Naive Bayes Classifier for Cost Sensitive Dynamic Learning

Mr. Jadhav Bharat S[1] Mr. Sandip A. Kahate[2]
[1]M. E. Student [2]Assistant Professor
[1,2]Department of Computer Engineering
[1,2]Sharadchandra Pawar College of Engineering, Dumbarwadi, Tal-Junnar, Dist-Pune-410504.(M.S.), India

Abstract— Both cost-sensitive classification and online learning have been broadly researched in data mining and machine learning communities, respectively. However, very restricted study addresses an essential intersecting issue, that is, "Cost-Sensitive Online Classification". In this paper we proposed, formally study this issue, and propose a new framework for Cost-Sensitive Online Classification by directly surging cost-sensitive symptoms implementing online gradient descent methods. Particularly, we present two novel cost-sensitive online classification algorithms, which are structured to directly amend two well-known cost-sensitive measures: (i) maximization of weighted sum of sensitivity and specificity, and (ii) minimization of weighted misclassification cost. We study the theoretical leap of the cost-sensitive measures made by the presented algorithms, and broadly observed their factual operation on a variety of cost-sensitive online classification works. Finally, we represent the application of the presented method for solving several online problem detection works, showing that the presented method could be a highly effective and efficient tool to tackle cost-sensitive online classification works in several application domains. malicious Uniform Resource Locator (URL) detection is an essential issue in web search and mining, which plays a complex role in internet protection.

Key words: Naive Bayes Classifier, Cost Sensitive Dynamic Learning

I. INTRODUCTION

In the age of big data, an immediate requirement in data mining and machine learning is to build effective and ascendable algorithms for mining immense quickly growing data. A promising direction is to verify *Online Learning*, a family of effective and ascendable machine learning techniques, which has been strongly analyzed in literature [1], [6], [18]. In general, the aim of online learning is to increasingly learn some divination models to make accurate divinations on a runnel of examples that arrive consequently. Online learning is benefited for its high coherence and ascendable for large-scale applications, and has been implemented to resolve online classification work in a various real-world data mining applications. Several online learning techniques have been actively presented in literature [6], [18]. Examples include the popular Perceptron algorithm, Passive-aggressive (PA) learning [6], and many other earlier presented algorithms [10], [13], [14]. notwithstanding being analyzed immensely, most present online learning methods are inappropriate for *cost-sensitive classification* operation, an measure issue for data mining which has to address assorted misclassification expense [9], [12]. The present online learning methods strongly might not be effective sufficient basically reason why most existing online learning analysis usually treat the operation of an online classification algorithm in terms of divination mistake rate or accuracy, which is clearly cost-insensitive and thus inappropriate for many actual applications in data mining, specifically for cost-sensitive classification work where datasets are usual class-imbalanced and the misclassification expense of occurrences from various classes can be very diverse [5], [11]. To address the above goal of cost-sensitive classification, researchers specifically in data mining literature have presented more significant metrics, such as the weighted sum of sensitivity and specificity and the weighted misclassification cost [1], [12]. Over the past years, considerable studies endeavors have been devoted to developing batch classification algorithms to enhance the cost-sensitive issues, including the weighted sum of sensitivity and specificity and the weighted misclassification cost metrics [1], [12]. However, these batch classification algorithms usual suffer indigent efficiency and ascendable when solving large-scale issues, which thus are improper for online classification applications. Although both cost-sensitive classification and online learning have been researched broadly in data mining and machine learning communities, respectively, there were very few inclusive studies on "Cost-Sensitive Online Classification" in both data mining and machine learning literature.

II. LITERATURE SURVEY

A. Related Work and Background

Our work is importantly concerned to three groups of research in data mining and machine learning: (i) cost-sensitive classification in data mining literature, (ii) online learning in machine learning literature, (iii) problem detection in both data mining and machine learning literature.

1) Cost-Sensitive Classification

Cost-sensitive grouping has been broadly analyzed in data mining and machine learning [20], Many real-world classification issues, such as fraud detection and medical diagnosis, are naturally cost-sensitive. For these issue, the cost of misclassifying a goal is much higher than that of a false-positive, and classifiers that are excellent under equality costs tend to under perform. The popular examples include the weighted sum of sensitivity and specificity and the weighted misclassification cost that takes cost into deliberation when computing classification operation [1], [12]. As a special case, when the weights are both equivalent to 0.5, the weighted sum of reactivity and specificity is diminished to the well-known balanced accuracy, which is broadly implemented in problem detection work. Over the past years, several batch learning algorithms have been presented for cost-sensitive classification in literature [9], [12] However, few studies signifies the case when data appears orderly, except the

Cost-sensitive Passive Aggressive(CPA) [6] and Perceptron Algorithms with Uneven Margin(PAUM) [21].

2) Online Learning

Online learning works on a order of data examples with time stamps. At time step t, the algorithm processes an incoming example xt ∈ Rd by first predicting its label ˆyt ∈ {−1,+1}. After the discernment, the true label yt ∈ {−1,+1} is disclosed and then the loss l(yt,ˆyt), which is the difference between its discernment and the disclosed true label yt, is suffered. Finally, the loss is implemented to update the weights of the model based on some criterion. Overall, the aim of online learning is to minimize the accumulative fault over the entire order of data examples [17].

The most well-known online learning algorithm perhaps is Perceptron. Particularly, whenever the online learner makes a incorrect classification, the perceptron algorithm simply updates the classifier as follows:

$$wt+1 = wt + ytxt.$$

Passive Aggressive (PA) learning [6] attempts to enhance Perceptron by showing the idea of margin escalation into the online learning mechanism. PA algorithms update the classifier whenever the online classifier does not make a large margin on the current received example.

Particularly, the loss of PA algorithms is based on the hinge loss: $l(wt; (xt, yt)) = \max\{0, 1-yt(wt \cdot xt)\}$. The escalation of the PA learning is formulated as:

$$wt+1 = \arg\min_{w \in Rd} 1/2||w - wt||2$$
$$\text{s.t. } l(w; (xt, yt)) = 0.$$

The closed-form solution to the above is expressed as:

$$wt+1 = wt + \eta tytxt, \quad (1)$$

where the optimal value of parameter

$$\eta t = (wt;(xt,yt))/||xt||2 .$$

To further make PA being able to manage non-separable instances, one can introduce a slack variable ξ into the escalation issue in (1):

$$wt+1 = \arg\min_{w \in Rd} 1/2||w - wt||2 + C\xi$$
$$\text{s.t. } l(w; (xt, yt)) \leq \xi \text{ and } \xi \geq 0.$$

The solution to the above soft-margin issue shares the same form as that of (1), but with different coefficient ηt as follows:

$$\eta t = \min\{C,(wt; (xt, yt))/||xt||2\}$$

The above two variants of PA algorithms are called "PA" and "PA-I", respectively. Unlike traditional first-order online learning algorithms (e.g., Perceptron and PA), Confidence-Weighted (CW) online learning [7], [10] considers the weight vector follows a Gaussian dispensing and updates the mean and covariance of the distribution for each received example. Particularly, consider the weight vector wt has the mean vector μ ∈ Rd and covariance matrix ∑ ∈ Rd×d, the CW learning operates the dispensing update by decreasing the Kullback-Leibler separation between the distributions of the new and old weight vectors, and meanwhile ensuring that the possibility of a accurate classification on the training instance is huge enough. Adaptive Regularization of Weights Learning (AROW) [8] was presented to overcome this limitation. However, to the best of our knowledge, very few presenting work in this area had attempted to directly optimize the two cost-sensitive metrics in an online learning setting, except which is based on online Naive Bayes approach. Also we note that our task is very different from another earlier online learning study, which results to amend AUC, but cannot be guaranteed to amend the cost-sensitive issue in our study. Finally, we note that this work is concentrated on observing online learning theory for learning linear models, and thus exclude the direct collation to other nonlinear online learning techniques [15], [17].

3) Anomaly Detection

Anomaly detection, also called as outlier detection or novelty detection, aims to search abnormal patterns ("anomalies") in data that do not grant with normal patterns/expected behaviors. It has been broadly analyzed over the past years in a several of research areas and application domains [4]. Anomaly detection methods have been widely implemented to manage issues in a wide range of real-world applications [4], such as detection of credit card fraud transactions, network intrusion detection, detection of abnormal jet engine performance, detection of malignant tumors from medical images, and so on. In literature, a several methods have been presented to solve anomaly detection in various application domains [3]. One major category of techniques produce anomaly detection as a classical supervised classification work by training a binary classification model in a batch/offline learning fashion to differentiate between anomalies and normal patterns. These methods often need to collect a substantial amount of training data in order to construct a good classification model for anomaly detection. In contrast, another type of methods produce it as an online unsupervised/semi supervised learning work to detect anomalies without needing label information of anomalies [21]. These methods however may suffer from poor detection operation without exploring any label/supervised data. Although anomaly detection has been well analyzed for a few years, it remains a very challenging research issue today, which is basically due to several reasons. First of all, it is often a highly class-imbalanced learning issue as the number of anomalies is importantly smaller than that of normal structure, which brings a critical challenge to many schemes implementing regular classification methods. Second, it is often very expensive to gather labeled data, especially the positive training data ("anomalies"), which restricts the application of some classical supervised classification approaches.

III. IMPLEMENTATION DETAILS

A. Classification:

In this paper the existing database is available to hospitals to check or verify the system goal i.e. whether number of person have diabetic symptoms or not. This is to be examined as per the previous conventional manner thorough which some outcomes are got in certain figures. The classification is done on the basis of positive and negative results; here database is trained for the proper results. The data of persons is checked on the basis of certain attributes likewise height, weight, gender. On the basis of these three features the person is to be diagnosed for diabetic symptoms.

B. Accuracy:

The accuracy is an essential part of any medical diagnosis systems. When the existing data is observed for the both results positive and negative. In such cases the results are

brought for the accurate outcomes but some times output are displayed wrong which leads to life risk of all the patients which were examined for diabetic issues. The life risk occurs due to negative and positive results, if this system shows negative results to the diabetic patient the it might be very dangerous. This proposed to sections C1and C2 , in C1 section positive results are stored and in C2 negative results. The C1 shows 80 results are accurate among the 100 observations and in C2 it shows 92 results positive. So C1 gives outcomes for life risk of 20 peoples is higher than compare to C2.

C. Online Classification:

The online classification is performed on the basis of existing results such if the some results are misleading or incorrect then it resolve the problem and update to the database for future operations.

This play a vital role in current system to advancement the standard and productivity of the system processing. Online classification bifurcate the positive and negative results very smoothly without occurring any errors while performing operations. The outputs are generated automatically as per the inserted inputs.

IV. DETECTING MALICIOUS WEBSITES:

Detecting Malicious Websites is the system in which the website which are very harmful or detrimental are detected and kept aside to avoid the uncertain risk. Malicious URL detection is about how to detect malicious URLs automatically or semi- automatically, which has been broadly analyzed in web and data mining communities for years [22, 23, 24]. In general, we can separate the current task into two categories: (i) non-machine learning techniques, such as blacklisting or rule-based approaches [25, 26]; and (ii) machine learning techniques. The non-machine learning approaches normally suffer from poor generalization to new malicious URLs and unseen malicious structure. In the following, we will focus on reviewing important relevant task implementation machine learning techniques[24][27].

1) Number of dots: There are many websites which has more than one dot in its URL (Uniform Resource Locator).The dots in URL represents its purpose and locations. This is an important stuff for all the websites.
2) Length: The length of website URL is to be measure under specific criterion which examines the actual size of the system.
3) Date: The launching date of the website is also an essential parameter for the detecting malicious websites. This record is maintained by this system for certain observations.
4) Registrar ID, name: The website contains registrar ID or name. Every website have its own registrar name where it is verified for the malicious activity under detection techniques.

V. NAÏVE BAYES CLASSIFIER

In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Consider a real-world online malicious URL detection issue where data example arrive orderly. In common case, this can be produce as a binary classification work where malevolent URL instances are from positive class (“+1”) and normal URL instances are from negative (“-1”). For an online malicious URL detection work, the goal is to build an online learner to incrementally build a classification model from a order of URL training data instances via an online learning fashion.

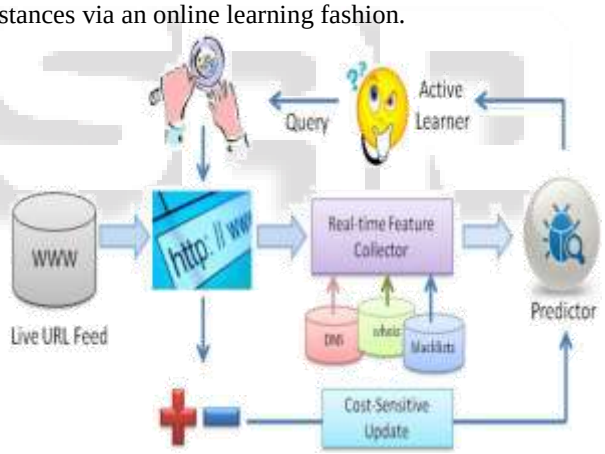

Fig. 1: Framework of the proposed CSOAL system for malicious URL detection

A. Cost Sensitive Updatable:

We present a method to assess route queries based on the novel concept of partial path occasions. Our method (1) maximizes operations by means of order scans or asynchronous I/O, (2) does not needed a special storage format, (3) relies on simple navigational primitives on trees, and (4) can be complemented by current logical and physical escalation such as duplicate destruction, duplicate avoidance and path rewriting. We implement a physical algebra which divides those navigation process that needed I/O from those that do not. All I/O operations essential for the evaluation of a path are outlying in a single operator, which may engaged effective I/O scheduling planning such as orderly scans or asynchronous I/O. Performance results for queries from the XMark benchmark displays that altering the navigation process can increase performance up to a factor of four.

VI. COST-SENSITIVE ONLINE CLASSIFICATIONS

In this section, we present our proposed Cost-Sensitive Online Classification (CSOC) mechanism, we first introduce the issues formulation and then existing the proposed algorithms.

A. Problem Formulation

Without loss of generalization, let us assume an online binary classification issue. At each learning round, the learner gets an instance and predicts its class label as "+1" or"-1". After making the prediction, the learner receives the true label of the instance and suffers a loss if the prediction is wrong. At the end of each round, the learner implement the received training example and it class label to update the prediction model.

Formally, let us denote by $x_t \in R^n$ the instance received at the t-th learning step, and $w_t \in R^n$ a linear prediction model learned from the earlier $t-1$ training examples. We also denote the prediction for the t-th instance as $\hat{y}_t = \text{sign}(w_t \cdot x_t)$, while the value $|w_t \cdot x_t|$, known as the "margin", is used as the confidence of the learner on the prophecy. The true label for instance x_t is denoted as $y_t \in \{-1,+1\}$. If $\hat{y}_t \neq y_t$, the learner made a mistake; otherwise it made a correct prediction.

For binary classification, the result of each prediction.For an instance can be classified into four cases: (1) True Positive (TP) if $\hat{y}_t = y_t = +1$; (2) False Positive (FP) if $\hat{y}_t = +1$ and $y_t = -1$; (3) True Negative (TN) if $\hat{y}_t = y_t = -1$; and (4) False Negative (FN) if $\hat{y}_t = -1$ and $y_t = +1$.

We now assume a order of training examples (x1, y1), . . . , (xT, yT) for online learning. Then, for convenience, we denote by M the set of indexes that correspond to the trials of misclassification:

$M = \{t \mid y_t \neq \text{sign}(w_t \cdot x_t), \forall t \in [T]\}$, where $[T] = \{1, \ldots, T\}$. Similarly, we denote by $M_p = \{t \mid t \in M \text{ and } y_t = +1\}$ the set of indexes for false negatives, and

$M_n = \{t \mid t \in M \text{ and } y_t = -1\}$ the set of indexes for false positives.

Further, we found notation $M = |M|$ to denote the number of mistakes, $M_p = |M_p|$ to denote the number of false negatives, and $M_n = |M_n|$ to denote the number of false positives. Also we implement notation $I_{pT} = \{i \in [T] \mid y_i = +1\}$ to denote the set of indexes of the positive examples, $I_{nT} = \{i \in [T] \mid y_i = -1\}$ to denote the set of indexes of negative examples,

$T_p = |I_p T|$ to denote the number of positive examples, and $T_n = |I_{nT}|$ to denote the number of negative examples.

For operations metrics, sensitivity is defined as the ratio between the number of true positives $T_p - M_p$ and the number of positive examples; specificity is defined as the ratio between $T_n - M_n$ and the number of negative examples; and accuracy is defined as the ratio between the number of correctly classified examples and the total number of examples.

These can be summarized as:

$\text{sensitivity} = T_p - M_p / T_p$,

$\text{specificity} = T_n - M_n / T_n$,

$\text{accuracy} = T - M / T$

Assume an online binary classification task, without loss of generalization, we consider positive class is the rare class, i.e., $T_p \leq T_n$, the number of positive examples is smaller than the number of negative examples. For simplicity, we also assume that $\|x_t\| \leq 1$. A more suitable metric is to count the sum of weighted sensitivity and specificity, i.e.,

$$\text{sum} = \eta_p \times \text{sensitivity} + \eta_n \times \text{specificity}, \quad (2)$$

where $\eta_p + \eta_n = 1$ and $0 \leq \eta_p, \eta_n \leq 1$ are two parameters to trade off between sensitivity and specificity. Notably, when $\eta_p = \eta_n = 0.5$, the corresponding sum is the well known balanced accuracy. In general, the higher the sum value, the better the classification operation. Besides, another approach is to measure the total misclassification cost suffered by the algorithm, which is defined as:

$$\text{cost} = c_p \times M_p + c_n \times M_n, \quad (3)$$

where $c_p + c_n = 1$ and $0 \leq c_p, c_n \leq 1$ are the misclassification cost inputs for positive and negative classes, respectively. The lower the cost value, the better the classification operation.

B. Algorithms

In this section, we present a mechanism of Cost-Sensitive Online Classification for cost-sensitive classification by escalation two cost-sensitive issues. Before showing our algorithms, we first prove the following main proposition that stimulate our solution.

Proposition 1. Consider a cost-sensitive classification problem, the goal of maximizing the weighted sum in (2) or minimizing the weighted cost in (3) is equivalent to minimizing the following objective:

$$\sum_{y_t=+1} \rho I(y_t w \cdot x_t < 0) + \sum_{y_t=-1} I(y_t w \cdot x_t < 0) \quad (4)$$

where $\rho = \eta_p T_n / \eta_n T_p$ for the maximization of the weighted sum, and $\rho = c_p / c_n$ for the minimization of the weighted misclassification cost.

Proof. Firstly, by analyzing the function of the weighted sum in (2), we can derive the following:

$$\text{sum} = \eta_p T_p - M_p / T_p + \eta_n T_n - M_n / T_n$$
$$= 1 - \eta_n / T_n [\eta_p T_n / \eta_n T_p \sum_{y_t=+1} I(y_t w \cdot x_t < 0) + \sum_{y_t=-1} I(y_t w \cdot x_t < 0)],$$

where I_π is the indicator function that outputs 1 if the statement π holds and 0 otherwise. Thus, maximizing sum is equivalent to minimizing

$$\eta_p T_n / \eta_n T_p \sum_{y_t=+1} I(y_t w \cdot x_t < 0) + \sum_{y_t=-1} I(y_t w \cdot x_t < 0).$$

Secondly, by analyzing the function of the weighted cost in (3), we can also derive the following:

$$\text{cost} = c_p M_p + c_n M_n$$
$$= c_n [\, c_p / c_n \sum_{y_t=+1} I(y_t w \cdot x_t < 0) + \sum_{y_t=-1} I(y_t w \cdot x_t < 0)].$$

Thus, minimizing cost is equivalent to minimizing c_p / c_n

$$\sum_{y_t=+1} I(y_t w \cdot x_t < 0) + \sum_{y_t=-1} I(y_t w \cdot x_t < 0).$$

Thus, the proposition holds by setting

$\rho = \eta_p T_n / \eta_n T_p$ for sum, and $\rho = c_p / c_n$ for cost.

Proposition 1 gives the explicit objective function for escalation, but the indicator function is not convex. To provide the online escalation task, we replace the indicator function by its convex surrogate, i.e., either one of the following modified hinge loss functions:

$$lI(w; (x, y)) = \max(0, (\rho * I(y=1) + I(y=-1)) - y(w \cdot x)) \quad (5)$$

$$lII(w; (x, y)) = (\rho * I(y=1) + I(y=-1)) * \max(0, 1 - y(w \cdot x)). \quad (6)$$

We could see that for $lI(w; (x, y))$, the needed margin for particular class modified compared to the conventional hinge loss, cause to more "frequent" updating;

while for lII(w; (x, y)), the slope of the loss function changed for specific class, leading to more "aggressive" updating. Fig. 1 illustrates the dissimilar of the modified hinge loss functions. As a result, we can formulate the optimization issue for cost-sensitive classification as follows:

$$F* T(w)= ½||w2 + C\sum Tt=1 l* (w; (xt, yt)) \text{ here } * \in \{I, II\}, \quad (7)$$

where ||w||2 is introduced to regularize the complexity of the linear classifier and C is a positive penalty parameter of the cumulative loss. The idea of the above formulation is somewhat relevant to the biased formulation of batch SVM for learning with imbalanced datasets [1]. Now our goal is to find an online learning solution to tackle the above convex optimization (7). To this end, we presented to solve the issue implementing the online gradient descent approach [51] as follows:

$$wt+1 = wt − \lambda \nabla t(wt),$$

where λ is a learning rate parameter and lt(w) = l∗ (w; (xt, yt)), ∀ ∗ ∈ {I, II}. Specifically, when using the loss function (5), the update rule can be expressed as:

$$wt+1 = \{wt + \lambda ytxt \text{ if } lt(wt) > 0$$

wt otherwise.

1) The proposed CSOGD algorithms.

Input: learning rate λ; bias parameter
p= npTn / nnTp for "sum" and p=cp/cn for "cost"
Initialization: w1=0.
For t=1,..T do
receive instance: xt∈ Rn;
predict: ^yt=sign(wt . Xt);
receive correct label: yt ∈ {-1,+1};
suffer loss lt(wt)=l*(wt;(Xt,yt)); * ∈ {I,II}
if (lt(wt)>0)
update classifier: wt+1 = wt- λ∇ lt(wt);
end if
end if
Output: wT+1

2) Cost-Sensitive Online Active Learning algorithm (CSOAL).

INPUT: penalty parameter C, bias parameter ρ and smooth parameter δ.
INITIALIZATION : w1 = 0.
for t = 1, . . . , T do
receive an incoming instance xt ∈ Rd;
predict label ˆyt= sign(pt), where
pt= wt · xt;
draw a Bernoulli random variable Zt∈ {0, 1} of parameter δ/(δ + |pt|);
if Zt = 1 then
query label| yt ∈ {−1,+1};
suffer loss ℓt(wt) = ℓ(wt; (xt, yt));
wt+1 = wt + τtytxt, where τt = min{C, ℓt(wt)};
else
wt+1 = wt + τtytxt, where τt = 0;
end if
end for

We refer to the above resulting cost-sensitive online classification algorithm as "CSOGD-I" for short. When using the loss function (6), the update rule can be expressed as:

$$wt+1 = \{wt + \lambda \rho tytxt \text{ if } \_t(wt) > 0 \text{ wt otherwise,}$$

where ρt = ρ ∗ I(yt=1) + I(yt=−1). We refer to the above resulting algorithm as "CSOGD-II" for short. Finally, Algorithm 1 summarizes the two proposed CSOGD algorithms. It is clear that the overall time complexity of the algorithm is O(T × n), which is linear with respect to the total number of received instances T and the dimensionality of the data n.

Remark. In Algorithm 1, one practical concern is about setting the value of ρ when the aim is to optimize the weighted sum operations. In the algorithm, ρ is formally defined as ρ = ηpTn/ ηnTp . However, one may argue the values of Tn and Tp might be unknown in a real world online classification task. To address this problem, a practical yet fairly effective approach is to estimate the ratio Tn/Tp as per to the distribution of online received training data instances over the historical order, and adaptively update this ratio during the online learning process. We will empirically observe this problem in the experimental section.

VII. THEORETICAL ANALYSIS OF COST-SENSITIVE MEASURE BOUNDS

Although the above presented algorithm is simple, very limited current study has formally observe it for online learning work. To ease our discussion, we denote by S the set of indexes that correspond to the trials when a margin error happens,

S = {t |lt(wt) > 0}. Similarly, we denote by Sp = {t |lt(wt) >0 and yt = +1},

Sn = {t |lt(wt) > 0 and yt = −1}, Sp = |Sp|, and Sn = |Sn|.

Initially, we prove the following lemma that gives the loss bound achieved by the online learning algorithm to provide subsequent theoretical analysis, which was inspired by the work in.

VIII. EXPERIMENTS

This section results to evaluate the empirical operation of the presented algorithms (CSOGD-I and CSOGD-II) for cost-sensitive online classification work. To ease our discussions, we denote by CSOC sum the proposed CSOC algorithm for increasing the weighted sum of sensitivity and specificity, and CSOCcos the proposed CSOC algorithm for diminishing the misclassification cost. The data sets and implementations of this work can be found in our project

A. Experimental Testbed and Setup

We compare our CSOGD algorithms with several stateof-the-art online learning algorithms [16], including Perceptron, "ROMMA" and its aggressive version "agg-ROMMA", and two versions of the PA algorithms [6], i.e., PA-I and PA-II. We also compare with two present cost sensitive online algorithms: prediction-based PA algorithm ('CPAPB') [6] and the perceptron algorithm with uneven margin ('PAUM'). To examine the operation, we test all the algorithms on several benchmark datasets from web machine learning repositories. For space limitation, we randomly choose a few for discussion, as listed in Table 1. All of them can be downloaded from LIBSVM website. To make a fair comparison, all algorithms adopt the same experimental setup. In particular, for all the compared algorithms, the penalty parameter C was set to 10; for the

proposed CSOCsum algorithms, we set ηp = ηn = 1/2 for all cases, while for CSOCcos, we set cp = 0.95 and cn = 0.05; for PAUM, the uneven margin was set to ρ; for PB-CPA, ρ(−1, 1)was set to 1 and ρ(1,−1) was set to ρ. The learning rate λ of CSOGD-I was set to 0.2, and the learning rate λ of CSOGD-II was set to 0.1. The value of ρ was set to cp Cn for CSOCcos and ηpTn ηnTp for CSOCsum, respectively. We also assess the parameter sensitivity about the cost-sensitive weights in our experiments. All the algorithms were used in MATLAB and run in a Windows machine with 2.33GHz. All the experiments were conducted over 20 random permutations for each dataset. The results are discribed by averaging over these 20 runs. We assess the online classification operation by several metrics: sensitivity, specificity, the weighted sum of sensitivity and specificity, and the weighted cost.

B. Evaluation of Weighted Sum Performance

We first evaluate the weighted sum operation. The first three columns of Table 2 summarize the results, and Fig. 2 shows the alteration of online average sum operation. Some results can be drawn below. First of all, by observing the sum results, we found that CSOGD always gains the best among all the datasets, which importantly outperforms all the online algorithms, including two cost-sensitive online algorithms (PAUM and CPA). This shows that it is essential to study effective cost sensitive algorithms. Second, by observing both sensitivity and specificity metrics, we found that CSOGD is not only insure to gain the best sensitivity for all cases, but also can build a fairly good specificity operation for most cases. This shows that the presented approach for CSOGD is efficient in enhancing the accuracy of predicting the examples from the rare class.

IX. Results and Discussion

Dataset:-In are present system we have taken total 700 website names in that First 500 names are normal names and remaining 200 are malicious. From the names of website we we measure the number of (.) in each website, then we find the length of the file and the date on which it was registered ,and also the name of the registrar. And we take Two classes Class(+1) is for normal website and Class(-1) for malicious website name. First we take 450 normal website names and 150 malicious file name from total 700 file names. They are cost sensitive dynamic online training of classifier. and remaining 100 are for testing.

Measures	ODLCS 1	ODLCS 2
TP	24	25
TN	39	40
FP	5	5
FN	32	30

Table 1: Evaluation of The Malicious URL Detection Performance In Terms Of The Cumulative Sum Measure

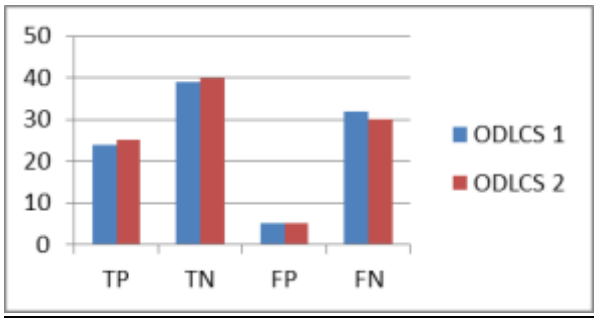

Fig. 2: Evaluation of the online cumulative average sum performance with respect to varied ratios

Measures	ODLCS 1	ODLCS 2
Accuracy	63	65
Sensitivity	79	80
Specificity	21	33
Sum	50	50
Cost	77	75

Table 2: Evaluation of the Malicious URL Detection Performance.

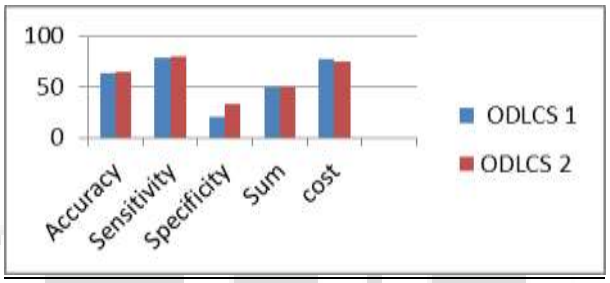

Fig. 3: Evaluation of the online cumulative average Cost performance with respect to varied ratios.

For cost we have taken CP=0.9 and CN=1
And for Sum we have taken NP=NR=0.5

X. Conclusion

In this paper we proposed as a try to build the bridge between cost-sensitive classification and online learning, this paper verified a new mechanism of Cost-Sensitive Online Classification to solve large-scale online classification work in real-world applications. We present two cost-sensitive online learning algorithms by directly escalation cost-sensitive issues depends on online gradient descent methods. We then theoretically inspect their cost-sensitive bounds, further observation their empirical performance, and finally demonstrated their applications to tackle real-world online anomaly detection tasks. This paper presents a novel mechanism of cost-sensitive online active learning (CSOAL) as a general, simple yet fairly effective approach to managing a real-world online malicious URL detection work. We proposed the CSOAL algorithms to escalate cost-sensitive issues and theoretically analyze the bounds of the implemented algorithms. We also broadly examined their empirical operation on a large-scale real-world data set.

References

[1] Jialei Wang, Peilin Zhao, and Steven C.H. Hoi, *Member, IEEE*, Cost-Sensitive Online Classification, VOL. 26, NO. 10, OCTOBER 2014

[2] Peilin Zhao, Steven C.H. Hoi School of Computer Engineering Nanyang Technological University 50 Nanyang Avenue, Singapore 639798 Cost-Sensitive Online Active Learning with Application to Malicious URL Detection August 11–14, 2013
[3] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in Proc. 15th ECML, Pisa, Italy, 2004, pp. 39–50.
[4] B. R. Bocka, "Methods for multidimensional event classification: A case study using images from a Cherenkov gamma-ray telescope," Nucl. Instrum. Meth., vol. 516, no. 2–3, pp. 511-528, 2004.
[5] G. Blanchard, G. Lee, and C. Scott, "Semi-supervised novelty detection," J. Mach. Learn. Res., vol. 11, pp. 2973–3009, Nov. 2010.
[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM CSUR, vol. 41, no. 3, Article 15, 2009.
[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, no. 1, pp. 321–357, 2002.
[8] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," J. Mach. Learn. Res., vol. 7, pp. 551–585, Mar. 2006.
[9] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence weighted learning," in Proc. NIPS, 2008, pp. 345–352.
[10] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in Proc. 5th ACM SIGKDD Int. Conf. KDD, San Diego, CA, USA, 1999, pp. 155–164.
[11] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in Proc. 25th ICML, Helsinki, Finland, 2008, pp. 264–271.
[12] C. Elkan, "The foundations of cost-sensitive learning," in Proc.17th IJCAI, San Francisco, CA, USA, 2001, pp. 973–978.
[13] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," Mach. Learn., vol. 37, no. 3, pp. 277–296, 1999.
[14] C. Gentile, "A new approximate maximal margin classification algorithm," J. Mach. Learn. Res., vol. 2, pp. 213–242, Dec. 2001.
[15] S. C. H. Hoi, R. Jin, P. Zhao, and T. Yang, "Online multiple kernel classification," Mach. Learn., vol. 90, no. 2, pp. 289–316, 2013.
[16] S. C. H. Hoi, J. Wang, and P. Zhao, "LIBOL: A library for online learning algorithms," J. Mach. Learn. Res., vol. 15, no. 1, pp. 495–499, 2014.
[17] J. Kivinen, A. J. Smola, and R. C. Williamson, "Online learning with kernels," in Proc. NIPS, 2001, pp. 785–792.
[18] Y. Li and P. M. Long, "The relaxed online maximum margin algorithm," in Proc. NIPS, 1999, pp. 498–504.
[19] Y. Li, H. Zaragoza, R. Herbrich, J. Shawe-Taylor, and J. S. Kandola, "The perceptron algorithm with uneven margins," in Proc. 19th ICML, San Francisco, CA, USA, 2002, pp. 379–386.
[20] C. X. Ling, V. S. Sheng, and Q. Yang, "Test strategies for costsensitive decision trees," IEEE Trans. Knowl. Data Eng., vol. 18, no. 8, pp. 1055–1067, Aug. 2006.
[21] H. Liu, S. Shah, and W. Jiang, "On-line outlier detection and data cleaning," Comput. Chem. Eng., vol. 28, no. 9, pp. 1635–1647, 2004.
[22] P. Zhao, S. C. H. Hoi, R. Jin, and T. Yang, "Online AUC maximization," in Proc. 28th ICML, 2011, Bellevue, WA, USA, pp. 233–240.
[23] M.-Y. Kan and H. O. N. Thi. Fast webpage classification using url features. In CIKM, pages 325–326, 2005.
[24] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In WWW, pages 83–92, Edinburgh, Scotland, 2006.
[25] 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.
[26] 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.
[27] 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.

