf{w}_1^\top, \mathbf{w}_2^\top, \ldots, \mathbf{w}_T^\top]^\top$ of revealed values. From this an adversary can infer $Q_T = [s_1, s_2, \ldots, s_T]$, the vector of selection results. Let $\mathcal{D}$ and $\mathcal{D}'$ be two neighboring databases that differ in the jth element: $\mathbf{x}_j = \mathbf{x}_p$ in $\mathcal{D}$ and $\mathbf{x}'_j = \mathbf{x}_q$ in $\mathcal{D}'$. We calculate the log-likelihood ratio:

$$\left|\log\frac{\mathbb{P}(W_T,Q_T|\mathcal{D})}{\mathbb{P}(W_T,Q_T|\mathcal{D}')}\right|$$
$$= \left|\log\frac{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t,s_t|\mathcal{D},W_{t-1})}{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t,s_t|\mathcal{D}',W_{t-1})}\right|$$
$$= \left|\log\frac{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t|\mathcal{D},W_{t-1},s_t)\mathbb{P}(s_t|\mathcal{D},W_{t-1})}{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t|\mathcal{D}',W_{t-1},s_t)\mathbb{P}(s_t|\mathcal{D}',W_{t-1})}\right|. \qquad (16)$$

Now, we use the fact that up until time j, the distribution of the two algorithms' decisions are identical:

$$\left|\log\frac{\mathbb{P}(W_T,Q_T|\mathcal{D})}{\mathbb{P}(W_T,Q_T|\mathcal{D}')}\right|$$
$$\overset{(a)}{=} \left|\log\frac{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t|\mathcal{D},\mathbf{w}_{t-1},s_t)\mathbb{P}(s_t|\mathcal{D},\mathbf{w}_{t-1})}{\prod_{t=1}^{T}\mathbb{P}(\mathbf{w}_t|\mathcal{D}',\mathbf{w}_{t-1},s_t)\mathbb{P}(s_t|\mathcal{D}',\mathbf{w}_{t-1})}\right|$$
$$\overset{(b)}{=} \left|\log\frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D},\mathbf{w}_{j-1},s_j)\mathbb{P}(s_j|\mathcal{D},\mathbf{w}_{j-1})}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}',\mathbf{w}_{j-1},s_j)\mathbb{P}(s_j|\mathcal{D}',\mathbf{w}_{j-1})}\right|$$
$$\leq \left|\log\frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D},\mathbf{w}_{j-1},s_j)}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}',\mathbf{w}_{j-1},s_j)}\right| + \left|\log\frac{\mathbb{P}(s_j|\mathcal{D},\mathbf{w}_{j-1})}{\mathbb{P}(s_j|\mathcal{D}',\mathbf{w}_{j-1})}\right|$$
$$\leq \epsilon_g + \epsilon_s, \qquad (17)$$

Equality (a) above results from the fact that given the most recent classifier $\mathbf{w}_t$, the probability distributions of $\mathbf{w}_t$ and s_t are independent of the previous classifiers. Equality (b) follows from the assumption the observed samples do not appear again in the stream, so except for the iteration where the streams are different, the conditional probabilities are identical. □

The immediate update strategy provides a baseline performance against which to compare the two mini-batch strategies in Algorithms 3 and 4.

4.3 Mini-batch Update with FLSW

Under the FLSW policy, updates happen every N observations. Since the individual point selections within this interval of length N are hidden (only the updates $\{\mathbf{w}_{kN}\}$ are revealed), this can possibly save in terms of the privacy properties.

Selection step.

Each update of the classifier in the FLSW algorithm consists of N differentially private selection steps. The only conclusion an adversary can make with regards to the selection process by observing the decision vectors $\{\mathbf{w}_{kN}\}$, is whether any samples have been selected in the kth window or not. Therefore, we consider only two events: one where no sample has been selected at all, i.e., $S_k = 0$, and the other where at least one sample is selected to be queried for its label, i.e., $S_k = 1$. First, consider the case where $\mathbf{w}_{(k+1)N+1} = \mathbf{w}_{kN+1}$, which means that $s_t = 0$ for all $kN < t \leq (k+1)N$. In this worst case scenario, there is no ambiguity as to which points have been selected during the given window. This means that the privacy guarantee when $S_k = 0$ selected is the same as when we use the immediate update.

As for when $S_k = 1$, the information that is revealed in this case is always less than when the selection results for every observation is available to the adversary, which happens in the immediate update scenario. Consequently, the privacy guarantee here is also no worse than that of the selection strategies in the immediate update method, which means that the entire selection step in the FLSW mini-batch method is ϵ_s-differentially private.

Gradient step.

The following Corollary states the privacy guarantee provided for a mini-batch gradient update.

Corollary 2. *The gradient update step in Algorithm 3 is ϵ_g-differentially private.*

Proof. The proof follows from Lemma 3 with batch size B_k. □

Note that the guarantee in this Corollary is ϵ_g by design of the gradient step. Because each step averages B_k points, the effective noise per sample is reduced by a factor of B_k. Since the gradient step is less noisy, the performance of the algorithm should improve, empirically. However, because the batch size varies in each length-N epoch, the noise level will also be variable. As we will see, this variability can affect the empirical performance of the FLSW method. For larger N the expected batch size $\mathbb{E}[B_k]$ will be larger, so longer windows can help improve performance at the expense of a (possibly) larger buffer size for the learning algorithm.

4.3.1 Overall Privacy Guarantees

Theorem 2. *The FLSW algorithm in Algorithm 3 with general N guarantees at most $(\epsilon_s + \epsilon_g)$-differential privacy.*

Proof. Suppose that we run the algorithm for K iterations each consisting of N sample observations and a gradient update. Similar to Theorem 1, define the collection of updates $W_K = (\mathbf{w}_0, \mathbf{w}_N, \mathbf{w}_{2N} \ldots, \mathbf{w}_{KN})$ and also $Q_K = (S_1, S_2, \ldots, S_K)$ where S_k indicates if any samples have been labeled during the kth interval. We have

$$\begin{aligned}
&\left|\log \frac{\mathbb{P}(W_{KN}, Q_K|\mathcal{D})}{\mathbb{P}(W_{KN}, Q_K|\mathcal{D}')}\right| \\
&= \left|\log \frac{\prod_{k=1}^{K} \mathbb{P}(\mathbf{w}_{kN}, Q_k|\mathcal{D}, W_{(k-1)N})}{\prod_{k=1}^{K} \mathbb{P}(\mathbf{w}_{kN}, Q_k|\mathcal{D}', W_{(k-1)N})}\right| \\
&\overset{(a)}{=} \left|\log \frac{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_{kN}|\mathcal{D}, \mathbf{w}_{(k-1)N}, s_t)\mathbb{P}(s_t|\mathcal{D}, \mathbf{w}_{(k-1)N})}{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_{kN}|\mathcal{D}', \mathbf{w}_{(k-1)N}, s_t)\mathbb{P}(s_t|\mathcal{D}', \mathbf{w}_{(k-1)N})}\right| \\
&\overset{(b)}{=} \left|\log \frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D}, \mathbf{w}_{j-1}, s_j)\mathbb{P}(s_j|\mathcal{D}, \mathbf{w}_{j-1})}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}', \mathbf{w}_{j-1}, s_j)\mathbb{P}(s_j|\mathcal{D}', \mathbf{w}_{j-1})}\right| \\
&\leq \left|\log \frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D}, \mathbf{w}_{j-1}, s_j)}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}', \mathbf{w}_{j-1}, s_j)}\right| + \left|\log \frac{\mathbb{P}(s_j|\mathcal{D}, \mathbf{w}_{j-1})}{\mathbb{P}(s_j|\mathcal{D}', \mathbf{w}_{j-1})}\right| \\
&\leq \epsilon_s + \epsilon_g, \qquad (18)
\end{aligned}$$

where j here indicates the window in which the jth entries of the datasets appear. Equalities (a) and (b) above are explained in Theorem 1. □

4.4 Mini-batch Update with FLMB

As mentioned earlier, the adversary can see the number of data points observed by the learner before it collects L labeled points. By randomizing the selection process and the gradient update rule, we guarantee differential privacy of the FLMB mini-batch algorithm. In this section, we discuss this guarantee.

Selection step.

We use the same query strategies that are introduced in the previous sections of this paper. However, the information revealed to an adversary is not about whether any point is selected or not, but about how many points are observed before L points are selected. In the worst case, the number of points in a given interval is also L, so the adversary could infer that all points in that interval were selected. Thus the privacy guarantee is the same as that of the selection strategy of the immediate update method when a point is selected. Any other event leaks less information than when the adversary is able to learn about the results of all individual selection decisions. Hence, the selection step in an FLMB mini-batch method is ϵ_s-differentially private.

Gradient Step.

The FLMB method guarantees a fixed batch size of L samples per iteration. Thus for the same privacy guarantee ϵ_g, it uses a factor L less noise.

Corollary 3. *The gradient update step in in Algorithm 4 is ϵ_g-differentially private.*

Proof. The proof follows from Lemma 3 with batch size L. □

4.4.1 Overall Privacy Guarantees

THEOREM 3. *The FLSW algorithm in Algorithm 4 with batch size L guarantees at most $(\epsilon_s + \epsilon_g)$-differential privacy.*

PROOF. Suppose that we run the algorithm for T iterations each consisting of T_l sample observations and a (possible) gradient update. As before, define the sets of outputs $W_T = (\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T)$ and $Q_T = (T_1, T_2, \ldots, T_T)$. We have

$$\begin{aligned}
&\left|\log \frac{\mathbb{P}(W_T, Q_T|\mathcal{D})}{\mathbb{P}(W_T, Q_T|\mathcal{D}')}\right| \\
&= \left|\log \frac{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_t, T_t|\mathcal{D}, W_{t-1})}{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_t, T_t|\mathcal{D}', W_{t-1})}\right| \\
&\overset{(a)}{=} \left|\log \frac{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_t|\mathcal{D}, \mathbf{w}_{t-1}, T_t)\mathbb{P}(T_t|\mathcal{D}, \mathbf{w}_{t-1})}{\prod_{t=1}^{T} \mathbb{P}(\mathbf{w}_t|\mathcal{D}', \mathbf{w}_{t-1}, T_t)\mathbb{P}(T_t|\mathcal{D}', \mathbf{w}_{t-1})}\right| \\
&\overset{(b)}{=} \left|\log \frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D}, \mathbf{w}_{j-1}, T_j)\mathbb{P}(T_j|\mathcal{D}, \mathbf{w}_{j-1})}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}', \mathbf{w}_{j-1}, T_j)\mathbb{P}(T_j|\mathcal{D}', \mathbf{w}_{j-1})}\right| \\
&\le \left|\log \frac{\mathbb{P}(\mathbf{w}_j|\mathcal{D}, \mathbf{w}_{j-1}, T_j)}{\mathbb{P}(\mathbf{w}_j|\mathcal{D}', \mathbf{w}_{j-1}, T_j)}\right| + \left|\log \frac{\mathbb{P}(T_j|\mathcal{D}, \mathbf{w}_{j-1})}{\mathbb{P}(T_j|\mathcal{D}', \mathbf{w}_{j-1})}\right| \\
&\le \epsilon_s + \epsilon_g. \qquad (19)
\end{aligned}$$

where j indicates the window in which the jth entries of the datasets appear. Equalities (a) and (b) above are explained in Theorem 1. □

5. NUMERICAL RESULTS

Datasets and preprocessing.

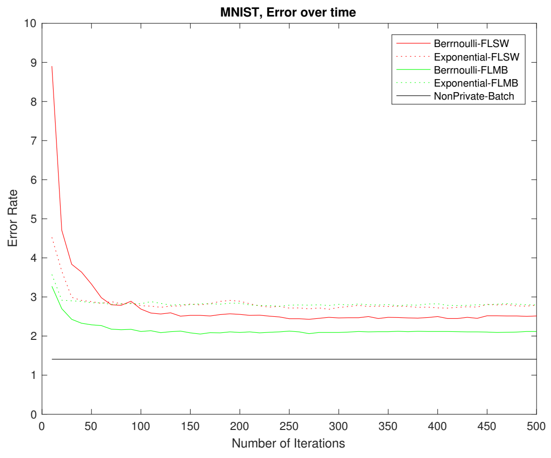
We apply our classification methods on two real-world datasets:

- The `MNIST` dataset contains images of handwritten digits [34]. The task we define here is two separate digit 0 from digit 9. The training dataset consists of around 12000 images of handwritten digits 0 and 9 and the test set contains less than 200 instances.
- The `KDD Cup 1999` dataset [12] is about network intrusion detection and more closely matches our anomaly detection motivation. This dataset consists of around half a million examples. The task here is to build an intrusion detection model that can differentiate "normal" network connections from "intrusions" or "attacks".

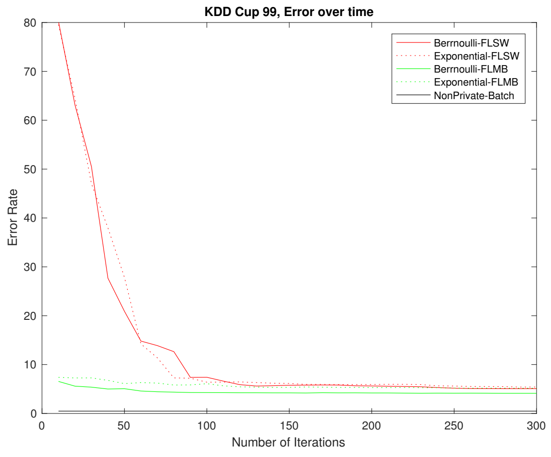
For the `MNIST` dataset, we apply PCA to reduce the dimentionality from the original 784 to 50. For the `KDD Cup 1999` dataset we remove to irrelevant features (`numoutbound_cmds` and `is_host_login`) and convert 7 remaining categorical features into binary vectors. This process leaves us with 120 features. We then project all entries of both data sets onto the unit ball.

Procedure.

In all experiments we set differential privacy guarantees $\epsilon_g = 1$ and $\epsilon_s = 1$. We set $p = e/(1+e)$ to make the Bernoulli selection processes also 1-differentially private. Parameter b in (7) is set to 0.2. We used step sizes $\eta_t = \eta/t$ and found values of η and the regularization parameter λ using



(a) Error versus number of iterations for `MNIST`



(b) Error versus number of iterations for `KDD Cup 1999`

Figure 2: Classifier performance as a function of number of labels requested.

cross-validation. Unless stated otherwise, the threshold τ in Bernoulli strategies is set to $e^{-0.2}$.

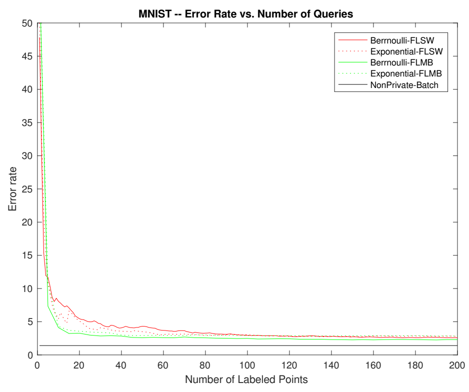
At every classifier update iteration we add noise according to (8), and the iterate is projected onto the uniform ball.

In all experiments, we averaged the results obtained over 10 random permutations of training data streams. The error bars in Figures 4 are 1-standard deviation error bars.

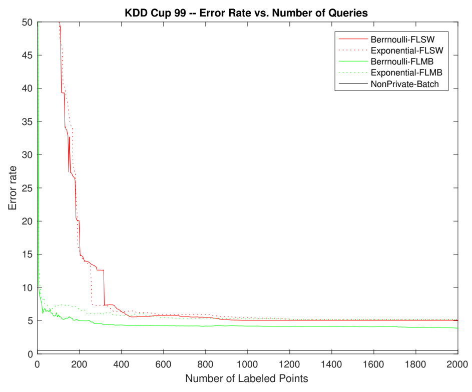
Error rate over time.

Figures 2a and 2b show the misclassification error rates of FLSW and FLMB update methods for each dataset. For both FLSW and FLMB, we set the selection window size to 5. We observe that even though all of the methods show fast convergence to their limit value, due to the added gradient noise we see a small gap between the non-private batch error rate and the limit error rates of our proposed methods. After processing many samples, progress slows due to the small step size in the SGD iterations. This shows the trade-off between privacy and accuracy in our algorithms.

A comparison between different methods based on these plots would not be fair as the number of labeled points that



(a) Error versus labels requested for `MNIST`



(b) Error versus labels requested for `KDD Cup 1999`

Figure 3: Classifier performance as a function of number of labels requested.

the algorithms use differs from one another. Therefore, In the next section, we compare the error rates against number of labeled points. However, from these experiments, we can see that in general the FLMB performance tends to be better than the FLSW performance. We conjecture that the fixed batch size controls the variance of the subgradient steps in the SGD iterations.

Error rate versus label cost.

Figures 3a and 3b show the misclassification error rates as a function of label costs, assuming that each label costs 1 to acquire. Both results on `MNIST` and `KDD Cup 99` show that the FLMB method is more efficient than FLSW in the sense that it achieves better error rates for a given label budget. We believe this is because for our choice of parameters, the number of labeled points per iteration, denoted by B in (10), in FLSW is smaller than or equal that of FLMB. This means, according to update rule (10), larger mini-batches are selected and smaller noise is added to the iterate during an FLMB update.

We note that in our FLSW experiment in this section, the values on the label complexity axis (x-axis) are not equally spaced and the spaces also differ across experiments over different permutations. In order to address this issue, we use piece-wise linear interpolation and sample the interpolated iterate values at the desired equally spaced points.

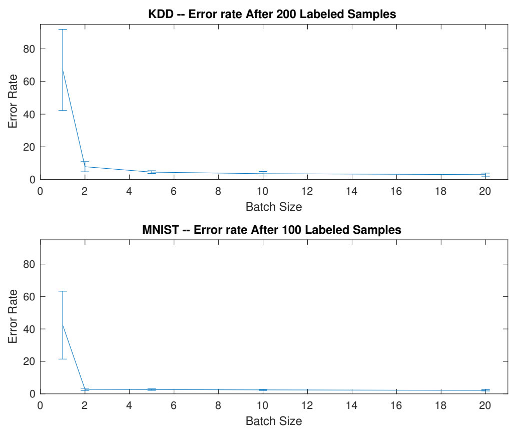


Figure 4: Classifier performance as a function of batch size for FLMB and FLSW with Bernoulli strategy

Selection method.

Surprisingly, we see from the experiments in Figures 2 and 3 that the difference between the Bernoulli and Exponential selection methods is negligible in terms of overall performance. We conjecture two reasons for this. First, by choosing a large value of ϵ_s for the selection step, the difference between the two sampling distributions is not too significant. Second, the active learning heuristic provides the most improvements in the early stages of the classifier training process: once the classifier has stabilized, additional training samples do not improve the performance significantly.

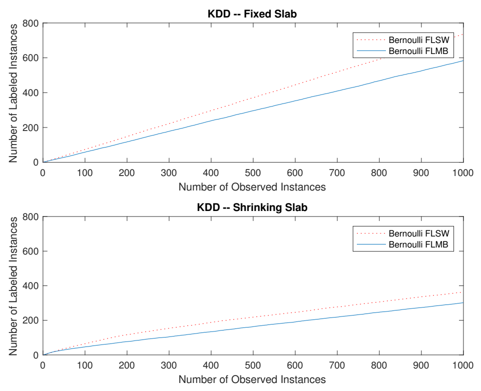
Mini-Batching.

Figure 4 shows the error rate of the FLMB update method with Bernoulli selection strategy for batch sizes $\{1, 2, 5, 10, 20\}$ over both datasets. For the `MNIST` dataset the error rate is reported after using 100 labeled data points, while for the `KDD Cup 99` the results are from 200 labeled examples. We observe that mini-batching, as expected, reduces the variance in results.

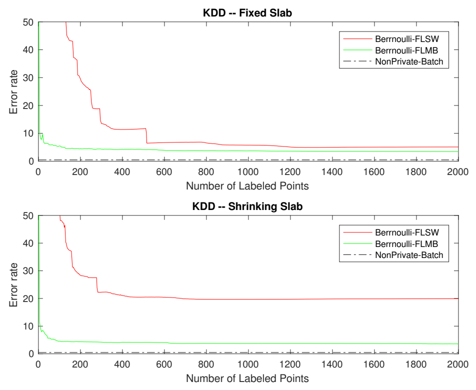
Time-varying selection strategies.

Label costs are not necessarily constant over time. It can be the case that the more the learner queries for labels, the more the oracle charges per label. Moreover, as the classifier is updated using more labeled points, it becomes more accurate and points far from the classifier are more likely to be noisy. Therefore, we would like to lower the chances of labeling uninformative, noisy data points by making the selection policy stricter.

As an example, we study the performance of both FLSW and FLMB methods with time-varying Bernoulli selection strategy over the `KDD Cup 99` dataset. We compare using a fixed threshold $\tau = e^{-0.2}$ versus a time-varying threshold $\tau = e^{-\frac{1}{t}}$. The latter case uses a selection slab that shrinks linearly over time. We remind the reader that t is the counter for classifier updates and not the data stream counter. The results of this experiment are given in Figures 5a and 5b. Figure 5a shows a considerable reduction in the label complexity in the time-varying query method. In fact, we observe that the selection rate is not linear anymore and becomes sublinear.



(a) Number of labels requested vs number of iterations for fixed and time-varying Bernoulli strategy in FLMB for `KDD Cup 1999`



(b) Error versus labels for fixed and shrinking slabs on `KDD Cup 1999`

Figure 5: Comparing fixed and shrinking slabs for different strategies in terms of labels complexity and error performance.

Interestingly, the generalization performance of the FLMB method, for given number of labeled points, is not compromised. This is not surprising since each update uses a fixed mini-batch size and the learner is simply more selective about informativeness and needs to wait for a longer time to collect a batch. The results show that this selection strategy is actually querying more informative points and does not lose much by discarding more points. We expect that for some datasets this strategy actually could improve the performance by using its labeling budget more judiciously.

The story for the FLSW method is different. The classification performance of this algorithm is considerably compromised. We suspect that with a shrinking selection slab the bar for selection is too high; so few instances can pass the informativeness threshold that this method (in later iterations) effectively loses the "mini-batch" property.

6. CONCLUSION

We propose two randomized query strategies for privacy-preserving selection of informative instances in a stream, and explore two mini-batching techniques that improve the classifier performance of the Tong-Koller-based active learning heuristic in our context. We provide theoretical analysis to quantify the differential privacy guarantees of these methods and experimental results showing classification performance of our methods. Our experiments show that the proposed algorithms have acceptable generalization performance while preserving differential privacy of the users whose data is given as input to the algorithms.

Our experiments also show that mini-batching reduces the variance in the results and improves the convergence of our algorithms. Although the differential privacy guarantees for our mini-batch methods are the same as the immediate update method, the worst case scenarios are less likely to happen as the selection window size gets larger. We also demonstrate empirically that a time-varying selection strategy where the informativeness criterion becomes stricter with time can substantially reduce label costs while showing acceptable generalization performance, as long as the mini-batching structure is not compromised.

Our procedure used a fixed cost function over time; designing cost-aware algorithms for settings where label cost increases with the number of queries seems challenging. Investigating tradeoffs with respect to ϵ and sample size could shed light on how stronger privacy guarantees affect the speed of learning and label complexity. Identifying how privacy risk and label complexity interact could yield algorithms which can switch between more and less aggressive active learning techniques.

Finally, because this work is motivated by problems in anomaly detection, evaluating and adapting the private online active learning framework here to domain-specific problems could give more guidance on how to set ϵ in practical scenarios. Implementing practical differentially private machine learning algorithms (as opposed to statistical aggregates/estimators) can tell us when reasonable privacy guarantees are or are not possible. Another related problem in this vein is online learning of feature maps in a differentially private manner. This could help characterize statistical properties of the anomalous class in an online and sample-efficient manner.

Acknowledgment

This material is based upon work supported by the U.S. Department of Homeland Security under Award Number 2009-ST-061-CCI002, by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under contract No. N66001-15-C-4070, and by the National Science Foundation under award 1453432.

References

[1] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. MIT Press, Cambridge, MA, 2009. ISBN 0262170051, 9780262170055.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15:1–15:58, 2009. URL http://doi.acm.org/10.1145/1541880.1541882.

[3] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, 2006. URL http://dx.doi.org/10.1007/11681878_14.

[4] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 464–473, 2014. URL http://dx.doi.org/10.1109/FOCS.2014.56.

[5] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 429–438, 2013. URL http://dx.doi.org/10.1109/FOCS.2013.53.

[6] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 Global Conference on Signal and Information Processing (GlobalSIP 2013)*, pages 245–248, 2013. URL http://dx.doi.org/10.1109/GlobalSIP.2013.6736861.

[7] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. Learning from data with heterogeneous noise using sgd. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 894–902, 2015. URL http://jmlr.org/proceedings/papers/v38/song15.html.

[8] Stanley L. Warner. Randomized response: a survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. URL http://dx.doi.org/10.1080/01621459.1965.10480775.

[9] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, October 2007. URL http://dx.doi.org/10.1109/FOCS.2007.41.

[10] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, November 2001. URL http://www.jmlr.org/papers/v2/tong01a.html.

[11] Victoria J. Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004. URL http://dx.doi.org/10.1007/s10462-004-4304-y.

[12] Saharon Rosset and Aron Inger. KDD-cup 99: knowledge discovery in a charitable organization's donor database. *ACM SIGKDD Explorations Newsletter*, 1(2):85–90, 2000. . URL http://doi.acm.org/10.1145/846183.846204.

[13] Hua Tang and Zhuolin Cao. Machine learning-based intrusion detection algorithms. *Journal of Computational Information Systems*, 5(6):1825–1831, 2009.

[14] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, pages 928–936, 2003. URL http://www.aaai.org/Papers/ICML/2003/ICML03-120.pdf.

[15] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011. URL http://dx.doi.org/10.1561/2200000018.

[16] H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014. URL http://arxiv.org/abs/1403.3465.

[17] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648. 2010. URL http://burrsettles.com/pub/settles.activelearning.pdf.

[18] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 65–72, 2006. URL http://doi.acm.org/10.1145/1143844.1143853.

[19] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283, 2013. URL http://dx.doi.org/10.1007/s10115-012-0507-8.

[20] Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011. URL http://dx.doi.org/10.1214/10-AOS843.

[21] Sivan Sabato and Tom Hess. Interactive algorithms: from pool to stream. In *Proceedings of the 2016 Conference On Learning Theory (COLT 2016)*, pages 1419–1439, 2016. URL http://www.jmlr.org/proceedings/papers/v49/sabato16.

[22] Rui M. Castro and Robert D. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008. URL http://dx.doi.org/10.1109/TIT.2008.920189.

[23] Michael Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963. URL http://dx.doi.org/10.1109/TIT.1963.1057832.

[24] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4): 211–407, 2014. URL http://dx.doi.org/10.1561/0400000042.

[25] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Privacy aware learning. *Journal of the ACM (JACM)*, 61(6):38:1–38:57, 2014. URL http://doi.acm.org/10.1145/2666468.

[26] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010. URL http://doi.acm.org/10.1145/1806689.1806787.

[27] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In *Proceedings of The First Symposium on Innovations in Computer Science (ICS 2010)*, 2010. URL http://www.wisdom.weizmann.ac.il/mathusers/naor/PAPERS/pan_private.pdf.

[28] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT 2012)*, volume 23 of *JMLR Workshop and Conference Proceedings*, pages 24.1–24.34, 2012. URL http://www.jmlr.org/proceedings/papers/v23/jain12/jain12.

[29] Maria-Florina Balcan and Vitaly Feldman. Statistical active learning algorithms. In *Advances in Neural Information Processing Systems*, pages 1295–1303, 2013. URL http://papers.nips.cc/paper/5101-statistical-active-learning-algorithms.pdf.

[30] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010. URL http://dx.doi.org/10.1198/jasa.2009.tm08651.

[31] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1376–1385, 2015. URL http://www.jmlr.org/proceedings/papers/v37/kairouz15.

[32] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011. URL http://www.jmlr.org/papers/v12/chaudhuri11a.

[33] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Communications of the ACM*, 53(9):89–97, 2010. URL http://doi.acm.org/10.1145/1810891.1810916.

[34] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998. URL http://dx.doi.org/10.1109/5.726791.