Cost-Sensitive Online Active Learning with Application to Malicious URL Detection

Peilin Zhao
School of Computer Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
peilinzhao@ntu.edu.sg

Steven C.H. Hoi
School of Computer Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
chhoi@ntu.edu.sg

ABSTRACT

Malicious Uniform Resource Locator (URL) detection is an important problem in web search and mining, which plays a critical role in internet security. In literature, many existing studies have attempted to formulate the problem as a regular supervised binary classification task, which typically aims to optimize the prediction accuracy. However, in a real-world malicious URL detection task, the ratio between the number of malicious URLs and legitimate URLs is highly imbalanced, making it very inappropriate for simply optimizing the prediction accuracy. Besides, another key limitation of the existing work is to assume a large amount of training data is available, which is impractical as the human labeling cost could be potentially quite expensive. To solve these issues, in this paper, we present a novel framework of Cost-Sensitive Online Active Learning (CSOAL), which only queries a small fraction of training data for labeling and directly optimizes two cost-sensitive measures to address the class-imbalance issue. In particular, we propose two CSOAL algorithms and analyze their theoretical performance in terms of cost-sensitive bounds. We conduct an extensive set of experiments to examine the empirical performance of the proposed algorithms for a large-scale challenging malicious URL detection task, in which the encouraging results showed that the proposed technique by querying an extremely small-sized labeled data (about 0.5% out of 1-million instances) can achieve better or highly comparable classification performance in comparison to the state-of-the-art cost-insensitive and cost-sensitive online classification algorithms using a huge amount of labeled data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Theory, Algorithms, and Experimentation

Keywords

Malicious URL Detection, Cost-Sensitive Learning, Online Learning, Active Learning

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

1. INTRODUCTION

The World Wide Web (WWW) allows people to access massive information on the internet, but also brings malicious information, such as fake drug, malware, and so on. A user accesses all kinds of information (benign or malicious) on the WWW by clicking on a URL (Uniform Resource Locator) that links to a particular website. It is thus very important for internet users to evaluate the risk of clicking a URL in order to avoid accessing the malicious web sites. This is however very challenging for individual internet users. To tackle this challenge, researchers have attempted to investigate techniques to automatically classify whether a URL is malicious or not over the past few years, which is formally known as "malicious URL detection" [20, 25, 26, 27].

In literature, a variety of techniques have been proposed to solve the malicious URL detection problem [20, 25, 26, 27]. One major category of techniques formulates the URL detection as a classical supervised classification task and attempts to train a binary classification model in an offline learning fashion to distinguish between malicious and normal URLs [20, 25]. These techniques usually require to collect a considerable amount of training data in order to build a good classification model. In contrast, another category of techniques formulates it as an online supervised learning task [27], which is more suitable for large-scale problems. However, all these algorithms try to maximize the classification accuracy of the learnt model by assuming the ratio between the malicious and benign ULRs is balanced explicitly or implicitly.

Although malicious URL detection has been well studied for years, it remains a very challenging research problem today, which is primarily due to several reasons. First of all, it is often a highly class-imbalanced learning problem as the number of malicious is significantly smaller than that of normal ones, which brings a critical challenge to many schemes using regular classification techniques. Second, it is usually very expensive to collect labeled data, especially the positive training data ("malicious"), which limits the application of some classical supervised classification approaches. Moreover, in a real-world application, data usually arrives in a sequential/online fashion and the size of data patterns can be very large, leading to a big challenge for developing efficient and scalable algorithms for malicious detection.

To address the above challenges of malicious URL detection, in this paper, we present a novel framework of Cost-Sensitive Online Active Learning (CSOAL) which can tackle malicious detection in a fairly natural, effective, and scal-

able approach. Unlike many existing batch learning approaches, the key idea of our framework is to formulate malicious URL detection as an online active learning task which aims to maximize the detection performance by actively querying a small amount of informative labeled data via a cost-sensitive online learning setting. In particular, we propose two CSOAL algorithms by optimizing two different cost-sensitive measures (i.e., the weighted sum of sensitivity and specificity and the weighted cost), and theoretically analyze the performance bounds of the proposed algorithms. We further validate the empirical performance of the proposed algorithms through an extensive set of experiments for a large-scale online malicious URL detection task.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed framework and algorithms, followed by their theoretical analysis in Section 4. Section 5 discusses our experimental results and Section 6 concludes our work.

2. RELATED WORK

Our work is closely related to two topics in web mining and machine learning: malicious URL detection and online learning. Although both have been well studied separately, to the best of our knowledge, this is the first work to tackle the malicious URL detection task using online active learning. Below briefly reviews important work in both areas.

2.1 Online Learning

Online learning represents a family of efficient and scalable machine learning algorithms [29, 11, 5, 9, 39, 33, 19]. Unlike conventional batch learning methods that assume all training instances are available prior to the learning task, online learning repeatedly updates the predictive models sequentially, which is more appropriate for web applications where training data often arrive sequentially.

In literature, a variety of online learning methods have been proposed in machine learning [30]. One very well-known method is the Perceptron algorithm [29, 13], which updates the model by adding a new example with some constant weight into the current set of support vectors when the example is misclassified. Recently a lot of new online learning algorithms have been developed based on the criterion of maximum margin [11, 15, 21, 9, 23]. One notable technique is the Passive-Aggressive (PA) method [9], which updates the classification function when a new example is misclassified or its classification score does not exceed some predefined margin. In this work, we apply the PA algorithm to solve the online learning task. Different from the regular PA learning setting which assumes class label of every online incoming instance will be revealed, our approach queries the class labels of only a limited amount of online incoming instances through active learning.

In addition to regular online learning techniques, our work is also closely related to another online learning topic in machine learning, that is, selective sampling [14, 4] or label-efficient learning [16, 7], which also queries class labels of a subset of online received instances by developing appropriate sampling strategies. However, conventional label-efficient learning approaches often aim to optimize the mistake rate (or equivalently the classification accuracy), which is clearly inappropriate for malicious URL detection tasks. In contrast, our approach addresses the challenge of online malicious URL detection by attempting to optimize cost-sensitive metrics (either weighted sum of sensitivity and specificity or weighted cost) [32].

Finally, our work generally belongs to the category of "online" active learning, which differs from a large family of "batch" active learning studies in literature [31, 17, 18].

2.2 Malicious URL Detection

Malicious URL detection is about how to detect malicious URLs automatically or semi-automatically, which has been extensively studied in web and data mining communities for years [20, 28, 36]. In general, we can divide the existing work into two categories: (i) non-machine learning methods, such as blacklisting [37] or rule-based approaches [38, 35]; and (ii) machine learning methods. The non-machine learning approaches generally suffer from poor generalization to new malicious URLs and unseen malicious patterns. In the following, we will focus on reviewing important related work using machine learning methods.

In literature, a variety of machine learning schemes have been proposed for malicious URL detection, which can be grouped into two categories: (i) regular batch machine learning methods [25, 8, 34], and (ii) online learning methods [26]. Most of the existing malicious URL detection methods employ regular batch classification techniques to learn a classification model (classifier) from a training data set of labeled instances [8], and then applies the model to classify a test/unseen instance. In general, the classification problem can be formulated as either binary classification (normal vs. abnormal) [25] or multi-class classification (assuming normal patterns come from multiple classes). In literature, a variety of classification techniques have been applied, such as Support Vector Machines (SVM) [25, 8], Logistic Regression [25], maximum entropy [20], Naive Bayes [3, 25], and so on. However, these algorithms typically require to collect and store all the training instances in advance and build the models in a batch learning fashion, which is both time and memory inefficient and suffers from very expensive retraining cost whenever any new training data arrives.

Unlike the batch machine learning algorithms, online Learning [26] has been recently proposed as a scalable approach to tackling large-scale online malicious URL detection tasks. In general, online learning methods are more suitable for large-scale, real-world online web applications due to their high efficiency and scalability. However, most of the previous online learning algorithms were designed to optimize the classification accuracy, typically by assuming the underlying training data distribution is class-balanced explicitly or implicitly. This is clearly inappropriate for online malicious URL detection tasks since the real-world URL data distribution is often highly class-imbalanced, i.e, the number of malicious URLs is usually significantly smaller than the number of benign URLs on the WWW. Therefore, it is very important to take this issue into consideration when designing a machine learning and data mining algorithm for solving a practical URL detection task.

Finally, all the existing learning approaches usually have to label a fairly large amount of training instances in order to build a sufficiently good classification model, which is impractical as the labeling cost is often expensive in a real-word application. This thus motivates us to study a unified learning scheme, which not only is able to minimize the labeling cost, but also maximize the predictive performance with the given amount of labeled training instances.

3. FRAMEWORK OF COST-SENSITIVE ONLINE ACTIVE LEARNING

3.1 Overview

Consider a real-world online malicious URL detection problem where data instances arrive sequentially. The goal of supervised malicious URL detection is to construct a predictive model that can accurately predict if an incoming URL instance is malicious or not. In general, this can be formulated as a binary classification task where malicious URL instances are from positive class ("+1") and normal URL instances are from negative ("-1"). For an online malicious URL detection task, the goal is to develop an online learner to incrementally build a classification model from a sequence of URL training data instances via an online learning fashion. In particular, for each learning round, the learner first receives a new incoming URL instance for detection; it then applies the classification model to predict if it is malicious or not; at the end of the learning round, if the truth class label of the instance can be revealed from the environment, the learner will make use of the labeled instance to update the classification model whenever the classification is incorrect (or the prediction loss is nonzero).

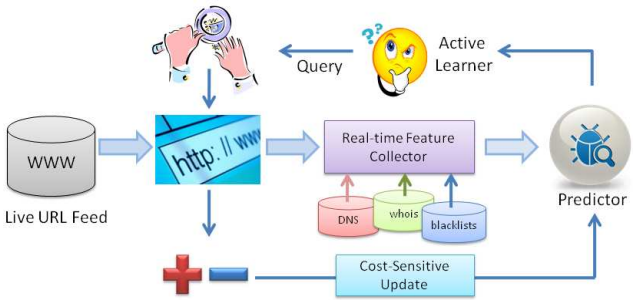

Figure 1: Framework of the proposed CSOAL system for malicious URL detection

In general, it is natural to apply online learning to solve online malicious URL detection. However, it is impractical to directly apply an existing online learning technique to solve the problem. This is because a conventional online classification task usually assumes the class label of every incoming instance will be disclosed so as to be used to update the classification model at the end of every learning round. Clearly it is impossible or highly expensive if the learner queries the class label of every incoming instance in an online malicious URL detection task. To address this challenge, we propose to investigate a novel framework of Cost-Sensitive Online Active Learning (CSOAL), as shown in Figure 1. In general, the proposed CSOAL framework attempts to address two key challenges in a systematic and synergic learning approach: (i) the learner must decide when it should query the class label of an incoming URL instance; and (ii) how to update the classifier in the most effective way where there is a new labeled URL instance. The basic idea of our unified learning approach is to explore active learning strategy to address the first issue, and to investigate cost-sensitive online learning strategy to address the second issue. Before presenting our detailed technique, we first give a formal formulation of the online malicious URL detection problem in the following.

3.2 Problem Formulation

Let us denote by $\mathbf{x}_t \in \mathbb{R}^d$ the feature vector of a URL instance received at the t-th learning round, and $\mathbf{w}_t \in \mathbb{R}^d$ a linear prediction model learned from the previous $t-1$ training examples. We also denote the prediction of the t-th instance as $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$. The value $|\mathbf{w}_t \cdot \mathbf{x}_t|$ is known as "margin", which can be used as the confidence of the learner on the prediction. The true label for instance $\mathbf{x}_t$ is denoted as $y_t \in \{-1, +1\}$. If $\hat{y}_t \neq y_t$, the learner made a mistake.

We consider a sequence of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ for online malicious URL detection, where class label y_t can be revealed after online prediction (depending on if it is queried). To solve such a task, traditional online learning would try to maximize the online accuracy (or minimize the online mistake rates equivalently). However, this is inappropriate for malicious URL detection problem because a trivial learner that simply classifies any example as negative could achieve a quite high accuracy for a dataset with highly rare malicious URLs. Thus, we propose to study new online learning algorithms, which can optimize a more appropriate performance metric, such as the *sum* of weighted *sensitivity* and *specificity*, i.e.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity \tag{1}$$

where $0 \leq \eta_p, \eta_n \leq 1$ and $\eta_p + \eta_n = 1$. When $\eta_p = \eta_n = 1/2$, sum is the well-known balanced accuracy, which is adopted as a metric in the existing studies for anomaly detection [22]. In general, the higher the *sum* value, the better the performance. Besides, another suitable metric is the total cost suffered by the algorithm, which is defined as:

$$cost = c_p \times M_p + c_n \times M_n \tag{2}$$

where M_p and M_n are the number of false negatives and false positives respectively, $0 \leq c_p, c_n \leq 1$ are the cost parameters for positive and negative classes, respectively, and we assume $c_p + c_n = 1$. The lower the *cost* value, the better the classification performance.

3.3 CSOAL Algorithms

We now propose an online learning framework for online malicious URL detection by optimizing the previous two cost-sensitive measures. Before presenting our algorithms, we prove an important proposition below to motivate our solution. For simplicity, we assume $\|\mathbf{x}_t\| = 1$ for the rest.

PROPOSITION 1. *Consider a cost-sensitive classification problem, the goal of maximizing the weighted sum in (1) or minimizing the weighted cost in (2) is equivalent to minimizing the following objective:*

$$\sum_{y_t=+1} \rho \mathbb{I}_{(y_t \mathbf{w} \cdot \mathbf{x}_t < 0)} + \sum_{y_t=-1} \mathbb{I}_{(y_t \mathbf{w} \cdot \mathbf{x}_t < 0)} \tag{3}$$

where $\rho = \frac{\eta_p T_n}{\eta_n T_p}$ for the maximization of the weighted sum, T_p and T_n are the number of positive examples and negative examples respectively, $\rho = \frac{c_p}{c_n}$ for the minimization of the weighted misclassification cost, and $\mathbb{I}_\pi$ is the indicator function that outputs 1 if the statement π holds and 0 otherwise.

The proof is omitted due to space limitation. Proposition 1 gives the explicit objective function for optimization, but the indicator function is non-convex. To tackle this issue, we replace the indicator function by its convex surrogate,

i.e., a modified hinge loss function:

$$\ell(\mathbf{w}; (\mathbf{x}, y)) = \max(0, \rho * \mathbb{I}_{(y=1)} + \mathbb{I}_{(y=-1)} - y(\mathbf{w} \cdot \mathbf{x})) \quad (4)$$

As a result, we can formulate the primal objective function for online malicious URL detection as follows:

$$\mathcal{F}_p^b(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T} \ell_t(\mathbf{w}) \quad (5)$$

where the regularization parameter $C > 0$, the loss function $\ell_t(\mathbf{w}) = \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = \max(0, \rho_t - y(\mathbf{w} \cdot \mathbf{x}))$ and $\rho_t = \rho * \mathbb{I}_{(y_t=1)} + \mathbb{I}_{(y_t=-1)}$. The idea of this formulation is somewhat similar to the biased formulation of batch SVM for learning with imbalanced datasets [1].

To online optimize the above objective (5), following the passive aggressive learning method [9], we have a similar online optimization objective:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2 + C\ell_t(\mathbf{w})$$

which enjoys the following closed-form solution:

$$\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \tau_t y_t \mathbf{x}_t, \quad \text{where } \tau_t = \min(C, \ell_t(\mathbf{w}_t)). \quad (6)$$

Based on the above derived updating method, we would develop an online active learning algorithm for malicious URL detection. However, unlike regular online learning [27], the key challenges to an online active scheme for malicious URL detection are two-fold: (i) a learner should decide when to query the class label of an incoming instance, and (ii) once the class label is queried and disclosed, how to exploit the labeled instance to update the learner in an effective way. To tackle these challenges, we propose a framework of Cost-Sensitive Online Active Learning (CSOAL), which adopts a simple yet effective active learning scheme to decide whether an incoming instance should be queried, and employ the above proposed cost-sensitive updating method (6) to exploit the labeled instance for updating the online learner.

Specifically, at the t-th round, the CSOAL algorithm decides if the class label should be queried according to a Bernoulli random variable $Z_t \in \{0, 1\}$ with probability

$$q_t = \delta/(\delta + |p_t|), \quad (7)$$

where $p_t = \mathbf{w}_t \cdot \mathbf{x}_t$ and $\delta > 0$ is a sampling factor parameter to trade off the ratio of queries. Such an approach is similar to the idea of margin-based active learning [31, 2] and has been adopted in other previous work [6, 12]. If the outcome $Z_t = 1$, the class label is queried and the outcome y_t is disclosed, then the CSOAL algorithm will adopt the proposed updating method (6) to update the linear classification model $\mathbf{w}_{t+1}$. If $Z_t = 0$, the class label will not be queried and the learner is not updated. Finally, Algorithm 1 summarizes the details of the proposed CSOAL algorithms.

Remark. It is interesting to analyze the impact of the sampling factor parameter δ. In general, the larger the value of δ, the larger the resulting number of queries issued by the online active learner. In particular, when setting $\delta \to \infty$, it is reduced to the extreme case of querying class label of every instance in the online learning process. In general, one can simply fix δ to some constant to trade off a proper ratio of queries. Besides, an even better approach is to adaptively change the value of δ during the online learning process. In particular, we expect to query more examples at the beginning of the online learning task in order to build a good classifier, and gradually reduce the ratio of queries when the classifier becomes more and more accurate during the online learning process. To this purpose, we suggest a simple yet effective scheme to adaptively update the parameter δ at the t-th learning step as: $\delta_t \leftarrow \delta_{t-1} * \frac{t}{t+1}$. We will examine the impact of the sampling factor δ in our experiments.

Algorithm 1 Cost-Sensitive Online Active Learning algorithm (**CSOAL**).

INPUT: penalty parameter C, bias parameter ρ and smooth parameter δ.
INITIALIZATION : $\mathbf{w}_1 = \mathbf{0}$.
for $t = 1, \ldots, T$ **do**
  receive an incoming instance $\mathbf{x}_t \in \mathbb{R}^d$;
  predict label $\hat{y}_t = \text{sign}(p_t)$, where $p_t = \mathbf{w}_t \cdot \mathbf{x}_t$;
  draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\delta/(\delta + |p_t|)$;
  if $Z_t = 1$ **then**
    query label $y_t \in \{-1, +1\}$;
    suffer loss $\ell_t(\mathbf{w}_t) = \ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))$;
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where $\tau_t = \min\{C, \ell_t(\mathbf{w}_t)\}$;
  else
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where $\tau_t = 0$;
  end if
end for

3.4 Time and Space Complexity

From Algorithm 1, it is obvious to see that the overall time complexity of the algorithm is $O(T \times d)$, which is linear with respect to T — the total number of instances in the online malicious URL detection task and d — the dimensionality of the input, and the space complexity of each learning step is $O(d)$ linear with respect to the data dimensionality. In practice, when the data set is sparse and high-dimensional (d can be large), one can exploit the sparse implementation trick to further reduce the time and space cost considerably.

4. ANALYSIS OF THEORETICAL BOUNDS

Although the proposed CSOAL algorithm is simple, it is the first approach proposed for online malicious URL detection tasks. Below gives theoretical analysis of its performance for malicious URL detection tasks in terms of two types of performance metrics. Before presenting the bounds, we begin by showing a lemma which would facilitate the proofs in this section. With this lemma, we could then derive the performance bounds for the CSOAL algorithm. For convenience, we introduce the following notations:

$\mathcal{M} = \{t \,|y_t \neq \hat{y}_t,\, t \in [T]\}$, $\mathcal{L} = \{t | y_t = \hat{y}_t, \ell_t(\mathbf{w}_t) > 0, t \in [T]\}$,

where $[T] = \{1, \ldots, T\}$.

LEMMA 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of input instances, where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ for all t. Let $\tau_t = \min(C, \ell_t(\mathbf{w}_t))$, then the following bound holds for any $\mathbf{w} \in \mathbb{R}^d$ and $\alpha > 0$*

$$\sum_{t=1}^{T} Z_t 2\tau_t \big[l_t(\alpha - |p_t|) + m_t(\alpha + |p_t|)\big] \leq \alpha^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T} \tau_t^2 \|\mathbf{x}_t\|^2 + \sum_{t=1}^{T} 2\alpha\tau_t \ell_t(\mathbf{w}),$$

where $m_t = \mathbb{I}_{(t \in \mathcal{M})}$ and $l_t = \mathbb{I}_{(t \in \mathcal{L})}$.

The detailed proofs of Lemma 1 and the following theorem are omitted, due to space limitation. According to the above lemma, we can prove the following theorem that bounds the expected weighted summation of mistakes, which can further bound the the cost-sensitive metrics for online malicious URL detection tasks.

THEOREM 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples where $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| = 1$ for all t. Then, for any vector $\mathbf{w} \in \mathbb{R}^d$, the expected weighted number of prediction mistakes made by CSOAL on this sequence of examples is bounded as:*

$$\mathbb{E}[\sum_{t=1}^{T} \rho_t m_t] \leq \frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\},$$

where $C \geq \rho$ is the aggressiveness parameter for CSOAL.

Now our goal is to analyze the performance of the proposed algorithm in terms of the metrics for malicious URL detection. We first consider the weighted sum of sensitivity and specificity, i.e.,

$$sum = \eta_p \times sensitivity + \eta_n \times specificity,$$

where $\eta_p + \eta_n = 1$ and $\eta_p \geq \eta_n > 0$. The following theorem gives the bound on the sum by the proposed CSOAL algorithm.

THEOREM 2. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| = 1$ for all t. By setting $\rho = \frac{\eta_p T_n}{\eta_n T_p}$, and assuming $C \geq \rho$, for any $\mathbf{w} \in \mathbb{R}^d$, we have the following bound for the proposed CSOAL algorithm:*

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}$$

Furthermore, when $\eta_p = \eta_n = 1/2$, the balanced accuracy (BA) is bounded from below by

$$\mathbb{E}[BA] \geq 1 - \frac{1}{2T_n}\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}$$

PROOF. Following the condition that $\rho = \frac{\eta_p T_n}{\eta_n T_p} \geq 1$ and the result of Theorem 1, we have

$$\begin{aligned}
&\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\} \\
&\geq \big(\rho\mathbb{E}M_p + \mathbb{E}M_n\big) \\
&= \big[(\frac{\eta_p T_n}{\eta_n T_p})\mathbb{E}M_p + \mathbb{E}M_n\big] = \frac{T_n}{\eta_n}\big[\eta_p(\frac{\mathbb{E}M_p}{T_p}) + \eta_n\frac{\mathbb{E}M_n}{T_n}\big] \\
&= \frac{T_n}{\eta_n}\big(\eta_p(1-\mathbb{E}sen) + \eta_n(1-\mathbb{E}spe)\big) \\
&= \frac{T_n}{\eta_n}(1 - \mathbb{E}[sum])
\end{aligned}$$

Rearranging the above inequality leads to the conclusion:

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}$$

□

Remarks. In the above, setting $\delta = 1$ leads to the following bound

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n}\left\{\|\mathbf{w}\|^2 + 2C\sum_{t=1}^{T}\ell_t(\mathbf{w})\right\}.$$

Setting $\delta = \sqrt{1 + \frac{4C\sum_{t=1}^{T}\ell_t(\mathbf{w}_t)}{\|\mathbf{w}\|^2}}$ leads to the following bound

$$\mathbb{E}[sum] \geq 1 - \frac{\eta_n}{T_n} * \left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T}\ell_t(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}\|\sqrt{\|\mathbf{w}\|^2 + 4C\sum_{t=1}^{T}\ell_t(\mathbf{w})}\right\}.$$

In the above approach, the bias parameter ρ is set to $\frac{\eta_p T_n}{\eta_n T_p}$, in which the ratio $\frac{T_n}{T_p}$ may not be available in advance. To alleviate this issue, we consider another approach using the cost based performance metric. Specifically, we propose to set $\rho = \frac{c_p}{c_n}$, where c_p and c_n are the predefined cost parameters of false negative and false positive, respectively. We assume $c_p + c_n = 1$ and $0 \leq c_n \leq c_p$ since we would prefer to improve the accuracy of predicting the rare positive examples. By this setting, the following theorem gives us the cumulative cost bound of the proposed CSOAL algorithm.

THEOREM 3. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples, where $\mathbf{x}_t \in \mathbb{R}^d$, $y_t \in \{-1, +1\}$ and $\|\mathbf{x}_t\| = 1$ for all t. By setting $\rho = \frac{c_p}{c_n}$, and assuming $C \geq \rho$, for any $\mathbf{w} \in \mathbb{R}^d$, the overall cost made by the proposed CSOAL algorithm over this sequence of examples is bounded as follows:*

$$\mathbb{E}[cost] \leq c_n\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\}$$

PROOF. Following the result of Theorem 1, we have

$$\begin{aligned}
&\frac{1}{\delta}\left\{(\frac{1+\delta}{2})^2\|\mathbf{w}\|^2 + \sum_{t=1}^{T}(1+\delta)C\ell_t(\mathbf{w})\right\} \\
&\geq (\mathbb{E}M_p(\rho) + \mathbb{E}M_n) \\
&= (\mathbb{E}M_p(\frac{c_p}{c_n}) + \mathbb{E}M_n) = \frac{1}{c_n}\mathbb{E}[cost]
\end{aligned}$$

Rearranging the above inequality concludes the theorem. □

Remarks. Setting $\delta = 1$ for the above theorem leads to the following bound:

$$\mathbb{E}[cost] \leq c_n\left\{\|\mathbf{w}\|^2 + 2C\sum_{t=1}^{T}\ell_t(\mathbf{w})\right\}.$$

Setting $\delta = \sqrt{1 + \frac{4C\sum_{t=1}^{T}\ell_t(\mathbf{w}_t)}{\|\mathbf{w}\|^2}}$ leads to the following bound:

$$\mathbb{E}[cost] \leq c_n\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{t=1}^{T}\ell_t(\mathbf{w}) + \frac{1}{2}\|\mathbf{w}\|\sqrt{\|\mathbf{w}\|^2 + 4C\sum_{t=1}^{T}\ell_t(\mathbf{w})}\right\}.$$

5. EXPERIMENTAL RESULTS

This section will evaluate the empirical performance of the proposed CSOAL algorithm for large-scale online malicious URL detection. Our experiments are designed to answer

several open questions: (i) how does the class imbalance issue affect the predictive performance of online malicious URL detection? (ii) if the proposed online active learning approach is effective to reducing the amount of labeled data significantly in order to maintain comparable performance?(iii) how is the efficiency and scalability of the proposed learning algorithms for a web-scale application?

5.1 Experimental Testbed

To examine the performance, we test all the algorithms on a large-scale benchmark dataset for malicious URL detection, which can be downloaded from [1]. The original data set was created in purpose to make it somehow class-balanced. In our experiment, we create a subset by sampling from the original data set to make it close to a more realistic distribution scenario where the number of normal URLs is significantly larger than the number of malicious URLs. Table 1 shows the data set used in our experiment for online malicious detection, where T_p/T_n denotes the ratio between the number of positive (malicious) instances and the number of negative (normal) instances. A variety of features were extracted to represent the content of a URL, including both lexical features (such as hostnames, primary domain, path tokens, etc) and host-based features (such as WHOIS info, IP prefix, AS number, Geographic, etc.).

Table 1: The data set of malicious URL detection.

dataset	# training examples	# features	T_p/T_n
URL	1,000,000	3,231,961	1:99

Note that we also did experiments on varied ratios of T_p/T_n and found the results were consistent to our observations and conclusions reported in this work. Thus, to keep the paper concise and easy to follow, we omit the details of duplicate results due to space limitation. All the datasets, code, and supplemental material will be made available in our project website: `http://murl.stevenhoi.org/`.

5.2 Compared Algorithms and Setup

We compare the proposed CSOAL algorithms against a variety of state-of-the-art algorithms as follows:

- "PE": the classical PErceptron algorithm [29], which queries label of every instance; this is impractical as it requires huge amount of labeled data, which is used as a yardstick to evaluate the efficacy of our algorithm;
- "PA": the regular Passive-Aggressive algorithm [9], which also queries class label of every instance; similarly, this is another yardstick for comparison;
- "CW-diag": the Confidence Weighted (CW) algorithm [10], which also queries label of every instance, and exploits the second-order info. we adopt the CW-diag version to make it feasible for high-dimensional data.
- "PAUM": this is the cost-sensitive Perceptron Algorithm with Uneven Margin [24], which also queries label of every instance;
- "CPA": the Cost-sensitive Passive-Aggressive algorithm based on prediction [9] which also queries all labels;
- "LEPE": the Label Efficient PErceptron algorithm [6], which actively queries label for informative instances;
- "CSRND": a variant of the proposed CSOAL algorithm, but *randomly* queries label of incoming instances;
- "CSOAL": the proposed Cost-Sensitive Online Active Learning algorithm as shown in Algorithm 1.

To make a fair comparison, all the algorithms adopt the same setup. All the compared algorithms learn a linear classifier for the malicious URL detection task. In particular, for all the compared algorithms, we set the penalty parameter $C = \rho = T_n/T_p$. For the proposed CSOAL$_{sum}$ algorithm, we set $\eta_p = \eta_n = 1/2$ for all cases, while for the CSOAL$_{cos}$, we set $c_p = T_n/T$ and $c_n = T_p/T$. The smoothing parameter δ for LEPE and CSOAL is set as $2^{[-10:2:10]}$ in order to examine varied ratios.

All the experiments were conducted over 5 random permutations of the dataset. The results were reported by averaging over these 5 runs. We evaluate the online classification performance by two key metrics: the weighted **sum** of sensitivity and specificity, and the weighted **cost**. We denote by CSOAL$_{sum}$ the algorithm aiming to improve the weighted sum of sensitivity and specificity, and CSOAL$_{cos}$ the algorithm aiming to improve the overall cost. All experiments were run on a machine of 2.3GHz CPU.

5.3 Evaluation on Fixed Ratio of Queries

The first experiment is to evaluate the performance by fixing the ratio of queries issued by the (active learning) algorithms. Table 2 shows the results of the *sum* performance under a fixed ratio of queries to about 2%, and Table 3 summarizes the *cost* performance under the similar query ratio.

Several observations can be drawn from the results. First of all, according the classification *accuracy* (a misleading metric for cost-sensitive classification), we found that both PE and PA algorithms significantly outperform the other algorithms, while, in terms of both *sum* and *cost* measures, they are considerably worse than their cost-sensitive variants (i.e., PAUM and CPA). This indicates the importance of taking the class imbalance issue into consideration for online malicious detection tasks. Second, when querying the same ratio of labeled data, in terms of both *sum* and *cost* performances, CSOAL significantly outperforms the LEPE algorithm, which validates the effectiveness of the proposed cost-sensitive online updating strategy. Third, when querying the same ratio of labels, CSOAL significantly outperforms CSRND, which implies the proposed querying strategy is able to actively select those fairly informative instances for querying labels, which are considerably better than just randomly querying. Moreover, among all the approaches, the proposed CSOAL algorithm and the PAUM algorithm achieve the highest *sum* performance. However, the proposed CSOAL only queried 2 percent of labels, while the PAUM algorithm requires to query the labels of all the incoming instances, which is very expensive to label 1-million training instances in a real-world application. We thus believe the proposed CSOAL algorithm is more practically attractive and suitable for a web-scale application.

Finally, we notice that the proposed CSOAL algorithm is able to achieves the best *sensitivity* performance, while at the same time achieves fairly good specificity performance which is generally quite comparable to the other algorithms. This implies that the proposed CSOAL algorithm can not only significantly improve the prediction accuracy on the rare class, but also not sacrifice much the prediction accuracy on classifying the other majority class. This promising observation again validates the effectiveness of the proposed CSOAL algorithm.

[1] http://sysnet.ucsd.edu/projects/url/

Table 2: Evaluation of the malicious URL detection performance in terms of the cumulative *sum* measure.

Algorithm	Measures					
	Sum (%)	Sensitivity(%)	Specificity (%)	Accuracy (%)	Time (s)	Query Ratio (%)
PE	87.012 ± 0.100	74.284 ± 0.199	99.741 ± 0.002	99.486 ± 0.004	18.903	100.000 ± 0.000
PA	87.203 ± 0.059	74.544 ± 0.115	99.862 ± 0.003	99.609 ± 0.004	27.458	100.000 ± 0.000
CW-diag	88.550 ± 0.067	77.160 ± 0.133	99.940 ± 0.001	99.712 ± 0.002	48.616	100.000 ± 0.000
PAUM	89.049 ± 0.083	78.770 ± 0.166	99.329 ± 0.002	99.123 ± 0.003	28.527	100.000 ± 0.000
CPA	**92.748 ± 0.078**	86.410 ± 0.154	99.087 ± 0.005	98.960 ± 0.006	41.248	100.000 ± 0.000
LEPE	79.162 ± 0.476	58.492 ± 0.957	99.833 ± 0.011	99.419 ± 0.010	19.414	**2.019 ± 0.057**
CSRND	87.776 ± 0.410	79.018 ± 0.711	96.534 ± 0.286	96.358 ± 0.284	20.984	**2.018 ± 0.025**
CSOAL	**92.697 ± 0.245**	88.156 ± 0.513	97.237 ± 0.045	97.146 ± 0.042	20.304	**2.029 ± 0.018**

Table 3: Evaluation of the malicious URL detection performance in terms of the cumulative *cost* measure.

Algorithm	Measures					
	Cost	Sensitivity(%)	Specificity (%)	Accuracy (%)	Time (s)	Query Ratio (%)
PE	2571.568 ± 19.862	74.284 ± 0.199	99.741 ± 0.002	99.486 ± 0.004	19.994	100.000 ± 0.000
PA	2533.800 ± 11.678	74.544 ± 0.115	99.862 ± 0.003	99.609 ± 0.004	28.800	100.000 ± 0.000
CW-diag	2267.124 ± 13.216	77.160 ± 0.133	99.940 ± 0.001	99.712 ± 0.002	47.747	100.000 ± 0.000
PAUM	2057.452 ± 20.843	79.840 ± 0.208	99.378 ± 0.005	99.182 ± 0.006	28.939	100.000 ± 0.000
CPA	**1435.806 ± 15.494**	86.410 ± 0.154	99.087 ± 0.005	98.960 ± 0.006	42.687	100.000 ± 0.000
LEPE	4214.998 ± 125.053	57.592 ± 1.275	99.832 ± 0.013	99.410 ± 0.007	21.655	**1.984 ± 0.045**
CSRND	2314.544 ± 126.476	80.030 ± 1.265	96.591 ± 0.130	96.425 ± 0.130	20.371	**2.027 ± 0.043**
CSOAL	1482.338 ± 31.270	87.742 ± 0.324	97.285 ± 0.029	97.189 ± 0.027	22.237	**2.027 ± 0.030**

5.4 Evaluation on Varied Ratios of Queries

This experiment is to evaluate the performance of the proposed algorithms by varying the ratios of queries for comparing different online malicious URL detection algorithms. Figure 2 and Figure 3 shows the online average *sum* performance and the online average *cost* performance under varied query ratios, respectively. From the experimental results, several observations can be drawn as follows.

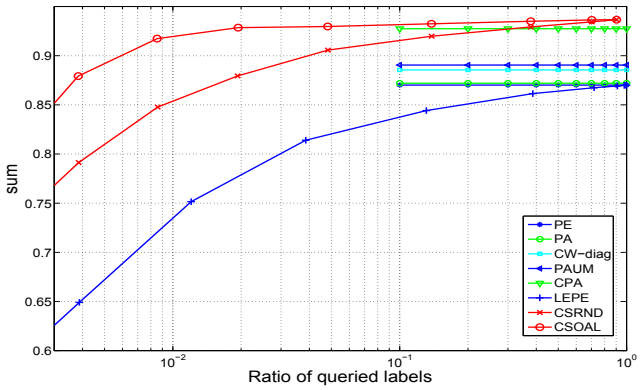

Figure 2: Evaluation of the online cumulative average *sum* performance with respect to varied ratios.

First of all, among all four fully supervised online learning algorithms (PE, PA, PAUM, and CPA), the cost-sensitive algorithms (PAUM and CPA) generally outperform the cost-insensitive versions. This result validates the importance of studying the proposed cost-sensitive online learning methodology for malicious URL detection tasks.

Second, compared with the CSRND algorithm that randomly queries the labels, CSOAL consistently achieves much higher *sum* and much lower *cost* performance over all the ratios of queried labels, especially when the query ratio is relatively small. This promising result indicates that the querying strategy of the proposed CSOAL technique is able to effectively query the informative labeled data from the online stream of unlabeled data instances.

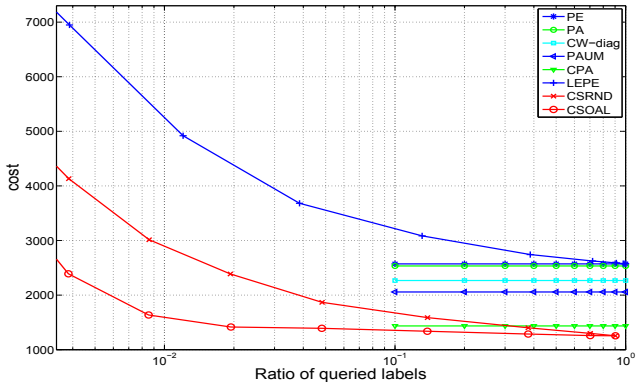

Figure 3: Evaluation of the online cumulative average *cost* performance with respect to varied ratios.

Third, compared with LEPE, CSOAL achieves higher *sum* over all the ratios of queried labels, which implies that the proposed online updating strategy is able to effectively exploits the labeled data for improving the classifier. In addition, compared with PA, CSOAL with query ratio equals to 1 (equivalent to querying label of every instance) achieves a significantly higher *sum* performance, which shows the biased penalty function does effectively optimize the objective metric of the weighted sum of sensitivity and specificity.

Finally, we notice that when the query ratio increases, we generally observe an improvement of the cost-sensitive classification performance by the proposed CSOAL algorithm. However, While the query ratio reaches about 1%, the improvement tends to become saturated, which is very close to the same algorithm that queries the label of every unlabeled data. This interesting observation indicates that the proposed learning strategy is able to attain potentially the best possible predictive performance using a small amount

Table 4: Evaluation of the malicious URL detection performance in terms of both *sum* and *cost* metrics.

Algorithm	Measures					
	Sum (%)	Sensitivity(%)	Specificity (%)	Accuracy (%)	Time (s)	Query Ratio (%)
CPA	**92.748 ± 0.078**	86.410 ± 0.154	99.087 ± 0.005	98.960 ± 0.006	37.087	100.000 ± 0.000
LEPE	69.556 ± 1.353	39.274 ± 2.712	99.838 ± 0.015	99.232 ± 0.025	16.777	**0.515 ± 0.017**
CSRND	80.724 ± 1.852	66.578 ± 3.855	94.871 ± 0.206	94.588 ± 0.177	17.039	**0.526 ± 0.011**
CSOAL	88.756 ± 0.746	83.628 ± 1.701	93.883 ± 0.373	93.781 ± 0.359	17.260	**0.513 ± 0.020**
CSOAL(a)	**92.401 ± 0.703**	89.054 ± 1.810	95.748 ± 0.406	95.681 ± 0.384	18.211	**0.510 ± 0.014**
Algorithm	Measures					
	Cost	Sensitivity(%)	Specificity (%)	Accuracy (%)	Time (s)	Query Ratio (%)
CPA	**1435.806 ± 15.494**	86.410 ± 0.154	99.087 ± 0.005	98.960 ± 0.006	36.640	100.000 ± 0.000
LEPE	6170.384 ± 152.639	37.818 ± 1.549	99.855 ± 0.009	99.235 ± 0.009	17.137	**0.525 ± 0.023**
CSRND	3618.938 ± 466.228	68.634 ± 5.240	94.811 ± 0.598	94.549 ± 0.546	16.843	**0.522 ± 0.017**
CSOAL	2265.136 ± 299.126	82.916 ± 3.226	94.204 ± 0.275	94.091 ± 0.249	17.603	**0.525 ± 0.017**
CSOAL(a)	**1484.396 ± 117.269**	89.498 ± 0.831	95.508 ± 0.490	95.448 ± 0.490	17.917	**0.525 ± 0.015**

of label data (only 1% or even less) over the entire training data set, which can thus save a significant amount of labeling cost in a practical real-world application.

5.5 Evaluation on Adaptive Sampling Factor

In the above experiments, the sampling factor δ was simply fixed to a constant. This experiment aims to examine if it is possible to further improve the proposed CSOAL approach using the adaptive sampling factor, denoted as "CSOAL(a)" for short (as discussed in the "remark" of Section 3.3). In this experiment, the initial value of δ is set to an extremely large value, i.e., $\delta_0 = 2^{14}$, and is updated adaptively using the proposed strategy in Section 3.3. To enable a fair comparison, we set appropriate parameters of the other algorithms (LEPE, CSRND and CSOAL) to make them sample the similar ratio of labeled data. Table 4 shows the experimental results, where "CSOAL" adopts the constant sampling factor. Some observations can be drawn from the results. First, the CSOAL(a) algorithm using the adaptive sampling factor significantly outperforms both CSRND using the random query strategy and CSOAL using a constant sampling factor under the same query ratio. Second, we found that by querying only 0.5% out of the entire 1-million instances, the proposed CSOAL(a) algorithm is able to achieve the best performance, which is almost the same (statistically no difference according to student t-test) to the state-of-the-art cost-sensitive algorithm CPA which has to query labels for all the 1-million instances. This promising result shows that the proposed CSOAL technique is able to save a significant amount of labeling cost while maintaining the state-of-the-art performance.

5.6 Evaluation on Eff ciency and Scalability

Finally, we examine the time efficiency of the proposed algorithms. The "time" columns of Table 2, 3 and 4 show the average time costs of the proposed CSOAL algorithms on the fixed query ratios. In addition to these tables, we also evaluate the scalability of the proposed algorithms, as shown in Figure 4, which measures the online cumulative time cost of different algorithms over the number of received instances in the online malicious URL detection process.

From the results, we can see that all the proposed online learning algorithms are fairly efficient and scalable, which typically took about 20 to 30 seconds to run on the data set with 1-million instances on a single regular machine. Moreover, by examining the efficiency and scalability of the proposed CSOAL algorithms, we found that CSOAL is among the most efficient and scalable algorithms, which is at least as efficient as the other algorithms and even slightly better than some of the other algorithms. These encouraging results again validate the practical value of the proposed CSOAL algorithm for web-scale real-world applications.

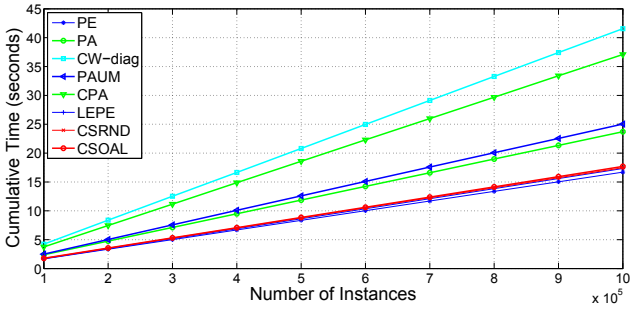

Figure 4: Evaluation of online cumulative time cost.

6. CONCLUSIONS

This paper proposed a novel framework of cost-sensitive online active learning (CSOAL) as a natural, simple yet fairly effective approach to tackling a real-world online malicious URL detection task. We presented the CSOAL algorithms to optimize cost-sensitive measures and theoretically analyze the bounds of the proposed algorithms. We also extensively examined their empirical performance on a large-scale real-world data set. Our encouraging results showed that (i) the proposed CSOAL method is able to considerably outperform a number of supervised cost-sensitive or cost-insensitive online learning algorithms for malicious URL detection tasks; (ii) the proposed CSOAL method is able to attain the comparable (or even better) state-of-the-art predictive performance of a cost-sensitive online learner by querying a significantly small amount of labeled data (0.5% or less);and(iii)the proposed CSOAL algorithms are highly efficient and scalable for web-scale applications.

Acknowledgements

This work is supported by Singapore MOE Academic tier-1 grant (RG33/11).

7. REFERENCES

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, pages 39–50, 2004.

[2] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *COLT*, pages 35–50, 2007.

[3] E. Baykan, M. R. Henzinger, L. Marian, and I. Weber. Purely url-based topic classification. In *WWW*, pages 1109–1110, 2009.

[4] G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear classification and selective sampling under low noise conditions. In *NIPS 21*, pages 249–256, 2008.

[5] N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. on Inf. Theory*, 50(9):2050–2057, 2004.

[6] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Worst-case analysis of selective sampling for linear classification. *JMLR*, 7:1205–1230, 2006.

[7] N. Cesa-bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. *IEEE Trans. Inform. Theory*, 51:77–92, 2005.

[8] H. Choi, B. B. Zhu, and H. Lee. Detecting malicious web links and identifying their attack types. In *Proceedings of the 2nd USENIX conference on Web application development*, WebApps'11, pages 11–11, Berkeley, CA, USA, 2011. USENIX Association.

[9] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *JMLR*, 7:551–585, 2006.

[10] K. Crammer, M. Dredze, and F. Pereira. Exact convex confidence-weighted learning. In *NIPS*, pages 345–352, 2008.

[11] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *JMLR*, 3:951–991, 2003.

[12] S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

[13] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

[14] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Mach. Learn.*, 28(2-3):133–168, 1997.

[15] C. Gentile. A new approximate maximal margin classification algorithm. *JMLR*, 2:213–242, 2001.

[16] D. Helmbold and S. Panizza. Some label efficient learning results. In *COLT'97*, pages 218–230, Nashville, Tennessee, United States, 1997.

[17] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *ICML*, pages 417–424. ACM, 2006.

[18] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Semisupervised svm batch mode active learning with applications to image retrieval. *ACM Transactions on Information Systems (TOIS)*, 27(3):16, 2009.

[19] S. C. H. Hoi, J. Wang, and P. Zhao. LIBOL*: A Library for Online Learning Algorithms*. Nanyang Technological University, 2012.

[20] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In *CIKM*, pages 325–326, 2005.

[21] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In *NIPS*, pages 785–792, 2001.

[22] S. Li and I. W. Tsang. Maximum margin/volume outlier detection. In *ICTAI*, pages 385–392, 2011.

[23] Y. Li and P. M. Long. The relaxed online maximum margin algorithm. In *NIPS*, pages 498–504, 1999.

[24] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. S. Kandola. The perceptron algorithm with uneven margins. In *ICML*, pages 379–386, 2002.

[25] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. In *KDD*, pages 1245–1254, 2009.

[26] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious urls: an application of large-scale online learning. In *ICML*, page 86, 2009.

[27] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Learning to detect malicious urls. *ACM TIST*, 2(3):30, 2011.

[28] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *WWW*, pages 83–92, Edinburgh, Scotland, 2006.

[29] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[30] S. Shalev-Shwartz and Y. Singer. Online learning:theory, algorithms, and applications. In *Ph.D thesis*, 2007.

[31] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.

[32] J. Wang, P. Zhao, and S. C. H. Hoi. Cost-sensitive online classification. In *ICDM*, pages 1140–1145, 2012.

[33] J. Wang, P. Zhao, and S. C. H. Hoi. Exact soft confidence-weighted learning. In *ICML*, 2012.

[34] C. Whittaker, B. Ryner, and M. Nazif. Large-scale automatic classification of phishing pages. In *NDSS*, 2010.

[35] G. Xiang and J. I. Hong. A hybrid phish detection approach by identity discovery and keywords retrieval. In *WWW*, pages 571–580, New York, NY, USA, 2009. ACM.

[36] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov. Spamming botnets: signatures and characteristics. *SIGCOMM Comput. Commun. Rev.*, 38(4):171–182, Aug. 2008.

[37] J. Zhang, P. Porras, and J. Ullrich. Highly predictive blacklisting. In *Proceedings of the 17th conference on Security symposium*, SS'08, pages 107–122, Berkeley, CA, USA, 2008. USENIX Association.

[38] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *WWW*, pages 639–648, New York, NY, USA, 2007. ACM.

[39] P. Zhao, S. C. H. Hoi, and R. Jin. Double updating online learning. *Journal of Machine Learning Research*, 12:1587–1615, 2011.