Online Phishing Classification Using Adversarial Data Mining and Signaling Games

Gaston L'Huillier
Universidad de Chile
Blanco Encalada 2120
Santiago, Chile
glhuilli@dcc.uchile.cl

Richard Weber
Universidad de Chile
Republica 701
Santiago, Chile
rweber@dii.uchile.cl

Nicolas Figueroa
Universidad de Chile
Republica 701
Santiago, Chile
nicolasf@dii.uchile.cl

ABSTRACT

In adversarial systems, the performance of a classifier decreases after it is deployed, as the adversary learns to defeat it. Recently, adversarial data mining was introduced, where the classification problem is viewed as a game mechanism between an adversary and an intelligent and adaptive classifier. Over the last years, phishing fraud through malicious email messages has been a serious threat that affects global security and economy, where traditional spam filtering techniques have shown to be ineffective. In this domain, using dynamic games of incomplete information, a game theoretic data mining framework is proposed in order to build an adversary-aware classifier for phishing fraud detection. To build the classifier, an online version of the Weighted Margin Support Vector Machines with a game theoretic prior knowledge function is proposed. In this paper, a new content-based feature extraction technique for phishing filtering is described. Experiments show that the proposed classifier is highly competitive compared with previously proposed online classification algorithms in this adversarial environment, and promising results were obtained using traditional machine learning techniques over extracted features.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.5.1 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*; K.4.4 [**Computers and Society**]: Electronic Commerce—*Security*

General Terms

Email Filtering, Game Theory, Data Mining

Keywords

Spam and Phishing Detection, Adversarial Classification, Games of Incomplete Information

1. INTRODUCTION

In security applications, modern threats are becoming more effective as adversaries are adapting and evolving over current security systems. In many domains, such as fraud, phishing, spam, intrusion detection, and other malicious activities, a permanent race between adversaries and classifiers. The evolution of the initial problem is driven by a rational change of the adversaries' behavior. In this context, one of the major problems of a classifier is to consider the drift concept and incremental properties of security systems. Recent studies on this topic [32], focus on the incremental characteristic of these applications, leaving the adversarial behavior as an open question in most of the previously mentioned domains.

In *Cyber-Crime*, one of the most common social engineering threats is phishing fraud. This malicious activity consists of email scams, where attackers ask for personal information to break into any site where victims store useful private information, such as financial institutions, e-commerce or massive services. The phishing filtering problem is not an easy task. While client side phishing filtering techniques have been developed by large software companies, server side filtering techniques have been a large research focus [1, 3, 5, 10]. Most of this work is based on machine learning approaches to determine the relevant features to extract from phishing emails, and data mining techniques to determine hidden patterns associated to the relationship between the extracted features.

There is an important issue when using data mining to build a classifier for the phishing detection task, and many other adversarial classification tasks: it must deal with the uncertainty of classifying malicious or regular activities, without information about the real intention of the message. This interaction can be modeled as a Bayesian game (or incomplete information game), where the classifier must choose a strategy without knowing the adversaries' real type, whether it was malicious or just happened to be a "malicious like" regular message. All this, using just the revealed set of features to decide.

The aim of this work is to present a game-theoretic data mining framework using dynamic games of incomplete information for the adversarial classification problem. A mechanism is proposed to model a signaling game between an adversary and a classifier, where equilibrium strategies and the classifier beliefs are used to build an online machine learning classifier to detect phishing emails.

Section 2 of this paper introduces previous work on adversarial data mining and latest research on phishing classifica-

tion. Problem definition and game properties are introduced in section 3. The adversary strategies definition, the classifier, and main contribution of this paper are presented in section 4, followed by the experimental settings and results in section 5. Finally, main conclusions and future work are presented in section 6.

2. PREVIOUS WORK

2.1 Adversarial Machine Learning

As described by Dalvi et al. [9], an *adversarial game* can be represented as a game between two players: A malicious agent whose adversarial activity reports its benefits, and a classifier whose main objective is to identify as many malicious activities as possible, maximizing its expected utility. The malicious agent tries to avoid detection by changing its behaviour (hence its features), inducing a high false-negative rate to the classifier. The adversary is aware that changing features to a non-adversarial behavior might not increase its benefit. Considering this, the adversary might try to maximize its benefit minimizing the cost of changing features. This framework, based on a single shot game of complete information, was initially tested in a spam detection domain [9] where the adversary-aware naïve Bayes classifier had significantly less false positives and false negatives than the classifier's plain version. Then a repeated version of the game was tested [9], where results showed that the adversary-aware classifier outperformed consistently the adversary-unaware naïve Bayes classifier.

Some extensions of the adversarial classification framework were recently developed. M. Kantarcioglu et al. [16], consider an *adversarial stackelberg game* model to define the interaction between the classifier and the adversary. They determine the subgame perfect equilibrium, reporting promising results.

Recently, several studies about the possibility that a classifier is intentionally mis-trained by the adversary or that its optimal strategies could be revealed in adaptive adversarial environments have been developed. Open questions such as "Can machine learning be secure?" are extensively discussed in [2]. More specifically, Nelson et al. present in [21] how to exploit a spam classifier to render it useless using a very specific attack framework, using indiscriminate, focused attacking and an optimal attacking function, all of them assuming that the training model used for the spam filter is based on naïve Bayes classifier.

Furthermore, Lowd and Meek proposed as the adversarial learning theory [18], which enables the adversary to reconstruct the classifier based on reasonable assumptions and reverse engineering algorithms. However, Biggio et al. present a promising alternative to randomize the classifier decision function using multi-classifier systems, in order to hide the classifier's strategy observed by the adversary, diminishing the adversarial learning and the possibilities to mis-train or learn from the classifier [6].

2.2 Phishing Classification

Spam filtering has been discussed over the last years, and many filtering techniques have been described [14]. Nevertheless, phishing classification is different in many aspects from the spam case, where most of the spam email just want to inform about some product. In phishing there is a more complex interaction between the message and the receiver, like following malicious links, filling in deceptive forms, or replying with useful information which is relevant for the message to succeed. Also, there is a clear difference among many phishing techniques, where the two main categories are known as *deceptive phishing* and *malware phishing*. While *malware phishing* has been used to spread malicious software to be installed on victim's machines, *deceptive phishing*, according to [4], can be categorized into the following six categories: *Social engineering, Mimicry, Email spoofing, URL hiding, Invisible content* and *Image content*. For each one of these subcategories, specific feature extraction techniques have been proposed [4] to help phishing classifiers to use the right characterization of their respective messages.

Among the countermeasures used against phishing, three main alternatives have been used [4]: Black listing and white listing, network and encryption based countermeasures and content based filtering. The first alternative, consists in using public lists of malicious phishing websites (the black list) and lists of legitimate non-malicious websites (white list), where each link in a message must be checked in both lists. The main problem of this countermeasure is that phishing websites do not persist long enough to be updated on-time in the black list, making difficult to keep an up-to-date list of malicious websites. The second alternative is based on email authentication methods, where the transaction time in encryption based methods could be a considerable computational cost. Besides, it is likely that a special technological infrastructure is needed for this countermeasure [4]. Previous work on content-based phishing filtering [1, 3, 4, 5, 10] focused on the extraction of a large number of features and the usage of popular machine learning techniques for classification. These approaches for automatic phishing filtering have shown promising results regarding the relative importance of features.

3. PROBLEM DEFINITION

Consider a message arriving at time t represented by the feature vector $x_t = (x_{t,1}, ..., x_{t,i}, ..., x_{t,a})$, where $x_{t,i}$ is the i^{th} feature of message x_t. Each message can belong to two classes: positive (or malicious) messages, and negative (or regular) messages. We define the adversarial classification under a dynamic game of incomplete information as a signaling game between an ADVERSARY, which attempts to defeat a CLASSIFIER by not revealing information about his real type, modifying x_i (a message of type i) into x_j (a message of type j) by using the transformation function $\phi(x_i) = x_j$.

Consider the incomplete information game, as defined by J. Harsanyi in [15], as the tuple

$$\Gamma^b = (\mathcal{N}, (A_n)_{n\in\mathcal{N}}, (T_n)_{n\in\mathcal{N}}, (p_n)_{n\in\mathcal{N}}, (U_n)_{n\in\mathcal{N}})$$

where $\mathcal{N} = \{1, ..., N\}$ is the set of players, A_n is the set of possible actions for player n, $\forall n \in \mathcal{N}$. T_n is the n^{th} player possible types set $\forall n \in \mathcal{N}$. p_n is a probability function $p_n : T_n \rightarrow [0, 1]$ which assigns a probability distribution over $\bigtimes_{j\in\mathcal{N}} T_j$ to each possible player type (T_n), $\forall n \in \mathcal{N}$. Finally, the utility function of player n is denoted by $U_n : (\bigtimes_{j\in N} A_j) \times (\bigtimes_{j\in N} T_j) \rightarrow \mathbb{R}$, which corresponds to the payoff of player n as a function over the actions of all players (A_n) and their types (t_n).

Based on the previous scheme, as described in [11, 13], dynamic games of incomplete information can be modeled as a signaling game. The model of incomplete information for the adversarial classification between an ADVERSARY ($\mathcal{A}$)

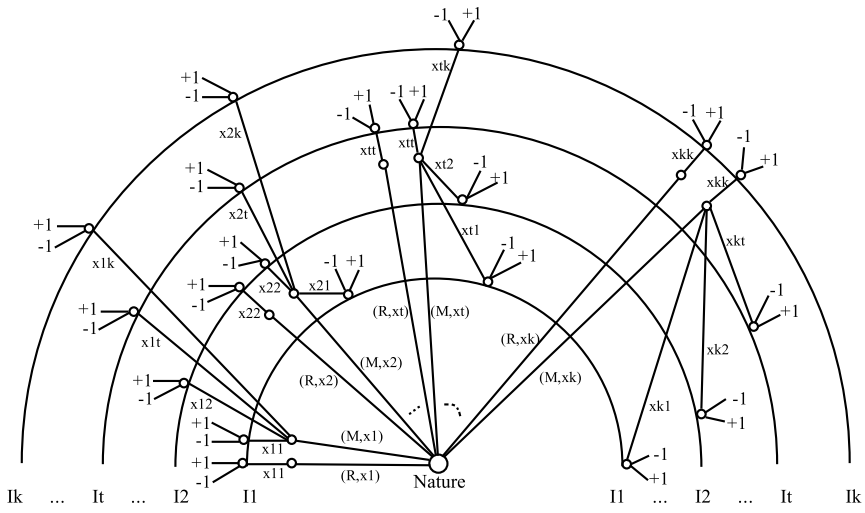

Figure 1: Extensive-form representation of the signaling game between the Classifier and the Adversary. On the figure, x_{ij} is defined by $\phi(x_i) = x_j$ and I_j is the j^{th} information set where the classifier has to decide $\mathcal{C}(x_j) = \{+1, -1\}$. All intermediate nodes between Nature and information sets, represents the strategy nodes for the adversary, where $\phi(x_i) = x_j$ is decided.

and a CLASSIFIER ($\mathcal{C}$), i.e. $\mathcal{N} = \{\mathcal{A}, \mathcal{C}\}$, behaves as the following sequence of events.

Firstly, NATURE draws a type t_i for the ADVERSARY from $T = \{t_{R,x_i}\}_{i=1}^{k} \bigcup \{t_{M,x_i}\}_{i=1}^{k}$, which states whether the adversary is Regular (R) or Malicious (M), and defines the initial optional message of type i, x_i. NATURE draws according to the probability distribution $p(t_i)$, where $p(t_i) > 0$, $\forall i$ and $\sum_{i=1}^{k} p(t_i) = 1$. Secondly, The ADVERSARY observes his type t_i, which can be either t_{R,x_i} or t_{M,x_i}, and chooses a message x_j from his set of actions $A_{\mathcal{A}} = \{\phi(x_i) = x_j\}_{j=1}^{k}$, where x_i is defined from the type t_{R,x_i} or t_{M,x_i}. The function $\phi : \mathbb{R}^a \rightarrow \mathbb{R}^a$ transforms a feature vector x_i into x_j, the message which the CLASSIFIER has to decide its class. A non malicious adversary does not have incentives to modify its behavior, so $\phi(x_i) = x_i$, when its type is t_{R,x_i}, $\forall i = \{1, ..., k\}$. Thirdly, The CLASSIFIER observes x_j (but not t_i) and chooses an action $\mathcal{C}(x_j)$ from its set of actions $A_{\mathcal{C}} = \{+1, -1\}$. It is important to notice that the CLASSIFIER is a single type player, so its type is common knowledge and there is no need to be mentioned further. Finally, payoffs are revealed by $U_{\mathcal{A}}(t_i, \phi(x_i), \mathcal{C}(\phi(x_i)))$ and $U_{\mathcal{C}}(t_i, \phi(x_i), \mathcal{C}(\phi(x_i)))$.

The extensive form game that represents the signaling game between the ADVERSARY and CLASSIFIER is presented in figure 1.

In order to analyze the optimal strategies for the CLASSIFIER in the proposed mechanism, special requirements and assumptions over the traditional Bayesian Nash equilibrium must be considered [13, 17].

DEFINITION 1. ***Signaling requirement 1 (S1)*** *After observing any message x_j, from $A_{\mathcal{A}}$, the* CLASSIFIER *must have a belief about which types could have sent x_j. Denote this belief by the probability distribution $\mu(t_i|x_j)$, where $\mu(t_i|x_j) \geq 0$, $\forall t_i \in T$ and $\sum_{t_i \in T} \mu(t_i|x_j) = 1$.*

DEFINITION 2. ***Signaling requirement 2 (S2C)*** *For each $x_j \in A_{\mathcal{A}}$, the* CLASSIFIER*'s optimal strategy defined as the probability distribution σ_C^* over the* CLASSIFIER*'s actions $\mathcal{C}(x_j) \in A_{\mathcal{C}}$, must maximize the* CLASSIFIER*'s expected utility, given the beliefs $\mu(t_i|x_j)$ about which types could have sent x_j. That is,*

$$\forall x_j, \sigma_{\mathcal{C}}^*(\cdot|x_j) \in arg\max_{\alpha_{\mathcal{C}}} \sum_{ti \in T} \mu(t_i|x_j) \cdot U_C(t_i, x_j, \alpha_{\mathcal{C}}) \quad (1)$$

where

$$U_C(t_i, x_j, \sigma(\cdot|x_j)) = \sum_{\mathcal{C}(x_j) \in A_{\mathcal{C}}} \sigma_{\mathcal{C}}(\mathcal{C}(x_j)|x_j) U_{\mathcal{C}}(t_i, x_j, \mathcal{C}(x_j)) \quad (2)$$

DEFINITION 3. ***Signaling requirement 3 (S2A)*** *For each $t_i \in T$, the* ADVERSARY*'s optimal message $x_j = \phi(x_i)$, defined by the probability distribution $\sigma_{\mathcal{A}}^*$ over the* ADVERSARY*'s actions $x_j \in A_{\mathcal{A}}$, must maximize the* ADVERSARY*'s utility function, given the* CLASSIFIER*'s strategy $\sigma_{\mathcal{C}}^*$. That is,*

$$\forall t_i, \sigma_{\mathcal{A}}(\cdot|t_i) \in arg\max_{\alpha_{\mathcal{A}}} U_{\mathcal{A}}(t_i, \alpha_{\mathcal{A}}, \sigma_{\mathcal{C}}^*) \quad (3)$$

where

$$U_{\mathcal{A}}(t_i, \sigma_{\mathcal{A}}, \sigma_{\mathcal{C}}) = \sum_{x_j \in A_{\mathcal{A}}} \sigma_{\mathcal{A}}(x_j|t_i) U_{\mathcal{A}}(t_i, x_j, \sigma_C(\cdot|x_j))$$

and

$$U_{\mathcal{A}}(t_i, x_j, \sigma_C(\cdot|x_j)) = \sum_{\mathcal{C}(x_j) \in A_{\mathcal{C}}} \sigma_{\mathcal{C}}(\mathcal{C}(x_j)|x_j) U_{\mathcal{A}}(t_i, x_j, \mathcal{C}(x_j))$$

DEFINITION 4. ***Signaling requirement 4 (S3)*** *For each $x_j \in A_{\mathcal{A}}$, if there exists $t_i \in T$ such that $\sigma_{\mathcal{A}}^*$, then the* CLASSIFIER*'s belief at the information set I_j corresponding to x_j must follow from Bayes' rule and the* ADVERSARY*'s strategy*

$$\mu(t_i|x_j) = \frac{\sigma_{\mathcal{A}}^*(x_j|t_i) \cdot p(t_i)}{\sum_{t_r \in T} \sigma_{\mathcal{A}}^*(x_j|t_r) \cdot p(t_r)} \quad (4)$$

If $\sum_{t_r \in T} \sigma_{\mathcal{A}}^*(x_j|t_r) \cdot p(t_r) = 0$, $\mu(t_i|x_j)$ can be defined as any probability distribution.

Sequential equilibria, a subset of perfect Bayesian equilibrium (PBE) in the adversarial signaling game is a pair of mixed strategies $\sigma^*_{\mathcal{A}}$ and $\sigma^*_{\mathcal{C}}$ and a belief $\mu(t_i|x_j)$ satisfying signaling requirements *S1, S2C, S2A*, and *S3*. It is clear, by construction of the mechanism, that requirements *S1* and *S3* are satisfied by the adversarial classification game. However, signaling requirement *S2A* will be considered satisfied as a first approach and a strong assumption on the game development. Whether adversarial behavior strategies, as described by [9], could represent a more reliable interaction will be considered as an open question to be treated as future work.

Recently, numerical approximation on the sequential equilibria refinement have been proposed by Turocy in [25], using a transformation of the logit quantal response equilibrium (QRE) correspondence, parameterized by a scalar precision parameter, which as tends to infinity, a numerical approximation for the sequential equilibria is obtained. This numerical algorithm has been implemented in Gambit [19], an open-source project for estimating equilibrium results in finite games.

4. STRATEGIES, TYPES AND CLASSIFIER MECHANISM

In this section the main characterization of the signaling game proposed and contribution of this work is presented. Firstly, the ADVERSARY's strategies and types are determined by the usage of unsupervised learning tecniques. Then, the classifier strategy represented by a novel data mining algorithm which includes game-theoretic parameters is extensively developed.

4.1 Phishing Features, Strategies and Types Extraction

4.1.1 Corpus Description

The previously defined classifier was tested over an English language *phishing* and *Ham* email corpus built using Jose Nazario's *phishing* corpus [20] and the Spamassassin *Ham* collection. The *phishing* corpus[1] consists of 4450 emails manually retrieved from November 27, 2004 to August 7, 2007. The Spamassassin collection, from the Apache SpamAssassin Project[2], is based on a collection of 6951 *Ham* email messages. The email collection was saved in a `unix mbox` email format, and was processed using Perl scripts.

4.1.2 Basic Features

As initially described in [10] and then in [3, 4, 5], the extraction of basic content-based features is needed for a minimum representation of phishing emails. These features, considered as binary variables for this study, are associated to structural properties of the email, link analysis, programming elements and the output of the spam filters. It is important to notice that basic features (a total of 15 features) are directly extracted from content-based properties of an email message, and each one can be considered as a strategy for the ADVERSARY to defeat the CLASSIFIER.

4.1.3 Word List and Clustering Features

Previously mentioned features are not sufficient for the appropriate characterization of a phishing message, and clearly not the complete representation of adversarial strategies. Following the content-based extraction techniques, a new list of features is proposed to characterize phishing emails, which is related to the ADVERSARY's strategy $A_{\mathcal{A}}$.

In the following, word-based features will be described as an approach to fulfill the needed phishing strategies' representation. These features will be presented as a binary variable for each word in a list of keywords, whose value is 1 if the word is used in the document, and 0 otherwise. The main idea is that phishing strategies are defined as a list of words used in a message. So, for each keyword cluster (ADVERSARY type), a list of relevant words will be associated, representing a phishing strategy.

First, a stop-words removal and stemming pre-processing is necessary to setup the email database. Let R be the total number of different words in the complete collection of phishing emails, and Q the total number of emails. A vectorial representation of a the phishing corpus is given by $M = (m_{ij}), i = 1, ..., R$ and $j = 1, ..., Q$, where m_{ij} is the weight word i in a document. The weights m_{ij} considered in this research are an improvement of the basic *tf-idf* term [27, 28] (*Term Frequency times inverse document frequency*), and are defined by

$$m_{ij} = f_{ij}(1 + sw(i)) \times log\left(\frac{Q}{n_i}\right) \tag{5}$$

where f_{ij} is the frequency of the i^{th} word in the j^{th} document, $sw(i)$ is a factor of relevance associated to word i in a set of words and n_i is the number of documents containing word i. On this case, $sw(i) = \frac{w^i_{\text{email}}}{TE}$, where w^i_{email} is the frequency of word i over all documents, and TE is the total amount of emails.

The *tf-idf* term is a weighted representation of the importance of a given word, in a document that belongs to a collection of documents. The *term frequency* indicates the weight of each word in a document, while the *inverse document frequency* states whether the word is frequent or uncommon in the document, setting a lower or higher weight respectively.

Based on the previous *tf-idf* representation, a clustering technique must be considered for the segmentation of the whole collection of phishing emails. k-Means clustering with the cosine between documents, as the distance function was used. Furthermore, the optimal number of clusters was determined using as stopping rules the minimization of the distance within every cluster and the maximization of the distance between clusters. Then, for each cluster the most relevant words are determined by

$$Cw(i) = \sqrt[|\zeta|]{\prod_{p \in \zeta} m_{ip}} \tag{6}$$

for $i \in 1, .., R$, where Cw is a vector containing the geometric mean of each word's weights within the messages contained in a given cluster. Here, ζ is the set of documents in each cluster and m_{ip} as defined in equation 5. Finally, the most important words for each cluster can be determined ordering the weights of vector Cw. This procedure is based on previous work described in [29]. Results on this method showed that the optimal number of clusters is 13, where the 30 most relevant words of each cluster were considered as features (a

[1] Available at http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus
[2] Available at http://spamassassin.apache.org/publiccorpus/

Table 1: Five most relevant words for each of the 13 clusters of the phishing corpus.

Cluster	Word 1	Word 2	Word 3	Word 4	Word 5
1	limit	use	credit	card	provid
2	address	follow	bill	communiti	violat
3	ebay	secur	bank	access	user
4	chase	repli	payment	answer	info
5	vector	area	desktop	loan	keybank
6	account	paypal	messag	inform	updat
7	signin	list	partner	site	offer
8	amazon	union	never	maintain	world
9	ebay	email	page	polici	help
10	login	respons	verif	window	yahoo
11	area	demo	hidden	expens	image
12	use	sidebar	card	repli	review
13	union	nation	answer	googl	barclay

total of 390). The first five relevant words of each cluster are presented in table 1.

4.1.4 Strategies and Types Extraction

Based on previously mentioned features (a total of 405 features), a feature selection algorithm is used to improve the performance of the classification algorithms, eliminating noisy features that do not represent the target value, and do not provide enough information about the underlying phenomenon observed by the game agents. This is a key step for eliminating word features considered arbitrarily as the 30 most relevant words for each cluster, giving a final list of attributes for the phishing/ham classification problem. These attributes represent the strategy profile for a given ADVERSARY. An information-theoretic feature selection algorithm was implemented, where the information gain for each feature was calculated over the whole database, eliminating those features that did not report a minimum threshold. 153 features where eliminated, obtaining the final set of 252 features.

The ADVERSARY's types $t_i \in T$ are extracted using k-Means clustering over the collection of emails (phishing and ham). Therefore, the number of clusters over the whole set of features (K_{features}) will represent the total number of types for the ADVERSARY player. For each message x_j, represented by a vector of 252 variables, the type will be determined by

$$t_i = \arg\min_i d(x_j, C_i), \forall i = \{1, .., K_{\text{features}}\} \tag{7}$$

where C_i is the centroid of cluster i, and function $d : \mathbb{R}^a \times \mathbb{R}^a \rightarrow \mathbb{R}$ represents the distance between two vectors of dimension a. The distance function used in this research is the Hamming distance, represented by the number of bits needed to change one vector into another.

4.2 Classifier Strategy

As mentioned before, the CLASSIFIER's optimal strategies are defined by the set $A_{\mathcal{C}} = \{+1, -1\}$. From the signaling requirement *S2C*, it can be shown that the CLASSIFIER's optimal strategy $\mathcal{C}^*(x_j)$ can be solved by the following conditional statement,

$$\mathcal{C}^*(x_j) = \begin{cases} +1 & \text{if condition 9 is satisfied} \\ -1 & \textit{Otherwise} \end{cases} \tag{8}$$

$$\sum_{t_i \in T_{\mathcal{M}}} \mu(t_i, x_j) \Delta U^{t_i}_{\mathcal{C},M}(x_j) > \sum_{t_i \in T_{\mathcal{R}}} \mu(t_i, x_j) \Delta U^{t_i}_{\mathcal{C},R}(x_j) \tag{9}$$

Where $T_{\mathcal{M}} = \{t_{M,x_i}\}_{i=1}^k$, $T_{\mathcal{R}} = \{t_{R,x_i}\}_{i=1}^k$, $\mu(t_i|x_j)$ is defined by equation 4,

$$\begin{aligned} \Delta U^{t_i}_{\mathcal{C},R}(x_j) &= \sigma^*_{\mathcal{C}}(-1|x_j) U_{\mathcal{C}}(t_{R,x_i}, x_j, -1) \\ &\quad - \sigma^*_{\mathcal{C}}(+1|x_j) U_{\mathcal{C}}(t_{R,x_i}, x_j, +1) \end{aligned}$$

and

$$\begin{aligned} \Delta U^{t_i}_{\mathcal{C},M}(x_j) &= \sigma^*_{\mathcal{C}}(+1|x_j) U_{\mathcal{C}}(t_{M,x_i}, x_j, +1) \\ &\quad - \sigma^*_{\mathcal{C}}(-1|x_j) U_{\mathcal{C}}(t_{M,x_i}, x_j, -1) \end{aligned}$$

In the following, these expressions will be considered as

$$\Delta U^{t_i}_{\mathcal{C},M}(x_j) = \cdot(\sigma^*_{\mathcal{C}}(+1|x_j) \cdot \epsilon_M + \sigma^*_{\mathcal{C}}(-1|x_j) \cdot \gamma_M) \cdot (w^T \cdot x_j + b)$$

and

$$\Delta U^{t_i}_{\mathcal{C},R}(x_j) = \cdot(\sigma^*_{\mathcal{C}}(-1|x_j) \cdot \epsilon_R + \sigma^*_{\mathcal{C}}(+1|x_j) \cdot \gamma_R) \cdot (w^T \cdot (e - x_j) + b)$$

where γ_M. γ_R, ϵ_M and ϵ_R must be defined based on microeconomic assumptions on the primitives of the game, and e is a vector of ones, whose dimension is a. The modeling intuition and the final analytical expression of the utility functions are intentionally omitted in this paper.

The previous game-theoretic result (condition 9), can be considered as a prior knowledge constraint in a classification problem, associated with the regularized risk minimization from the statistical learning theory proposed by Vapnik in [26]. All this, is formulated as the following quadratic problem,

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \sum_{i=1}^{a} w_i^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t} \quad & y_i \left(w^T \cdot x_i + b \right) \cdot \Psi(x_i) \geq (1 - \xi_i) \\ & \forall i \in \{1, .., N\} \\ & \xi_i \geq 0 \ \forall i \in \{1, .., N\} \end{aligned} \tag{10}$$

Where,

$$\Psi(x_i) = \frac{1 + \psi(x_i)}{\sum_{k=1}^{a} w_k + 2 \cdot b}$$

and

$$\psi(x_i) = \frac{\sum_{t_r \in T_{\mathcal{M}}} \mu(t_r|x_i) \cdot (\epsilon_M \cdot \sigma^*_{\mathcal{C}}(+1|x_i) + \gamma_M \cdot \sigma^*_{\mathcal{C}}(-1|x_i))}{\sum_{t_r \in T_{\mathcal{R}}} \mu(t_r|x_i) \cdot (\epsilon_R \cdot \sigma^*_{\mathcal{C}}(-1|x_i) + \gamma_R \cdot \sigma^*_{\mathcal{C}}(+1|x_i))}$$

The online algorithm to solve the proposed minimization problem, is based on solving its dual formulation using the Sequential Minimal Optimization (SMO) described by Platt in [22]. The SMO algorithm is used to train SVMs breaking up the large Quadratic Programming (QP) representation of the dual into small series of QP problems, which are solved analytically by the algorithm. Small changes in the SMO algorithm, such as explained in previous prior knowledge inclusion in SVMs [31] were considered. Based on previous work on Online Support Vector Machines algorithms described by Gentile in [12] and later by Sculley in [23], the proposed adversary-aware classifier is stated as follows,

Algorithm 4.1: Bayesian Adversary-Aware Online SVM

Data: $(x_1, y_1), ..., (x_n, y_n), \gamma_M, \gamma_R, \epsilon_M, \epsilon_R, m, \tau, Gp, C$
Result: $f(x_t) = w^T \cdot x_t + b_t$

```
Initialize w_0 := 0, b_0 := 0, seenData := {};
foreach x_t, y_t do
    Classify x_t using f(x_t) = w_{t-1}^T · x_t + b_{t-1};
    if y_t (w_{t-1}^T · x_t + b_{t-1}) Ψ(x_t) < τ then
        Find w', b' with prior knowledge SMO with
        parameter C on seenData, with w_{t-1} and b_{t-1}
        as seed hypothesis, and Ψ(x_t);
        set w_t := w' and b_t := b';
    if size(seenData) > m then
        remove oldest example from seenData;
    if T mod Gp = 1 then
        Approximate sequential equilibrium strategies
        using logit QRE;
    add x_t to seenData;
    update p(t_i) based on observed messages on
    seenData;
    update beliefs μ(t_i|x), ∀t_i ∈ T, x ∈ seenData using
    signaling requirement S3;
    update Ψ(x_i), ∀i ∈ seenData;
return 1 ;
```

Previous algorithm 4.1 presents the online learning algorithm, *Bayesian Adversary-Aware Online* SVM (BAAO-SVM). Based on the CLASSIFIER's beliefs and sequential equilibrium strategies, the hyperplane parameters are updated, incorporating as prior knowledge constraints the game theoretic results. The main idea of the algorithm, is that given an incoming message x_t, a label is assign using the classification function $f(x_t) = w_{t-1}^T \cdot x_t + b_{t-1}$. If the CLASSIFIER's optimal strategy is not satisfied (equation 9), the hyperplane parameters are updated using a modified version of the SMO algorithm over the seen messages (seenData set). A memory parameter m is used to set the number of messages in seenData. Then, every Gp periods, the sequential equilibrium strategies are updated using logit QRE. Finally, x_t is added to seenData and the type's probabilities are updated, hence beliefs and $\Psi(x_i) \forall i \in$ seenData. At $t = 0$, $\Psi(x_i)$ is initialized with all mixed strategies set to $\frac{1}{2}$, as with no prior information, all outcomes can be considered equally likelly to happen. It is important to notice that the algorithm evolves dynamically as messages are presented to the CLASSIFIER.

5. EXPERIMENTAL SETTINGS AND RESULTS

In this section, the experimental settings for batch and online learning performance evaluation, as well as the evaluation criteria is presented.

5.1 Experiments

The classification of phishing emails is a natural extension of text mining, where the most promising classification algorithms are Support Vector Machines, naïve Bayes, Random Forest, among other text categorization algorithms [24]. In the online setting, the problem associated to the email inbox nature, where messages arrive from an undetermined set of messages. In this context, the following experiments will be determined to give the right benchmark results for the proposed feature extraction between previous results and batch learning SVMs. Likewise, the main objective of the experimental setting is to show the accuracy and effectiveness between different online classification algorithms and BAAO-SVM, the proposed online adversary aware classifier.

Firstly, a 10 times 10 cross validation learning schema using SVM on the complete database characterized with 265 features was developed, using the libSVM-library [7], and the same learning schema was used to train a naïve Bayes model implemented in Weka [30]. Then, for the online setting, the Relaxed Online SVM (ROSVM) proposed by Sculley in [23] was used, as well as an incremental evaluation of naïve Bayes, and BAAO-SVM were evaluated in this schema.

The adversary aware classifier was developed using the 265 features as possible ADVERSARY's strategies, and the $\{+1, -1\}$ set as the CLASSIFIER's strategies. Types where considered as previously described type extraction method, where a total of 7 clusters were obtained. Approximation on the sequential equilibria was determined using logit QRE, implemented in Gambit [19] software command-line tool (`gambit-logit`). The CLASSIFIER's strategy (adversary-aware classifier) described in section was implemented in C++, extending D. Sculley's Online SVM implementation [23], with a modified version of SMO for prior knowledge described in [31]. BAAO-SVM parameters tunning were estimated over a 20% subset from the overall dataset, setting $m = 100$ for the time window, $\tau = 0.6$ for the threshold, $G_p = 250$ for the game period, and $C = 100$ for the SVM objective function.

The values of γ_M. γ_R, ϵ_R and ϵ_M where defined as an initial estimation over the primitives of the game. More details on this model parameters finding where intentionally omitted by the authors.

5.1.1 Evaluation Criteria

The resulting confusion matrix can be described using four possible outcomes: Correctly classified phishing messages or True Positives (TP), correctly classified ham messages or True Negative (TN), wrong classified ham messages as phishing or False Positive (FP) and wrong classified phishing messages as ham or False Negative (TN). The evaluation criteria considered are: The False Positive Rate (FP-Rate) and the False Negative Rate (FN-Rate) as the proportion of wrongly classified ham and phishing email messages respectively. Precision, as the classifier's safety, states the degree in which messages identified as phishing are indeed malicious. Recall, as the classifier's effectiveness, states the percentage of phishing messages that the classifier manages to classify correctly. F-measure, the harmonic mean between the Precision and Recall, and Accuracy, the overall percentage of correct classified email messages.

5.2 Results

5.2.1 Batch Learning Performance

As shown in Table 2, the F-measure obtained for a 10 times 10 fold cross-validation SVM is 99.32% and for the naïve Bayes algorithm under the same learning schema the F-measure obtained is 94.84%. Previous results for the same email corpus reported an F-measure of 99.89% obtained by Bergholz et al. in [5]. In some evaluating measures, these results are slightly worst than previously obtained results,

Table 2: Experimental results for the benchmark machine learning algorithms in the Batch Learning context.

Model	FP-Rate	FN-Rate	Accuracy
Bergholz's SVM	0.07%	1.11%	**99,52%**
10x10xv SVM	1.21%	0.33%	**99.48%**
Naïve Bayes	4.47%	6.60%	94.31%
Model	Precision	Recall	F-measure
Bergholz's SVM	99.89%	99.89%	**99.89%**
10x10xv SVM	99.67%	98.97%	**99.32%**
Naïve Bayes	93.35%	96.38%	94.84%

Table 3: Experimental results for the benchmark machine learning algorithms in the online learning context.

Model	FP-Rate	FN-Rate	Accuracy
Inc. Naïve Bayes	1.33%	25.66%	81.18%
ROSVM	15.45%	14.26%	**85.20%**
BAAO-SVM	14.69%	12.26%	**86.63%**
Model	Precision	Recall	F-measure
Inc. Naïve Bayes	99.78%	74.34%	85.20%
ROSVM	85.20%	86.83%	**86.01%**
BAAO-SVM	87.64%	87.74%	**87.69%**

but are highly competitive. This points out an interesting open question: as a future work, a combined feature extraction technique could achieve better results. However, results for the False Positive Rate is considerable better than previously obtained with a value of 0.33%, compared to 1.11% respectively.

5.2.2 *Online Algorithms Performance*

To identify the online property of learning algorithms is not an easy task. In this work, a first approach using previously mentioned classification performance measures in section 5.1.1, the applicability and accuracy of the overall proposed algorithm were tested. Here, as shown in table 3, ROSVM obtained an F-measure of 86.01% with an accuracy of 85.20%, for an online version of naïve Bayes the F-measure is 85.20% whose accuracy is 81.18% and for the proposed adversary aware classifier (BAAO-SVM) the F-measure is 87.69% whose accuracy is 86.63%, with a better performance than previously used online classification algorithms on these evaluating criteria.

6. CONCLUSIONS AND FUTURE WORK

An extension of the Adversarial Classification framework for Adversarial Data Mining was presented, considering dynamic games of incomplete information as a new approach to make classifiers improve their performance in adversarial environments. This work considered strong assumptions on the ADVERSARY strategies, the utility function modeling for the CLASSIFIER, and experimental setups related to the database processing.

The proposed adversary-aware classifier, BAAO-SVM, whose core is mainly the Support Vector Machines model, considers a signaling game where beliefs, mixed strategies and probabilities for the messages' types are updated and incorporated as prior knowledge, as new email messages arrives. This enables the classifier to change the margin error parameter dynamically as the game evolves, considering an embedded awareness of the adversarial environment. More specifically, this is considered in the miss-classification constraint in the optimization problem for the SVM algorithm. As a first approach, the experimental settings showed promising results over previous online text categorization algorithms used for email filtering.

Feature extraction is a key component for the game strategies and types for the game proposed. Results showed that the proposed strategies used as features are highly competitive in comparison with previous feature extraction work in phishing filtering. Future work could be oriented to consider a mixture of former and present feature extraction techniques. This could estimate a better strategy space for the ADVERSARY, therefore improving the ADVERSARY types. This is an important topic that affects directly the definition of the signaling game, hence the CLASSIFIER's performance.

Determining the actual drift concept of the game, e.g. ADVERSARY learning new phishing strategies, is an important open question. An experimental setup to show the impact on the classifier's performance related to the inclusion of new ADVERSARY strategies within an already defined set of strategies (features) might help to answer this question in future work.

In game modeling, adversaries must be considered as strategic agents. For this, their strategies could be estimated using linear programming, as previous authors recommended in the original Adversarial Classification framework [9]. However, this first approach in adversarial classification with dynamic games of incomplete information showed interesting empirical and theoretical results. An extension on theoretical aspects of the game theoretical framework, such as refinements on these equilibria, using for example the intuitive criteria proposed by Cho and Kreps [8], among other special refinements for the perfect Bayesian [11] equilibria could be considered.

7. ACKNOWLEDGMENTS

Support from the Millennium Science Institute on Complex Engineering Systems (www.sistemasdeingenieria.cl) and the Center for Analysis and Modeling for Security (www.ceamos.cl) is greatly acknowledged.

8. REFERENCES

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair. A comparison of machine learning techniques for phishing detection. In *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69, New York, NY, USA, 2007. ACM.

[2] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, and J. D. Tygar. Can machine learning be secure? In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, New York, NY, USA, 2006. ACM.

[3] R. Basne, S. Mukkamala, and A. H. Sung. *Detection of Phishing Attacks: A Machine Learning Approach*, chapter Studies in Fuzziness and Soft Computing, pages 373–383. Springer Berlin / Heidelberg, 2008.

[4] A. Bergholz, J. D. Beer, S. Glahn, M.-F. Moens, G. Paass, and S. Strobel. New filtering approaches for phishing email. *Journal of Computer Security*, 2009. Accepted for publication.

[5] A. Bergholz, J.-H. Chang, G. Paass, F. Reichartz, and S. Strobel. Improved phishing detection using model-based features. In *Fifth Conference on Email and Anti-Spam, CEAS 2008*, 2008.

[6] B. Biggio, G. Fumera, and F. Roli. Multiple classifier systems for adversarial classification tasks. In J. A. Benediktsson, J. Kittler, and F. Roli, editors, *MCS*, volume 5519 of *Lecture Notes in Computer Science*, pages 132–141. Springer, 2009.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[8] I.-K. Cho and D. M. Kreps. Signaling games and stable equilibria. *The Quarterly Journal of Economics*, 102(2):179–221, May 1987.

[9] N. Dalvi, P. Domingos, M. Sumit, and S. DeepakVerma. Adversarial classification. In *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining*, volume 1, pages 99–108, Seattle, WA, USA, 2004. ACM Press.

[10] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 649–656, New York, NY, USA, 2007. ACM.

[11] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, October 1991.

[12] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, Vol. 2:213–242, December 2001.

[13] R. Gibbons. *Game Theory for Applied Economists*. Princeton University Press, 1992.

[14] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, Vol. 50(2):24–33, 2007.

[15] J. C. Harsanyi. Games with incomplete information played by bayesian players. the basic probability distribution of the game. *Management Science*, 14(7):486–502, 1968.

[16] M. Kantarcioglu, B. Xi, and C. Clifton. A game theoretic framework for adversarial learning. In *CERIAS 9th Annual Information Security Symposium*, 2008.

[17] D. M. Kreps and R. Wilson. Sequential equilibria. *Econometrica*, 50(4):863–94, July 1982.

[18] D. Lowd and C. Meek. Adversarial learning. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 641–647, New York, NY, USA, 2005. ACM.

[19] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 0.2007.01.30, 2007.

[20] J. Nazario. Phishing corpus, 2004-2007.

[21] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia. Exploiting machine learning to subvert your spam filter. In *LEET'08: Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 1–9, Berkeley, CA, USA, 2008. USENIX Association.

[22] J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[23] D. Sculley and G. M. Wachman. Relaxed online svms for spam filtering. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 415–422, New York, NY, USA, 2007. ACM.

[24] F. Sebastiani. Text categorization. In A. Zanasi, editor, *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*, pages 109–129. WIT Press, Southampton, UK, 2005.

[25] T. L. Turocy. Using quantal reponse to compute nash and sequential equilibria. *Economic Theory*, Vol. 42, Issue 1, 2010.

[26] V. N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, 1999.

[27] J. Velasquez, H. Yasuda, T. Aoki, and R. Weber. A new similarity measure to understand visitor behavior in a web site. *IEICE Transactions on Information and Systems, Special Issues in Information Processing Technology for web utilization*, vE87-D i2.:389–396, 2004.

[28] J. D. Velasquez and V. Palade. *Adaptive Web Sites: A Knowledge Extraction from Web Data Approach*. IOS Press, 2008.

[29] J. D. Velasquez, S. A. Rios, A. Bassi, H. Yasuda, and T. Aoki. Towards the identification of keywords in the web site text content: A methodological approach. *International Journal of Web Information Systems information*, Vol. 1(1):pp. 53–57, 2005.

[30] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[31] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 326–333, New York, NY, USA, 2004. ACM.

[32] P. Zhang, X. Zhu, and Y. Shi. Categorizing and mining concept drifting data streams. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 812–820, New York, NY, USA, 2008. ACM.