Data Min Knowl Disc (2017) 31:92–133
DOI 10.1007/s10618-016-0459-9

Adversarial classification using signaling games with an application to phishing detection

Nicolas Figueroa[1] · **Gastón L'Huillier**[2] · **Richard Weber**[3]

Received: 7 February 2013 / Accepted: 7 March 2016 / Published online: 22 March 2016

Abstract In adversarial classification, the interaction between classifiers and adversaries can be modeled as a game between two players. It is natural to model this interaction as a dynamic game of incomplete information, since the classifier does not know the exact intentions of the different types of adversaries (senders). For these games, equilibrium strategies can be approximated and used as input for classification models. In this paper we show how to model such interactions between players, as well as give directions on how to approximate their mixed strategies. We propose perceptron-like machine learning approximations as well as novel Adversary-Aware Online Support Vector Machines. Results in a real-world adversarial environment show that our approach is competitive with benchmark online learning algorithms, and provides important insights into the complex relations among players.

Responsible editor: Bing Liu.

✉ Richard Weber
rweber@dii.uchile.cl

Nicolas Figueroa
nicolasf@uc.cl

Gastón L'Huillier
g@sudoai.com

1 Institute of Economics, Pontificia Universidad Católica de Chile, Santiago, Chile

2 Sudo Technologies, Inc, 530 Oak Grove Ave, Menlo Park, CA 94025, USA

3 Department of Industrial Engineering, FCFM, Universidad de Chile, Av. Beauchef 851, Santiago, Chile

Keywords Adversarial classification · Signaling games · Perfect Bayesian equilibrium · Incremental learning · Support vector machines · Cybercrime · Phishing filtering

1 Introduction

Adversarial classification goes beyond traditional classification in regard to at least two issues. On the one hand, classifiers have to deal with adversaries who change their behavior intentionally in order to fool the classifier. On the other hand, classifiers do not have complete information on the adversaries' real intentions, which can be malicious or regular, and can be inferred only from their actions. This issue becomes even more complex since in many applications classifiers have to take the existence of many regular, i.e., non-malicious actions from users without malicious intentions, into account. Examples are the detection of fraudulent activities in credit card transactions, the identification of loan borrowers who do not intend to pay back the credit they get (Bravo et al. 2015), or the identification of malicious messages, such as spam or phishing emails. In this paper the latter example will be analyzed in more extensive detail.

In Game Theory, signaling games have been used to model this kind of situation, in which *privately informed* agents make decisions, and another agent infers their real intentions as well as possible, and then chooses an action. Given these actions every agent obtains a payoff. These models have been successful at explaining seemingly irrational actions by strategic agents, such as a monopolist setting prices lower than optimal, or companies engaging in uninformative advertising. The common strategic concept is that the informed agent chooses actions not only for immediate profit, but also to influence the *beliefs* of other agents.

This framework suits our analysis of adversarial classification very well, as applied, for example, to phishing detection. Here, senders are privately informed agents (only they, themselves, know their real intentions and the message they want to send), who choose an action: their message. In the next step, the classifier attempts to infer the real nature of their intentions, based on the actions chosen by senders. The classifier's objective is to block malicious messages, using its respective beliefs about their sender.

In this paper we use the standard signaling framework developed in Game Theory, combined with classification techniques from Data Mining. This combination allows the development of novel adversarial classification models, which will be applied to classifying phishing emails.

The main contribution of this paper is a combination of two disciplines (Game Theory and Data Mining) within a dynamic classification scheme. We develop a generic model that combines the strategic interaction between agents (a signaling game) with sophisticated feature selection techniques and Support Vector Machines (SVM). This model has been implemented and applied to the specific case of phishing email detection in which the proposed approach achieves competitive results regarding classification accuracy. Furthermore, and in addition to competing approaches, it provides interesting insights into the interaction among agents.

Section 2 of this paper introduces the general framework for adversarial classification, as well as extensions, and recent work in this area. The theoretical backgrounds

of game properties, learning strategies, and game characterization are presented in Sect. 3. In Sect. 4, we exhibit classifiers based only on the game-theoretic concept of quantal response equilibria (QRE) and show how the phenomenon of dynamic interactions between classifier and adversary can be modeled. Two classification algorithms in adversarial environments, with static (S-QRE) and perceptron-like (P-QRE) interactions among the agents, are introduced. The approach of sequential equilibria is combined with state-of-the-art Data Mining techniques in Sect. 5, where novel adversary-aware classifiers using game-theoretical components are developed. An application of the proposed framework, as well as the experimental setup, evaluation criteria, and results, is presented in Sect. 6.

Finally, the main conclusions and suggestions for future work are outlined in Sect. 7. Technical details, such as the techniques employed for feature extraction and a list of symbols, are provided in Appendices 1 and 2, respectively.

2 Adversarial classification

In this section we present first the general adversarial classification framework. Subsequently, different extensions, as well as recent developments of such a framework, are overviewed.

2.1 Adversarial classification framework

As proposed by Dalvi et al. (2004), an *Adversarial Game* can be represented as a game between two players: A malicious agent (ADVERSARY) whose activity reports benefits, and a CLASSIFIER whose main objective is to identify as many malicious activities as possible, maximizing an expected utility. The ADVERSARY tries to avoid detection by changing his/her behavior which leads to modified feature values describing those activities. This also increases incorrectly identified regular messages, inducing a high false-positive rate to the CLASSIFIER. The malicious agent is aware that changing to non-adversarial behavior will not increase his/her benefit.

In a classification framework the activities (messages) are the objects to be classified. Each object is described by an A-dimensional feature vector $\mathbf{x} = \{x_1, \ldots, x_A\}$, where x_a is the value of the ath feature, $a \in \{1, \ldots, A\}$. Let y be the dependent variable, where $y = +1$ represents a malicious instance, and $y = -1$ represents a regular instance.[1]

Let $\mathcal{F}_{x_a}$ be the set of possible values for x_a and $\mathcal{F}_y = \{-1, +1\}$ the set of possible values for y. $\mathcal{F}_{\mathbf{x}} := \times_{a\in\{1,\ldots,A\}} \mathcal{F}_{x_a}$.

An instance i is described by its feature vector and its dependent variable $(\mathbf{x}^i, y_i)$. Let the training set $\mathcal{T}_r$ and the test set $\mathcal{T}_e$ be defined as partitions over the entire set of T observations $\mathcal{T} = \{(\mathbf{x}^i, y_i)\}_{i=1}^{T}$.

Definition 1 (*Adversarial Classification Game*; Dalvi et al. 2004). An adversarial classification is defined as a game between two players: The CLASSIFIER who learns

[1] Regular instance refers to an object (activity) that has a regular innocent intention but nevertheless could potentially be classified as a malicious activity.

from $\mathcal{T}_r$ the classification function $\mathcal{C} : \mathcal{F}_{\mathbf{x}} \rightarrow \{+1, -1\}$, which is used to predict the behavior of adversaries in $\mathcal{T}_e$, and an ADVERSARY, whose objective is to hide his/her behavior by modifying the feature vector $\mathbf{x}$ to $\mathbf{x}' = \phi(\mathbf{x})$.

In Definition 1, function $\phi(\cdot)$ represents the strategy for modifying the original features into a new feature vector. This function will be called the *mimetic function*, which is similar to the so-called *Minimum Cost Camouflage (MCC)* (Dalvi et al. 2004). This function gives the ADVERSARY the option of mimicking regular non-malicious messages.

On the one hand, the CLASSIFIER is characterized by a utility function $U_C : \{-1, +1\} \times \{-1, +1\} \rightarrow \mathbb{R}^1$ that for a given instance $(\mathbf{x}, y)$ assigns the CLASSIFIER's utility $U_C(\mathcal{C}(\mathbf{x}), y)$ which is the utility obtained when classifying an instance whose real class is y as $\mathcal{C}(\mathbf{x})$. Generally, it can be considered that $U_C(+1, -1) < 0$ and $U_C(-1, +1) < 0$, are the costs (negative utilities) for the CLASSIFIER when instances are misclassified. Furthermore, we assume $U_C(-1, -1) > 0$ and $U_C(+1, +1) > 0$, i.e., a positive utility when correctly classifying an object. The CLASSIFIER's objective is to maximize his/her expected utility.

On the other hand, the ADVERSARY is characterized by a cost function $W : \mathbb{R}^A \times \mathbb{R}^A \rightarrow \mathbb{R}^1$, which represents the cost of changing the feature vector $\mathbf{x}$ to $\mathbf{x}'$, where $W(\mathbf{x}, \mathbf{x}) = 0$, and his/her utility function $U_A : \{-1, +1\} \times \{-1, +1\} \rightarrow \mathbb{R}^1$ which for a given instance $(\mathbf{x}, y)$ assigns the ADVERSARY'S utility $U_A(\mathcal{C}(\mathbf{x}), y)$. In general, $U_A(-1, +1) > 0, U_A(+1, +1) < 0$ and $U_A(-1, -1) = U_A(+1, -1) = 0$. The ADVERSARY also seeks to maximize his/her expected utility.

A task's complexity is determined by the computation of optimal strategies, the type of CLASSIFIER, the feature space, the ADVERSARY modifications considered to generate malicious patterns, and the respective costs. Applications on spam filtering, using a Naïve Bayes CLASSIFIER, and attacks as adversarial strategies have been presented in (Dalvi et al. 2004).

Considering the latter adversarial game properties, the ADVERSARIES might try to maximize their benefit minimizing the cost of changing their behavior. This framework, based on a single shot game of complete information, was initially tested in a spam detection domain where the adversary-aware naïve Bayes CLASSIFIER had significantly fewer false positives and false negatives than the CLASSIFIER'S plain version (Dalvi et al. 2004), i.e., without the adversarial interaction. Then a repeated version of the game was tested, in which results showed that the adversary-aware CLASSIFIER consistently outperformed the adversary-unaware naïve Bayes CLASSIFIER.

2.2 Adversarial classification extensions

The adversarial classification framework presented by Dalvi et al. (2004) assumes complete information between both players, which seems unrealistic since ADVERSARIES do not share their real intentions. As a result, this framework has been extended, by changing mainly the strategic interaction between agents. In spite of those changes, adversarial properties of spam classification have been widely used to evaluate these extensions empirically.

Different approaches consider an *adversarial Stackelberg game* model for defining the interaction between CLASSIFIER and ADVERSARY; see e.g., Brückner and Scheffer (2011), Kantarcioglu et al. (2011), Liu and Chawla (2010), and Tambe (2011). Furthermore, *Adversarial Learning*, introduced by Lowd and Meek (2005), states that any ADVERSARY who is aware of some properties of a CLASSIFIER, can reconstruct the CLASSIFIER based on reasonable assumptions and reverse engineering algorithms. Later, Biggio et al. (2008) presented a promising alternative: randomizing the CLASSIFIERS' decision function in order to hide their strategies, negatively affecting the ADVERSARIES' learning, and their possibilities of fooling the CLASSIFIERS. In addition to the latter, a multi-CLASSIFIER system is presented in (Biggio et al. 2009) for defeating the adversarial behavior in the spam context. Brückner et al. (2012) model the game between CLASSIFIER and ADVERSARY as a static game of complete information and derive conditions under which this prediction game has a unique Nash equilibrium.

Among approaches that are not using game theoretic elements, Biggio et al. (2011) present an alternative: building robust SVMs that are less sensitive to noise in the data when under the attack of ADVERSARIES. Also, in Zhou et al. (2012), specific adversarial strategies (for example, arbitrary data corruption) have been considered at the moment of modeling the SVM classifier, including a specific evaluation framework for learning these malicious activities specifically.

3 Adversarial classification using signaling games

This section provides the conceptual basis for the subsequent developments. In particular, we will need the concept of a perfect Bayesian equilibrium. Section 3.1 outlines the strategic interaction between the CLASSIFIER and the ADVERSARY.

Section 3.2 presents the definitions necessary for determining a perfect Bayesian equilibrium (PBE) in adversarial classification games, which will be used consistently throughout this article. Finally, Sect. 3.3 provides relevant details on computing such a perfect Bayesian equilibrium.

3.1 The strategic interaction

The adversarial classification game between an ADVERSARY with private information and a CLASSIFIER is considered. Basically, the ADVERSARY knows which message he/she *wants* to send (and this is private information) but he/she can choose to distort this message so as to mimic a non-malicious sender. On the other side, the CLASSIFIER chooses with which probability to reject a message, trying to accept messages coming from a non-malicious sender and reject the ones with malicious intentions.

We use the framework of a signaling game, as described by Fudenberg and Tirole (1991) and Gibbons (1992). In a signaling game, a privately informed agent (in our case the ADVERSARY) chooses an action (in our case: which message to send). Based on the action observed, and making rational inferences, another agent (the CLASSIFIER) must take an action (accepting or rejecting the message). The proposed model of incomplete information for the adversarial classification between an ADVERSARY ($\mathcal{A}$) and a CLASSIFIER ($\mathcal{C}$) has the following timing:

1. NATURE draws a type $t = (\iota, i) \in T_{\mathcal{A}} = \{R, M\} \times \{1, \ldots, k\}$ for the ADVERSARY. A type t states whether the ADVERSARY is Regular (R) or Malicious (M), and defines the preferred message i, among k existing messages.[2] NATURE draws according to the probability distribution $p(t)$, where $p(R, i) > 0, \forall \mathrm{i} \in \{1, \ldots, k\}$. In the following, we use the notation $p(t_{M,i}) := p(t = (M, i))$ and $p(t_{R,i}) := p(t = (R, i))$.
2. The ADVERSARIES observe their type t, which can be either (R, i) or $(M, i), i \in \{1, \ldots, k\}$. If it is of type (R, i), they just send message i. If, however, it is of type (M, i), they choose (possibly randomly) a message $\mathbf{x}^j$ with $j \in \{1, \ldots, k\}$. Therefore, the strategy of the ADVERSARY can be summarized by $\phi : \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \longrightarrow \Delta_k$, where Δ_k is the simplex of dimension $k - 1$ defined by $\Delta_k = \{(q_1, \ldots, q_k) | \sum_{i=1}^{k} q_i = 1, q_i \geq 0, \forall i\}$. Here $\phi_i(\mathbf{x}^l)$ denotes the probability of sending a message i when the preferred message is l.
3. The CLASSIFIER, observes the received message $\mathbf{x}^j$ (but not the sender's type $t = (\cdot, i)$) and chooses an action from $\{+1, -1\}$. Since—differing from Definition 1 in Sect. 2.1—we allow for randomization, the CLASSIFIER'S strategy becomes $\mathcal{C} : \{\mathbf{x}^1, \ldots, \mathbf{x}^k\} \longrightarrow \Delta_2$, where Δ_2 is the uni-dimensional simplex $\Delta_2 = \{(p_{+1}, p_{-1}) \in \mathbb{R}^2 | p_{+1} + p_{-1} = 1, p_{+1}, p_{-1} \geq 0\}$. Here $\mathcal{C}(\mathbf{x}^l) = (p_{+1}(\mathbf{x}^l), p_{-1}(\mathbf{x}^l))$ denotes the probability of accepting or rejecting a message of type l, respectively.
4. Since strategies may involve randomization, agents maximize their expected utilities. Therefore, the utility of the adversaries depends on their type (t), the messages they chose to send (x^l), and the decision made by the classifier (d). The same is true for the classifier. In our example below (see Sect. 6), we model these utility functions (U_C, U_A) explicitly, in a way that is consistent with the respective application for phishing detection.

The extensive-form game that represents the signaling game between ADVERSARY and CLASSIFIER is presented in Fig. 1, where we can see what happens when agents want to send messages $\mathbf{x}^i$. First, if the agents are non-malicious, they choose $\mathbf{x}^i$. In the opposite case, they can choose either $\mathbf{x}^i$ or a different message $\mathbf{x}^j$. In both cases, their messages are "pooled" with the ones sent by other agents (malicious or not). Based on only these messages, the classifiers make the decision to either accept or reject. The dotted sets represent what in Game Theory is known as *information sets*, i.e., a set of nodes from which a player chooses but the nodes in such a set are indistinguishable for the player who is choosing; Fudenberg and Tirole (1991).

In our context the CLASSIFIER must choose the same action for every node in that information set. In our application (see Sect. 6) we will determine such information sets by clustering messages, i.e., grouping similar messages together and having dissimilar messages in different sets. Then each information set will be represented by its centroid.

[2] Parameter k, which is the total number of messages that can be drawn by NATURE. It has to be determined context-dependently; see, for example, the application in Sect. 6

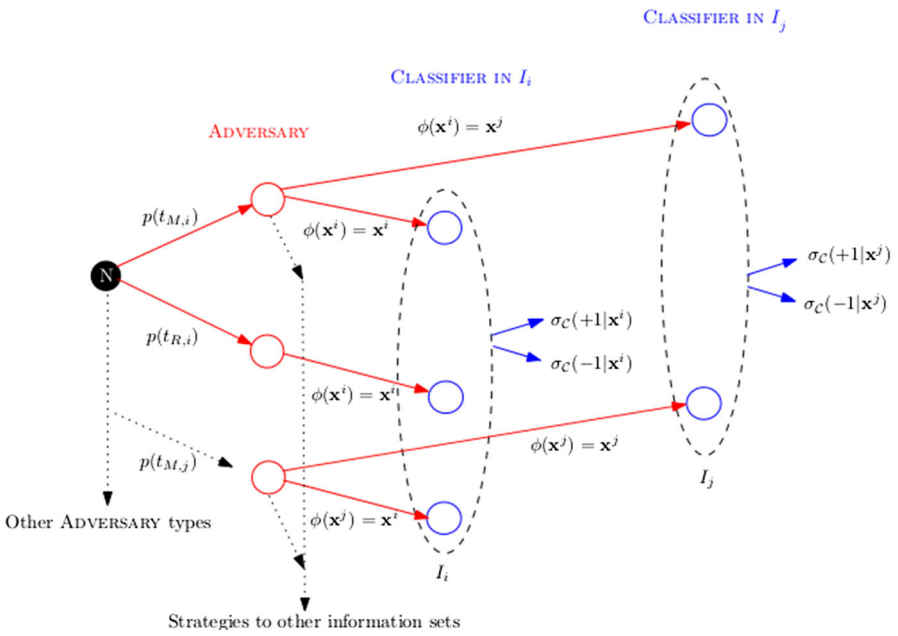

Fig. 1 Extensive-form representation of the signaling game between the Classifier and the Adversary

3.1.1 Adversarial classification using incomplete information games: an example

The example shown in Fig. 2 presents a simple case for the proposed game interaction between a Classifier and an Adversary. In this case, values for payoffs were given just as examples for illustrating how the Classifier and the Adversary interact.

As shown in Fig. 2, there are two types of Adversaries and a Classifier. The Adversaries' types are characterized by their intention of committing fraud or not committing fraud by sending either regular or malicious messages. The types' probabilities have the following distribution: $\frac{1}{20}$ for regular senders who send messages of the "fraudulent type" but with no bad intentions (i.e., using malicious words in regular emails), $\frac{4}{20}$ for malicious senders whose intention is to commit fraud, and $\frac{15}{20}$ for those regular senders whose messages do not contain malicious words.

Figure 2 also exhibits the payoffs as described here. Whenever the Classifier commits a mistake by classifying regular messages as malicious (i.e., rejects a non-fraud message from a regular non-malicious type sender), his/her payoff is -40. The sender also has a payoff of -40 in both cases (sending unintentionally malicious messages, or just regular typed messages). In the case where a regular message is well classified, both players have a utility of $+2$. If a malicious-type Adversary sends a malicious message, and the Classifier performs correctly, their payoffs are -2 and $+2$, respectively. In this case, whenever the Classifier is wrong (i.e., lets fraud messages pass through), their payoffs are $+20$ and -20, respectively. Finally, when the Adversary decides to change the message features (e.g., words, or URLs, among

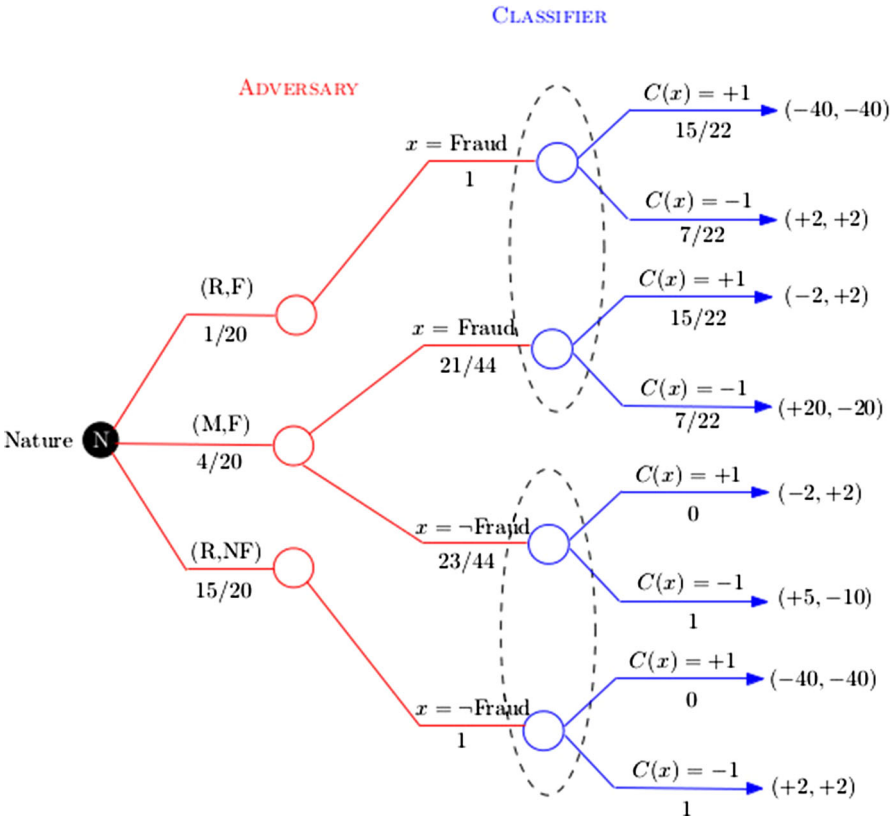

Fig. 2 Dynamic interaction between CLASSIFIER and ADVERSARY

other fraud strategies), so as to mislead the CLASSIFIER, and is successful, payoffs are $+5$ and -10 for the ADVERSARY and the CLASSIFIER, respectively.

In the equilibrium of this game, mixed strategies are determined as here outlined. The ADVERSARY modifies his/her message with probability $\frac{23}{44}$, accepting the cost of a worse payoff in case the message gets through in exchange for a higher probability of getting the message accepted by the CLASSIFIER. This is reasonable since in equilibrium the CLASSIFIER always accepts messages of the non-fraudulent type, while it accepts fraudulent types only with probability $\frac{7}{22}$.

3.2 Sequential equilibria in adversarial classification

We adopt the concept of Bayesian Nash equilibrium (Harsanyi 1968). Basically, we require that each agent plays an optimal strategy as a response to the other player's strategy (sequential rationality), considering his/her own beliefs. These properties will be presented in this subsection and used in Sect. 5.2 in order to determine classifier strategies. Moreover, these beliefs must be consistent with the strategies that the players are using.

A belief $\mu(t|\mathbf{x}^j)$ represents the probability of observing a particular type, t, given that the observed message is $\mathbf{x}^j$.

Definition 2 (*Consistent Belief (CB)*): For each observed message $\mathbf{x}^j$, the CLASSIFIER's belief $\mu(t|\mathbf{x}^j)$ is said to be **consistent** if it follows Bayes' rule given the prior beliefs and the ADVERSARY'S strategy:

$$\mu(t|\mathbf{x}^j) = \begin{cases} \frac{\phi(\mathbf{x}^j|t)p(t)}{\sum_l [p(R,\mathbf{x}^l)+p(M,\mathbf{x}^l)\phi(\mathbf{x}^j|(M,\mathbf{x}^l))]} & \text{if} \quad t=(M,\mathbf{x}^i) \\ \frac{p(t)}{\sum_l [p(R,\mathbf{x}^l)+p(M,\mathbf{x}^l)\phi(\mathbf{x}^j|(M,\mathbf{x}^l))]} & \text{if} \quad t=(R,\mathbf{x}^j) \\ 0 & \text{if} \quad t=(R,\mathbf{x}^i),\ j\neq i \end{cases} \tag{1}$$

This requires beliefs to be consistent in information sets that are reached with positive probability in equilibrium (otherwise, Bayes rule cannot be applied) (Fudenberg and Tirole 1991).

Given these beliefs, each player must be reacting optimally to the strategy played by his/her opponent. In particular, the CLASSIFIER must accept a message, if and only if, the expected payoff when accepting it (according to its beliefs about who sent it) is greater than the payoff obtained when rejecting it. In case both payoffs are equal, the CLASSIFIER can play a mixed strategy, accepting and rejecting the message with positive probability.

Definition 3 (*Sequential Rationality for the* **CLASSIFIER** *(SRC)*) For each observed message $\mathbf{x}^j$, the CLASSIFIER's optimal strategy, defined as the probability distribution $\mathcal{C}^*(\mathbf{x}^j)$ over $\{-1,+1\}$ represented by $\alpha_C \in \Delta_2$, must maximize the CLASSIFIER's expected utility, given the beliefs $\mu(t|\mathbf{x}^j)$, that is,

$$\mathcal{C}^*(\mathbf{x}^j) \in \arg\max_{\alpha_C\in\Delta_2} \sum_{t\in T_A} \mu(t|\mathbf{x}^j) \sum_{d\in\{-1,+1\}} \alpha_C U_C(t,\mathbf{x}^j,d)), \forall \mathbf{x}^j \tag{2}$$

Given the messages that the CLASSIFIER is accepting, ADVERSARIES of the malicious type must choose the messages they send optimally.

Definition 4 (*Sequential Rationality for the* **ADVERSARY** *(SRA)*): For each type $(M,\mathbf{x}^j)$, the ADVERSARY's optimal strategy $\phi^*(\mathbf{x}^j)$, must maximize the ADVERSARY's utility function, given the CLASSIFIER's strategy $\mathcal{C}^*$, that is

$$\phi^*(\mathbf{x}^j) \in \arg\max_{\phi_A\in\Delta_k} \sum_{i=1}^{k} \phi_{A,i} \sum_{d\in\{-1,+1\}} \mathcal{C}_d^*(\mathbf{x}^i) U_A((M,\mathbf{x}^j),\mathbf{x}^i,d), \forall \mathbf{x}^j \tag{3}$$

Based on previous definitions, a PBE is defined.

Definition 5 A PBE in the adversarial signaling game is a pair of mixed strategies ϕ^*, $\mathcal{C}^*$ and a belief system $\mu(t|\mathbf{x}^j)$ satisfying *SRC, SRA*, and *CB*.

In the following, mixed strategies ϕ^* and $\mathcal{C}^*$ will be denoted as $\sigma_{\mathcal{A}}^*$ and $\sigma_{\mathcal{C}}^*$, respectively.

It is important to mention that throughout the paper, utility functions for the ADVERSARY and the CLASSIFIER will be represented by functions that depend on the message type, the feature vector, and the CLASSIFIER'S decision.

3.3 Computing perfect Bayesian equilibrium

In practice, solving for a PBE may involve some difficulties, not least because of the existence of multiple equilibria. We have already mentioned that every information set is reached with positive probability, since an information set for the CLASSIFIER is a message x^i, and certainly some messages of this type are observed, at least the ones sent by non-malicious agents of type (R, i). This helps to prune many undesirable equilibria that rely on unreasonable off-path beliefs.[3] Still, multiple equilibria may exist.

In order to select a plausible equilibrium, we introduce the notion of QRE. We relax the assumption that agents react optimally to the other agents' actions and incorporate stochastic elements in the agents' decisions. We assume that agents are more likely to select better choices than worse choices, but do not necessarily succeed in selecting the very best choice. Given a strategy profile π, the probability that an action a is chosen if the information set $I(a)$ is reached is given by

$$\pi_a = \frac{e^{\lambda U_a(\pi)}}{\sum_{b \in I(a)} e^{\lambda U_b(\pi)}} \tag{4}$$

where $U_a(\pi)$ is the expected utility associated with playing the action a if all other agents play according to the strategy profile π. This equilibrium depends crucially on parameter λ. For $\lambda = 0$, every agent randomizes uniformly over the available actions at each information set. As λ goes to infinity, agents approximate the best response. Then, we can approximate a sequential equilibrium through a sequence of logit approximations as stated in the following theorem.

Theorem 1 (logit QRE (LQRE) Sequential Equilibria) *For every finite extensive form game, every limit point of a sequence of logit-QREs with λ going to infinity corresponds to the strategy of a sequential equilibrium, and in particular, of a PBE (McKelvey and Palfrey 1998).*

We follow this approximating strategy, computing a particular PBE as the limit of equilibria where agents, after a longer playing history, play more in accordance with a best response. This allows us to consider an equilibrium not only as a static concept, reached in a non-specified way, but as the limit of approximated games, in which agents trust their perceptions more and more about the frequency with which other agents play their strategies.

Computationally, this approach also has significant advantages. Computing such a QRE equilibrium for a given λ is easy, since it only requires tracing zeroes in a linear system of equations. A numerical approximation was proposed by Turocy (2010) based on previous work (McKelvey and Palfrey 1998; Turocy 2005), using a transformation of the logit QRE correspondence. Moreover, an open-source project for estimating equilibrium results in finite games is computationally simple and has

[3] An information set is said to be off-path if it cannot be reached with the given strategy profile. The concept of sequential equilibria puts almost no discipline on the beliefs at information sets that are not reached in equilibria. A significant literature on "refinements" has tried to rule out unreasonable equilibria. See, for example, Cho and Kreps (1987).

been implemented in `Gambit` (McKelvey et al. 2010), a tool which we also used in Sect. 6 for our application on phishing detection. Further details on this open-source project and numerical experiments are presented in Turocy (2010).

4 Quantal response equilibrium classifiers

Different from standard equilibrium definitions in Game Theory which assume agents to be fully rational, the concept of QRE incorporates stochastic elements in the respective decision-making process and assumes that agents are more likely to select better choices than worse choices. In games of incomplete information, beliefs determine the expected payoff of choosing an action. In sequential equilibria, agents are assumed to choose the action with the highest expected payoff with probability 1. In QRE, however, agents do not completely trust their beliefs (because they have been constructed from a short history, for example) and simply assign a higher probability to actions that have a higher expected value.

As the quantal response function assigns the highest probability to the very best action, the respective equilibrium converges to the standard notion of bayesian equilibrium. In our approach, we assume that for early stages of the game, the classifiers use relatively flat quantal response functions, since they do not trust much in their beliefs, which are based on past observations. As the classifier observes more cases to be classified, the quantal response function becomes steeper, leading to a classification strategy which almost always chooses the very best action.

However, QRE classifiers classify messages based just on game-theoretical considerations, i.e., without using state-of-the-art Data Mining algorithms and as such serve as benchmarks for different classifiers developed in Sect. 5 and applied for phishing detection in Sect. 6.

Section 4.1 introduces the utility functions that will be used in the subsequent Sects. 4.2 and 4.3 where S-QRE as well as P-QRE are presented.

4.1 Utility function modeling

The payoffs for the CLASSIFIER and the ADVERSARY are determined for an incoming message $\mathbf{x}^i$ which has been defined by NATURE. Furthermore, we assume the decision function $d : \mathbb{R}^A \to \{+1, -1\}$ for the CLASSIFIER, and the transformation of message $\mathbf{x}^i$ into $\mathbf{x}^j$ by the mimetic function ϕ chosen by the ADVERSARY. Also, let $y \in \{+1, -1\}$ be the label of a given message. Eqs. (5) and (6) represent the utility function for the CLASSIFIER and ADVERSARY, respectively.

$$U_{\mathcal{C}}(t, \cdot, d) = \begin{cases} u_{\mathcal{C}} & \text{if} \quad t = (M, \cdot) \text{ and } d = +1 \\ u_{\mathcal{C}} & \text{if} \quad t = (R, \cdot) \text{ and } d = -1 \\ -\Omega_1 & \text{if} \quad t = (M, \cdot) \text{ and } d = -1 \\ -\Omega_2 & \text{if} \quad t = (R, \cdot) \text{ and } d = +1 \end{cases} \tag{5}$$

where $\Omega_1 \geq \Omega_2 \geq 0$. In this case, Ω_2 is the cost of classifying a non-malicious message as malicious, and Ω_1 is the cost of classifying a malicious message as non-

malicious, and parameter $u_{\mathcal{C}}$ indicates the baseline utility for the CLASSIFIER. In this case, the utility function is determined only by the malicious or regular type and its respective decision outcome.

Let $\theta_1 = \{\mathbf{x}^i | ((M, \mathbf{x}^i) = +1 \wedge d = +1) \vee ((R, \mathbf{x}^i) = -1 \wedge d = -1)\}$ be the set of messages that are correctly classified, and $\theta_2 = \{\mathbf{x}^j | ((M, \mathbf{x}^j) = +1 \wedge d = -1) \vee ((R, \mathbf{x}^j) = -1 \wedge d = +1)\}$ be the set of messages that are incorrectly classified, where $t = (\cdot, \cdot)$ is the type of agent, indicating if it is malicious or not and the message he/she wants to send, as defined in Sect. 3.1.

Following the general rules for utility functions presented in Sect. 2.1, the ADVERSARY'S utility function is proposed as,

$$U_{\mathcal{A}}(t, \mathbf{x}^j, d) = \begin{cases} u_{\mathcal{A}} - (\eta_\epsilon + \eta_{\mathcal{A}})\delta(c(\mathbf{x}^i), c(\mathbf{x}^j)) & \text{if} \quad \mathbf{x}^j \in \theta_1 \\ -\eta_\epsilon \delta(c(\mathbf{x}^i), c(\mathbf{x}^j)) & \text{if} \quad \mathbf{x}^j \in \theta_2 \end{cases} \tag{6}$$

Values η_ϵ and $\eta_{\mathcal{A}}$ represent factors which amplify the distance function $\delta : \mathbb{R}^A \times \mathbb{R}^A \to \mathbb{R}^1$ between the centroid of the information set that contains message $\mathbf{x}^i$, and the centroid of the information set that contains message $\mathbf{x}^j$ sent by the ADVERSARY, and $\eta_{\mathcal{A}} \geq \eta_\epsilon$. Finally, parameter $u_{\mathcal{A}}$ indicates the baseline value for the ADVERSARY's utility.

4.2 Static quantal response equilibrium classifier

The S-QRE is a one-shot representation of the interaction between CLASSIFIER and ADVERSARY in a signaling game. Here, the objective is to determine whether in a completely strategic interaction, the probability distribution $\sigma^*_{\mathcal{C}}$ over the CLASSIFIER's actions (or mixed strategies) in the sequential equilibria can be used for the decision of labeling a given message $\mathbf{x}^j \in \mathcal{T}$ as malicious ($\mathcal{C}(\mathbf{x}^j) = +1$) or regular ($\mathcal{C}(\mathbf{x}^j) = -1$).

The interaction between ADVERSARY and CLASSIFIER is determined by messages arriving at time τ, extending definitions introduced in Sect. 2. Throughout this paper, a message $\mathbf{x}^j$ that arrives at time τ will be represented by $\mathbf{x}^j_\tau$.

Algorithm 1 represents the simplest case for the interaction between a CLASSIFIER and an ADVERSARY. The decision function is based on a mixed strategies profile $\sigma^*_{\mathcal{C}}$ determined by the QRE over the training set ($\mathcal{T}_r$). The test dataset $\mathcal{T}_e$ is used to evaluate the classification performance with function $\mathcal{E}$rr. This approach represents the simplest case for strategic analysis in which no dynamism is considered, and the evaluation of messages is static. In this case, intuition tells that the classification error is incremental over time given the incremental properties of the game between agents.

The following Algorithm 1 takes a dataset $\mathcal{T}$ and a number k of sets of messages as input. These sets are the information sets $\{I_i\}_{i=1}^k$ introduced in Sect. 3.1 which can be determined using any kind of clustering technique that provides centroids $\{c_i\}_{i=1}^k$ (see e.g., Xu and Wunsch (2005)) representing the respective clusters, i.e., types of messages.

Then, the probabilities of the types $\{p(t_i)\}_{i=1}^k$ are calculated based on the frequency of message types in $\mathcal{T}$. Subsequently, utilities for the ADVERSARY and the CLASSIFIER

Algorithm 1 S-QRE

Input: $\{\mathcal{T}_r, \mathcal{T}_e, k, u_C, u_A, \eta_A\}$
Output: Classification Hypothesis $\{H_{\mathcal{T}_e} = h_\tau, \forall \mathbf{x}_\tau \in \mathcal{T}_e\}$, Error $\mathcal{E}rr$

$\{c_i\}_{i=1}^k$ = getClusters$(\mathcal{T}_r, k)$
$\{I_i\}_{i=1}^k$ = build k information sets, represented by centroids $\{c_i\}_{i=1}^k$.
$p(t_i) = |\{x|x$ is a message in cluster $c_i\}|/|\mathcal{T}_r|, \forall i \in \{1, \ldots, k\}$
$\mathbf{u}_A \leftarrow [\cdot], \mathbf{u}_C \leftarrow [\cdot]$
for each $t_i, t_j \in T_{\mathcal{A}}, d \in \{+1, -1\}$ **do**
  set $\mathbf{u}_C[t_i, d] \leftarrow U_{\mathcal{C}}(t_i, d)$ according to Eq. 5
  set $\mathbf{u}_A[t_i, t_j, d] \leftarrow U_{\mathcal{A}}(t_i, t_j, d)$ according to Eq. 6
end for
$\sigma_{\mathcal{C}}^*$ = `traceZeroes`$(\mathbf{u}_A, \mathbf{u}_C, \{I_i\}_{i=1}^k, \{p(t_i)\}_{i=1}^k, \{c_i\}_{i=1}^k)$
for each $(\mathbf{x}_\tau, y_\tau) \in \mathcal{T}_e$ **do**
  h_τ = `flipCoin`$(\sigma_{\mathcal{C}}^*)$
  $\mathcal{E}$rr(h_τ, y_τ)
end for

are determined. Using all of the previous as input for the `traceZeroes` procedure,[4] the mixed strategies for the classifier $\sigma_{\mathcal{C}}^*$ are established.

Finally, function `flipCoin`: $[0, 1] \times [0, 1] \rightarrow [0, 1]$ takes values from $\sigma_{\mathcal{C}}^*(+1|\cdot)$ and flips a biased coin for the classification hypothesis. If $h_\tau = +1$, the CLASSIFIER's decision is $+1$ (a malicious message), and if $h_\tau = -1$, the CLASSIFIER's decision is -1 (a regular message). The performance of the classifier is determined by the cumulative classification error, calculated by function $\mathcal{E}$rr.

4.3 Quantal response equilibrium perceptron classifier

For the QRE-perceptron algorithm[5] (P-QRE), the game starts when both agents (CLASSIFIER and ADVERSARY) are first introduced to each other with the first message whose type is unknown. The CLASSIFIER, who does not have any information about the game, can have only a 50 % chance of classifying the message as malicious or regular. Then, as the game develops, new information about the ADVERSARY is presented, forcing the CLASSIFIER to infer mixed strategies $\sigma_{\mathcal{C}}^*$ dynamically for the game.

Initially, mixed strategies and type probabilities are considered as a uniform distribution over the set of elements in both cases, i.e., $\sigma_{\mathcal{C}}^*(+1|\cdot) = 0.5$. The QRE Perceptron algorithm follows:

In this case, all game-theoretical values are updated as the interaction between agents evolves. The main idea, however, is to modify the respective parameters only if the CLASSIFIER commits an error in his/her decisions, reducing the time the CLASSIFIER needs for evaluation. This property is relevant if online classification algorithms are to be implemented in real-world applications.

[4] This procedure is based on input parameters defined in the `Gambit` software tool for the QRE computation (Turocy 2010).

[5] Based on the dynamic update of parameters given classification mistakes originally proposed by Rosenblatt (1962).

Algorithm 2 P-QRE

Input: $\{\mathcal{T}, k\}$
Output: Classification Hypothesis $\{H_{\mathcal{T}} = h_\tau, \forall \mathbf{x}_\tau \in \mathcal{T}\}$, Error $\mathcal{E}$rr
1: Initialize $\mathcal{S} \leftarrow \emptyset$ $\sigma_C^* = (0.5, 0.5)$
2: **for** each $\mathbf{x}_\tau, y_\tau$ **do**
3: h_τ = `flipCoin`(σ_C^*)
4: $\mathcal{E}\text{rr}(h_\tau, y_\tau)$
5: $\mathbf{u}_A \leftarrow [\cdot]$, $\mathbf{u}_C \leftarrow [\cdot]$
6: **if** $h_\tau \neq y_\tau$ **then**
7: Initialize $\{c_i\}_{i=1}^k$ = getClusters$(\mathcal{S}, k)$
8: $\{I_i\}_{i=1}^k$ = build k information sets, represented by centroids $\{c_i\}_{i=1}^k$.
9: $p(t_i) = |\{x | x \text{ is a message in cluster } c_i\}|/|\mathcal{T}_r|, \forall i \in \{1, \ldots, k\}$
10: **for** each $t_i, t_j \in T_{\mathcal{A}}, d \in \{+1, -1\}$ **do**
11: set $\mathbf{u}_C[t_i, d] \leftarrow U_{\mathcal{C}}(t_i, d)$ according to Eq. 5
12: set $\mathbf{u}_A[t_i, t_j, d] \leftarrow U_{\mathcal{A}}(t_i, t_j, d)$ according to Eq. 6
13: **end for**
14: σ_C^* = `traceZeroes`$(\mathbf{u}_A, \mathbf{u}_C, \{I_i\}_{i=1}^k, \{p(t_i)\}_{i=1}^k, \{c_i\}_{i=1}^k)$
15: **end if**
16: $\mathcal{S}$.add$(\mathbf{x}_\tau, y_\tau)$
17: **end for**

Intuitively, this approach does not represent the strategic interaction, since the CLASSIFIER is forced to decide at each $\mathbf{x}_\tau$ without sufficient information to compute the mixed strategies. However, this perceptron-like algorithm represents the extreme case of the incremental interaction, as mixed strategies are updated every time the classifier fails. Therefore, it is considered to be relevant as a benchmark strategic algorithm in Sect. 6.3.

5 Adversary-aware online support vector machines

In this section, we present the paper's main contribution which is a combination of Game Theory and Data Mining. Using Game Theory, we model the strategic interaction between ADVERSARY and CLASSIFIER. Unlike previous algorithms presented in Sect. 4, which are simple classification rules using just the strategic interaction between agents, in this section we propose *explicitly* integrating such an interaction into a Support Vector Machines (SVM) formulation as prior knowledge. Furthermore, sequential rationality conditions as introduced in Sect. 3.2 are included into the novel SVM formulation as new constraints. Its online evaluation leads to the proposed Adversary-Aware Online SVM algorithm, called AAO-SVM.

Additionally, we propose advanced techniques for feature selection. Whereas previously introduced QRE algorithms (S-QRE and P-QRE) do not consider a classification function shaped by object features, the approach presented in this section includes such features for building a classifier explicitly, considering mixed strategies as input.

Subsequently, we first describe utility functions and payoffs, then the CLASSIFIER'S strategies, and finally the classification model and the respective algorithm.

Table 1 Classification payoff matrix

	$y =$ Malicious	$y =$ Regular
$\mathcal{C}(\mathbf{x}) = +1$	ϵ_M	$-\gamma_R$
$\mathcal{C}(\mathbf{x}) = -1$	$-\gamma_M$	ϵ_R

5.1 Utility functions and payoffs

We introduce our utility modeling using the terminology of the subsequent application of phishing filtering (see Sect. 6). The basic ideas, however, are generic and can be used in any application of adversarial classification.

5.1.1 Utility function for the CLASSIFIER

The construction of the CLASSIFIER'S utility function will be based on the non-negative parameters ϵ_M, ϵ_R, γ_M, and γ_R introduced in the payoff matrix presented in Table 1. Payoff is positive (negative) in the case of a correct (incorrect) decision.

We first motivate the utility function's structure for $A = 1$ feature and proceed subsequently with the general case of any number of features.

Consider a dataset characterized by just one binary feature x_1. Let us suppose that this feature represents the usage of a word often used in phishing emails, e.g., "password". Therefore, if "password" is used in a message, then $x_1 = 1$; in the opposite case we will have $x_1 = 0$. Suppose the classification function is defined by $y = w \cdot x_1 + b$. If $y \geq \nu$, for a given threshold ν, the message is classified as phishing (malicious), and if $y < \nu$, it is classified as regular (non-malicious).

Suppose that for a particular message the correct label is *malicious*. If the respective feature is used in this message (i.e., $x_1 = 1$), then the value for the classification function is $y = w + b$, which makes it more likely that $y > \nu$, and therefore that the message will be classified as malicious. The CLASSIFIER'S utility, if the message is correctly classified as malicious, is proposed to be $U_{\mathcal{C}} = \epsilon_M(w + b)$, i.e., the maximum reward. Furthermore, if the CLASSIFIER decides incorrectly that the message is not malicious, even if $x_1 = 1$, he/she must be penalized with the maximum cost $U_{\mathcal{C}} = -\gamma_M(w + b)$, where $\gamma_M > 1$ is a misclassification cost parameter. If $x_1 = 0$, then the classifiers are rewarded (or penalized) with a lower value, as they manage to classify the given message without phishing-like properties correctly as malicious, or misclassify the malicious message without phishing properties as regular, respectively. Based on this idea, the CLASSIFIER'S payoffs for the **malicious** cases are defined as follows:

$$U_{\mathcal{C}}(M, x_1, \mathcal{C}(x_1)) = \begin{cases} \epsilon_M \cdot (w + b) & \text{if} \quad x_1 = 1, \quad \mathcal{C}(x_1) = +1 \\ \epsilon_M \cdot b & \text{if} \quad x_1 = 0, \quad \mathcal{C}(x_1) = +1 \\ -\gamma_M \cdot (w + b) & \text{if} \quad x_1 = 1, \quad \mathcal{C}(x_1) = -1 \\ -\gamma_M \cdot b & \text{if} \quad x_1 = 0, \quad \mathcal{C}(x_1) = -1 \end{cases}$$

If, on the other hand, the type of a given message is **regular**, the payoffs behave differently. Here, the maximum payoff can be achieved when the classifier decides it is "not phishing", if no phishing-like feature is present. The minimum payoff is obtained when the regular message does not use a phishing-like feature but is classified as malicious. Here, parameters ϵ_R and γ_R are introduced to differentiate the payoffs from the previous case.

$$U_{\mathcal{C}}(R, x_1, \mathcal{C}(x_1)) = \begin{cases} \epsilon_R \cdot (w + b) & \text{if} \quad x_1 = 0, \quad \mathcal{C}(x_1) = -1 \\ \epsilon_R \cdot b & \text{if} \quad x_1 = 1, \quad \mathcal{C}(x_1) = -1 \\ -\gamma_R \cdot (w + b) & \text{if} \quad x_1 = 0, \quad \mathcal{C}(x_1) = +1 \\ -\gamma_R \cdot b & \text{if} \quad x_1 = 1, \quad \mathcal{C}(x_1) = +1 \end{cases}$$

For the general case with $A \geq 1$ features, the classification function is defined by $y = \mathbf{w}^T \cdot \mathbf{x} + b$, where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^A$.

Recalling that $T_{\mathcal{A}} = \{t_{R,J}\}_{J=1}^{k} \bigcup \{t_{M,J}\}_{J=1}^{k}$, the general case for the CLASSIFIER's utility functions can be separated into four cases. The first two instances occur when a message of regular type ($\{t_{R,J}\}_{J=1}^{k}$) is classified as malicious or regular. Two other instances arise when a message of malicious type ($\{t_{M,J}\}_{J=1}^{k}$) is classified as malicious or regular. These four cases are modeled by the following utility functions:

$$U_{\mathcal{C}}(t_{R,J}, \mathbf{x}, \mathcal{C}(\mathbf{x})) = -\gamma_R \cdot (\mathbf{w}^T \cdot (\mathbf{e} - \mathbf{x}) + b), \text{ for } \mathcal{C}(\mathbf{x}) = +1 \tag{7}$$

$$U_{\mathcal{C}}(t_{R,J}, \mathbf{x}, \mathcal{C}(\mathbf{x})) = \epsilon_R \cdot (\mathbf{w}^T \cdot (\mathbf{e} - \mathbf{x}) + b), \text{ for } \mathcal{C}(\mathbf{x}) = -1 \tag{8}$$

$$U_{\mathcal{C}}(t_{M,J}, \mathbf{x}, \mathcal{C}(\mathbf{x})) = -\gamma_M \cdot (\mathbf{w}^T \cdot \mathbf{x} + b), \text{ for } \mathcal{C}(\mathbf{x}) = -1 \tag{9}$$

$$U_{\mathcal{C}}(t_{M,J}, \mathbf{x}, \mathcal{C}(\mathbf{x})) = \epsilon_M \cdot (\mathbf{w}^T \cdot \mathbf{x} + b), \text{ for } \mathcal{C}(\mathbf{x}) = +1 \tag{10}$$

where $\mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^A$. It is important to consider that the previous utility functions are specially designed for binary features that capture only the existence of malicious elements, i.e., if all features are set to 1, the message is likely to be malicious, and if all features are set to 0, it is likely to be regular.

5.1.2 Utility function for the ADVERSARY

To model the ADVERSARY'S utility function, we have to remember that malicious ADVERSARIES may change their originally intended message; i.e., payoffs should be related to the distance between the original message type t_l and the finally emitted message of type $t_j, \mathbf{x}_\tau^j$ at time τ. This can be calculated as the distance between the original message type and the centroid of the information set which contains the emitted message using a distance function $\delta : \mathbb{R}^A \times \mathbb{R}^A \to \mathbb{R}$.

In the simple case, where the ADVERSARY is regular, his/her utility function properties are the same as for the CLASSIFIER, since in this case both agents are rewarded in the same way.

In case the ADVERSARY is malicious, modeling the utility function must contemplate different evaluation scenarios. First, when the classification is $+1$ and the original

message type has been changed in order to disable the malicious feature, i.e., $x_1 = 0$, the utility is negative and less than the case when the original type would not have been changed. Second, when the classification label is -1, the utility takes positive values considering differences whether or not the malicious feature was used. In case the feature is used, the distance between the original type and the one chosen by the ADVERSARY must be taken into consideration.

$$U_{\mathcal{A}}(M, x_1, \mathcal{C}(x_1)) = \begin{cases} -\epsilon_M \cdot (w \cdot (1+1) + b) & \text{if} \quad x_1 = 0, \mathcal{C}(x_1) = +1 \\ -\epsilon_M \cdot (w + b) & \text{if} \quad x_1 = 1, \mathcal{C}(x_1) = +1 \\ \gamma_M \cdot b & \text{if} \quad x_1 = 0, \mathcal{C}(x_1) = -1 \\ \gamma_M \cdot (w + b - 1) & \text{if} \quad x_1 = 1, \mathcal{C}(x_1) = -1 \end{cases}$$

Following the same idea used for the general case when modeling the CLASSIFIER'S utility function (i.e., (7), …, (10)), the utility functions for the ADVERSARY are:

$$U_A(t_{R,j}, \mathbf{x}, \mathcal{C}(\mathbf{x}) = +1) = -\gamma_R \cdot (\mathbf{w}^T \cdot (e - \mathbf{x}) + b) \tag{11}$$

$$U_A(t_{R,j}, \mathbf{x}, \mathcal{C}(\mathbf{x}) = -1) = \epsilon_R \cdot (\mathbf{w}^T \cdot (e - \mathbf{x}) + b) \tag{12}$$

$$U_A(t_{M,j}, \mathbf{x}, \mathcal{C}(\mathbf{x}) = +1) = -\epsilon_M \cdot (\mathbf{w}^T \cdot ((e - \mathbf{x}) + \delta(e - \mathbf{x}, \mathbf{c}_j)) + b) \tag{13}$$

$$U_A(t_{M,j}, \mathbf{x}, \mathcal{C}(\mathbf{x}) = -1) = \gamma_M \cdot (\mathbf{w}^T \cdot \mathbf{x} + b - \delta(\mathbf{x}, \mathbf{c}_j)) \tag{14}$$

where $\mathbf{c}_j$ is the centroid of the cluster associated with type $t_{\cdot,j}$. As already mentioned above, Eq. (7) is identical to Eq. (11) and Eq. (8) is identical to Eq. (12).

5.2 CLASSIFIER'S strategies

The sequential rationality for the CLASSIFIER *SRC* postulates the maximization of the CLASSIFIER'S expected utility solving problem (2); see Definition 3. This leads to constraints that will be added to the respective classification problem as prior knowledge.

$$\mathcal{C}^*(\mathbf{x}^j) \in \arg\max_{\alpha_C \in \Delta_2} \sum_{t \in T} \mu(t|\mathbf{x}^j) \alpha_C \cdot (U_C(t, \mathbf{x}^j, +1), U_C(t, \mathbf{x}^j, -1)), \forall \mathbf{x}^j \tag{15}$$

As mentioned in Sect. 3.1, the CLASSIFIER'S optimal strategies are defined by $\alpha_C \in \Delta_2$, which represents the mixed strategies over the set of actions $l \in \{+1, -1\}$. Considering the case where the expected utility achieves its maximum values for $\mathcal{C}(\mathbf{x}_\tau^j) = +1$, and pure strategies, Problem (2) leads to $\mathcal{C}^*(x^j) = 1$ iff

$$\sum_{t \in T_{\mathcal{A}}} \mu(t|\mathbf{x}^j) \cdot U_{\mathcal{C}}(t, \mathbf{x}^j, +1) > \sum_{t \in T_{\mathcal{A}}} \mu(t|\mathbf{x}^j) \cdot U_{\mathcal{C}}(t, \mathbf{x}^j, -1) \tag{16}$$

This can be interpreted as follows: For a given message $\mathbf{x}^j$ the optimal decision by the classifier is +1 iff the total belief when deciding +1 for all adversarial types is greater than the total belief when deciding -1 for all adversarial types.

We can now partition the set of all adversarial types as $T_{\mathcal{A}} = T_{\mathcal{M}} \cup T_{\mathcal{R}}$, where $T_{\mathcal{M}} = \{t_{M,J}\}_{j=1}^{k}$ and $T_{\mathcal{R}} = \{t_{R,J}\}_{j=1}^{k}$ contain the messages available to the malicious ($\mathcal{M}$) and regular ($\mathcal{R}$) ADVERSARY, respectively. Therefore, we may rewrite condition (16) as:[6]

$$\sum_{t_{M,J} \in T_{\mathcal{M}}} \mu\left(t_{M,J}|\mathbf{x}^j\right) \Delta U_{\mathcal{C},M}^{t_{M,J}}(\mathbf{x}^j) > \sum_{t_{R,J} \in T_{\mathcal{R}}} \mu\left(t_{R,J}|\mathbf{x}^j\right) \Delta U_{\mathcal{C},R}^{t_{R,J}}(\mathbf{x}^j) \tag{17}$$

where $\mu(t_{R,J}|\mathbf{x}^j)$ is determined by Eq. (1), and the values for $\Delta U_{\mathcal{C},M}^{t_{M,J}}(\mathbf{x}^j)$ and $\Delta U_{\mathcal{C},R}^{t_{R,J}}(\mathbf{x}^j)$ are calculated, using the payoff matrix shown in Table 1 as

$$\begin{aligned}
\Delta U_{\mathcal{C},M}^{t_{M,J}}(\mathbf{x}^j) &:= U_{\mathcal{C}}(t_{M,J}, \mathbf{x}^j, +1) - U_{\mathcal{C}}(t_{M,J}, \mathbf{x}^j, -1) && (18)\\
&= (\epsilon_M + \gamma_M) \cdot (\mathbf{w}^T \cdot \mathbf{x}^j + b) && (19)\\
\Delta U_{\mathcal{C},R}^{t_{R,J}}(\mathbf{x}^j) &:= U_{\mathcal{C}}(t_{R,J}, \mathbf{x}^j, -1) - U_{\mathcal{C}}(t_{R,J}, \mathbf{x}^j, +1) && (20)\\
&= (\epsilon_R + \gamma_R) \cdot (\mathbf{w}^T \cdot (\mathbf{e} - \mathbf{x}^j) + b) && (21)
\end{aligned}$$

Using this on Condition (17) we obtain

$$\mathcal{C}^*(x^j) = 1 \iff (\mathbf{w}^T \cdot \mathbf{x}^j + b) > \frac{(\mathbf{w}^T \cdot \mathbf{e} + 2b)}{\frac{\sum_{t_{M,J} \in T_{\mathcal{M}}} \mu(t_{M,J}, \mathbf{x}^j)(\epsilon_M + \gamma_M)}{\sum_{t_{R,J} \in T_{\mathcal{R}}} \mu(t_{R,J}, \mathbf{x}^j)(\epsilon_R + \gamma_R)} + 1} \tag{22}$$

Defining

$$\Psi(\mathbf{x}^j) := \frac{1 + \psi(\mathbf{x}^j)}{\mathbf{w}^T \cdot e + 2b} \text{ with} \tag{23}$$

$$\psi(\mathbf{x}^j) := \frac{\sum_{t_{M,r} \in T_{\mathcal{M}}} \mu(t_{M,r}|\mathbf{x}^j) \cdot (\epsilon_M + \gamma_M)}{\sum_{t_{R,r} \in T_{\mathcal{R}}} \mu(t_{R,r}|\mathbf{x}^j) \cdot (\epsilon_R + \gamma_R)} \tag{24}$$

Condition (22) can be written as

$$\mathcal{C}^*(x^j) = 1 \iff (\mathbf{w}^T \cdot \mathbf{x}^j + b) > \frac{1}{\Psi(\mathbf{x}^j)} \tag{25}$$

This constraint, derived using Game Theory (Condition (17)), will now be used as a *prior knowledge* constraint in a classification problem, derived from regularized risk minimization proposed by Vapnik (1995).

Considering Condition (17), when a CLASSIFIER'S optimal strategy is to classify a given message at time τ as $f(x_\tau) = sgn(\mathbf{w}^T \cdot \mathbf{x}_\tau^j + b) = +1$, the hyperplane evaluation $(\mathbf{w}^T \cdot \mathbf{x}_\tau^j + b)$ must hold Condition (25).

[6] In the following, for the sake of simplicity, messages at time τ of type $j, \mathbf{x}_\tau^j$, will be considered as a generic message of type $\mathbf{x}^j$.

Given previous considerations and assumptions, using a slack variable ξ to include a soft-margin classification, the constraint presented in Condition (25) can be expressed as Condition (26).

$$y_\tau \left(\mathbf{w}^T \cdot \mathbf{x}_\tau + b\right) \cdot \Psi(x_\tau) \geq (1 - \xi_\tau) \tag{26}$$

This result derived from game-theoretic considerations will be used within a classification framework as will be shown in the following subsection.

5.3 Adversary-aware online SVM: classification model and algorithm

We now incorporate the result derived in the previous section (i.e., Eq. (26)) explicitly into a SVM-based classifier. The CLASSIFIER takes the fact he/she is dealing with rational malicious adversaries explicitly into account, adversaries who strategize optimally over the set of messages they have available. Given that, the classifier computes a best response that indicates which messages to accept. This leads to the following model:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\sum_{a=1}^{A} w_a^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) \cdot \Psi(\mathbf{x}_i) \geq (1 - \xi_i), \forall i \in \{1, \ldots, N\} \\ & \xi_i \geq 0 \; \forall i \in \{1, \ldots, N\} \end{aligned} \tag{27}$$

and its dual

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\sum_{j=1}^{N} y_i \Psi(\mathbf{x}_i) \cdot y_j \Psi(\mathbf{x}_j) \cdot \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to} \quad & \sum_{i=1}^{N} y_i \Psi(\mathbf{x}_i) \cdot \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C_i, \forall i \in \{1, \ldots, N\} \end{aligned} \tag{28}$$

Algorithm 3 presents a novel online learning algorithm, which we call AAO-SVM, with two important characteristics.

- Sequential equilibrium strategies are explicitly incorporated into the determination of the hyperplane parameters used in SVM. This previous knowledge makes the idea relevant that messages do not necessarily reflect the characteristics of a malicious sender, who might be altering content in order to avoid detection.
- Moreover, the CLASSIFIER'S beliefs are updated over time, taking the message history into account, which induces a recalculation of optimal strategies and therefore of the prior knowledge incorporated into the SVM classifier.

Algorithm 3 Adversary-aware online SVM

Input: $\{\mathcal{T}, m, \nu, Gp, C, k, \epsilon_R, \gamma_R, \epsilon_M, \gamma_M\}$
Output: Classification function $h(\mathbf{x}_\tau) = w_\tau^T \cdot \mathbf{x}_\tau + b_\tau, \forall \tau$
Initialize $w_0 := 0, b_0 := 0, \mathcal{S} \leftarrow \emptyset$
for each $(\mathbf{x}_\tau, y_\tau) \in \mathcal{T}$ **do**
  Classify $\mathbf{x}_\tau$ using $f(\mathbf{x}_\tau) = w_{\tau-1}^T \cdot \mathbf{x}_\tau + b_{\tau-1}$
  if $y_\tau \left(w_{\tau-1}^T \cdot \mathbf{x}_\tau + b_{\tau-1}\right) \Psi(\mathbf{x}_\tau) < \nu$ **then**
    Find w', b' with prior knowledge SMO on $\mathcal{S}$, with $w_{\tau-1}$ and $b_{\tau-1}$ as seed hypothesis, and $\Psi(\mathbf{x}_\tau)$
    set $w_\tau := w'$ and $b_\tau := b'$
  end if
  if $|\mathcal{S}| > m$ **then**
    remove oldest example from $\mathcal{S}$
  end if
  if $T \bmod Gp = 0$ **then**
    Initialize $\{c_i\}_{i=1}^k$ = getClusters$(\mathcal{S}, k)$
    $\{I_i\}_{i=1}^k$ = build k information sets, represented by centroids $\{c_i\}_{i=1}^k$.
    $p(t_i) = |\{x | x \text{ is a message in cluster } c_i\}|/|\mathcal{T}_r|, \forall i \in \{1, \ldots, k\}$
    $\mathbf{u}_A \leftarrow [\cdot], \mathbf{u}_C \leftarrow [\cdot]$
    for each $t_i, t_j \in T_\mathcal{A}, d \in \{+1, -1\}$ **do**
      set $\mathbf{u}_C[t_i, t_j, d] \leftarrow U_\mathcal{C}(t_i, c_j, d)$ according to 7,8,9, and 10
      set $\mathbf{u}_A[t_i, t_j, d] \leftarrow U_\mathcal{A}(t_i, c_j, d)$ according to 11,12,13, and 14
    end for
    $\sigma_\mathcal{C}^*, \sigma_\mathcal{A}^*$ = `traceZeroes`$(\mathbf{u}_A, \mathbf{u}_C, \{I_i\}_{i=1}^k, \{p(t_i)\}_{i=1}^k, \{c_i\}_{i=1}^k)$
    update $\Psi(\mathbf{x}_\tau), \forall i \in \mathcal{S}$
  end if
  $\mathcal{S}$.add$(\mathbf{x}_\tau, y_\tau)$
end for

Every Gp periods, all game-theoretical parameters are updated, and type probabilities and strategies are recalculated using logit QRE; see Sect. 3.3. This leads to a particular expression for the prior knowledge constraint (26). During these Gp periods, if the CLASSIFIER'S optimal strategy is not satisfied (Condition (17)), then the hyperplane parameters are updated using the Sequential Minimal Optimization (SMO) algorithm (Platt 1999) over observed messages (i.e., set $\mathcal{S}$). As a consequence $\Psi(\mathbf{x}^i)$ appears as a parameter in model (27), thus maintaining the basic structure of an SVM-based classification model, i.e., a quadratic objective function with linear constraints.

SMO breaks the quadratic optimization problem into a series of smaller quadratic optimization problems, which are then solved analytically, thus avoiding a time-consuming numerical optimization as an inner loop. Minor modifications in the SMO algorithm, such as those that have been explained in previous work on inclusion of *prior knowledge* in SVMs (Wu and Srihari 2004), were included.

6 Application of adversary-aware online SVM for phishing detection

This section emphasizes the potential of the proposed AAO-SVM approach by means of an application for phishing detection, a problem which is outlined in the next subsection. The corresponding game, defined in Sect. 6.2 before Sect. 6.3, shows how

the developed classifiers will be evaluated. The respective results and comparisons to benchmark algorithms will be discussed in Sect. 6.4.

6.1 Phishing classification

Currently, phishing attacks occur everywhere in cyber-space (Zareapoor and Seeja 2015). According to the Anti-Phishing Working Group (APWG; www.apwg.org) "... the number of unique phishing sites detected by the APWG reached an all time high of 63,253 ..." in April 2012; APWG (2012).

Fraud through email messages, *Facebook*, *Twitter*, and even other communication media, such as SMS's, are being used as a primary source to deceive naïve users, affecting global security and the economy.

By the year 2006, many important phishing attacks had been perpetrated against *e-Bay* and *Paypal* customers through the use of fraudulent emails sent by phishers (Bergholz 2009). Here, the basic strategy was to urge users that they had to send their personal information or their accounts were in danger of being closed by the company. Phishing attacks have evolved into many other sophisticated attacks. *Spear-phishing*, *whaling*, *vishing*, among other fraud techniques that are yet to be revealed (Bergholz 2009), are based on and inspired by social engineering methods for making victims believe that they must share personal identifying information with someone.

In this context, traditional content-based characterization of spam messages has proven to be insufficient for phishing detection, since most of the spam email is designed just to inform about some product. In phishing there is a more complex interaction between the message and the receiver, like being led to follow malicious links, to fill in deceptive forms, or to reply with personal information which is needed for the goal of the message to succeed. Also, there is a clear difference among many phishing techniques, where the two main message categories are *deceptive phishing* and *malware phishing*. While *malware phishing* has been used to spread malicious software installed on the victims' machines, *deceptive phishing* can be categorized as follows (Bergholz et al. (2010)):

- Social engineering,
- Mimicry,
- Email spoofing,
- URL hiding,
- Invisible content, and
- Image content.

For each one of these categories, specific feature extraction techniques have been proposed by Bergholz et al. (2010) to help phishing classifiers to use the right characterization of these malicious messages.

In our application we show how a novel approach for phishing feature extraction, closely related to the messages' semantic information by using text mining and latent semantic analysis, could enhance the performance of the posterior AAO-SVM for email classification.

We used an English language *phishing* email corpus that was built using Jose Nazario's *phishing* corpus (Nazario (2007)) and the Spamassassin *Ham* collection.

The *phishing* corpus[7] consists of 4450 emails retrieved manually from November 27, 2004 to August 7, 2007. The *Ham*[8] corpus was built using the Spamassassin collection from the Apache SpamAssassin Project,[9] based on a collection of 6951 *Ham* email messages. Both phishing and ham messages are available in UNIX mbox format.

6.2 Definition of the game: strategies and types

In this section, the main game components are defined for the phishing application beginning with how phishing strategies are identified. Using text mining we extract features that characterize email messages and hence represent the respective strategies. Then, ADVERSARY types are determined by clustering all messages and thus grouping similar messages together.

6.2.1 Determining adversarial strategies

Phishing messages are characterized using latent semantic analysis and keyword finding techniques (L'Huillier et al. 2010). The message processing is based on a combination and selection of features generated by text mining algorithms based on probabilistic topic modeling such as latent Dirichlet allocation (LDA) (Blei et al. 2003; Bíró et al. 2009), and singular value decomposition (SVD) (Deerwester et al. 1990) together with keyword finding techniques (Velasquez et al. 2005).

The proposed characterization and feature extraction from messages is described in more detail in Appendix 1, and its evaluation is presented in (L'Huillier et al. 2010). We used the previously mentioned email corpus as the input dataset. Three feature extraction methodologies were applied on this corpus in order to characterize the messages. The obtained data set was used to train all classification algorithms.

6.2.2 Determining adversarial types

The different adversarial types are determined by clustering all emails (phishing and ham messages) from the initial training set according to their features. The resulting clusters are the *information sets* introduced in Sect. 3.1 which contain messages that are indistinguishable for the CLASSIFIER. We propose using k-means clustering where k indicates the number of different types, but any other clustering method could be used as well. The number of clusters k should be determined context dependently and considering the structure of the initial training data set where cluster validity measures (such as e.g., Davies-Bouldin (Halkidi et al. 2001)) provide aid in choosing an adequate cluster number.

For each message $\mathbf{x}$, its type t is determined by

$$t = \arg\min_i \delta(\mathbf{x}, c_i) \tag{29}$$

[7] Available at http://bit.ly/jnazariophishing [online: access 01/01/2013]

[8] "Ham" is the name used to describe regular messages that are neither spam nor phishing.

[9] available at http://spamassassin.apache.org/publiccorpus/ [online: accessed 01/01/2013]

where c_i is the centroid of cluster i, and function $\delta : \mathbb{R}^A \times \mathbb{R}^A \rightarrow \mathbb{R}$ represents the distance between two vectors of dimension A. Given that messages are characterized using binary features, the distance function used is the Hamming distance; Hamming (1950).

The fact that types are determined over the whole phishing and ham corpus is relevant to the game modeling. In this way, it is possible to determine which messages are closest to a malicious type or a regular type, affecting NATURE's probabilities of choosing a given type, and hence the beliefs calculated for the CLASSIFIER.

6.3 CLASSIFIERS' evaluation

For the proposed game-theoretical models, the sequential equilibria were approximated using the `Gambit` software tool (`gambit-logit`) (McKelvey et al. 2010). The S-QRE and P-QRE approaches were implemented in `Perl` programming language, whereas the AAO-SVM classifier was implemented in `C++` extending Sculley's Online SVM development (Sculley and Wachman 2007). As benchmark techniques for the previously mentioned game-theoretical classification models (S-QRE, P-QRE, and AAO-SVM), we considered the relaxed online SVM (RO-SVM) proposed by Sculley and Wachman (2007), as well as a version of the Repeated Game Adversarial Classifier proposed by Dalvi et al. (2004), an incremental version of Naïve Bayes. RO-SVM was deployed from its original implementation, and the incremental naïve Bayes classifier was implemented in the Weka open source data mining tool (Hall et al. 2009) using the `NaiveBayesUpdateableBayes` version of the algorithm. The repeated game adversarial classifier was coded in `Python`. More details about the algorithm are presented in Sect. 6.3.1.

The performances of the algorithms were evaluated by calculating the evaluation criteria mentioned in Sect. 6.3.2, i.e., every 500 observations of the respective test sets. Furthermore, batch learning algorithms were trained on a small subset of the corpus (30 %) to test the database dynamics in terms of ADVERSARIAL changes in the incoming email messages, and then tested over the remaining part of the data (70 %). This evaluation was executed under the same instance order under which the online algorithms were evaluated. The error was also calculated every 500 observations. Section 6.3.3 presents our methodology for sensitivity analysis and robust testing.

6.3.1 Repeated game for adversarial classification

As an extension of the framework presented in Sect. 2.1, the Repeated Game for Adversarial Classification (RGAC) considers that both the ADVERSARY and the CLASSIFIER generate representations of each other that evolve after every interaction. Both players solve an intrinsic optimization problem within a game-theoretical framework to find their optimal strategies. Assuming that the CLASSIFIER is a naïve Bayes algorithm, according to Dalvi et al. (2004) the framework can be summarized by the following steps:

1. The ADVERSARY, trying to guess the CLASSIFIER, trains a naïve Bayes algorithm based on the outputs of the CLASSIFIER, which is shared at the end of the previous

Algorithm 4 Adversarial strategy (repeated games)

Input: Instance $\mathbf{x}$
Output: Adversarial strategy $\mathcal{AD}(\mathbf{x})$
$W \leftarrow gap(\mathbf{x})$
$(MinCost, MinList) \leftarrow$ FINDMCC(n, W)
if $NB(\mathbf{x}) = +1$ and $MinCost < \Delta U_A$ **then**
  for all $(i, x'_i) \in MinList$ **do**
    $new\mathbf{x}_i \leftarrow x'_i$
  end for
  return $new\mathbf{x}$
else
  return $\mathbf{x}$
end if

Algorithm 5 Adversary Aware naïve Bayes (Repeated Games)

Input: Instance $\mathbf{x}'$
Output: Classification $\mathcal{C}(\mathbf{x}')$
$P^{-}_{\mathbf{x}'} \leftarrow \hat{P}(-1) \prod_i \hat{P}(x'_i| - 1)$
$P^{+}_{\mathbf{x}'} \leftarrow \hat{P}(+1) \hat{P}_A(\mathbf{x}'| + 1)$
$U(+1|\mathbf{x}') \leftarrow P^{+}_{\mathbf{x}'} U_C(+1, +1) + P^{-1}_{\mathbf{x}'} U_C(-1, +1)$
$U(-1|\mathbf{x}') \leftarrow P^{+}_{\mathbf{x}'} U_C(-1, +1) + P^{-}_{\mathbf{x}'} U_C(-1, -1)$
if $U(+1|\mathbf{x}') > U(-1|\mathbf{x}')$ **then**
  return $+1$
else
  return -1
end if

round. The ADVERSARY's guess for the CLASSIFIER in round $i - 1$ is labeled NB_{i-1}.

2. Then, using NB_{i-1}, an optimal adversarial strategy $\mathcal{AD}_i$ (Algorithm 4) is used to determine the data for deceiving the CLASSIFIER at round i ($\mathcal{T}_i = \mathcal{AD}_i(\mathcal{T}_{i-1})$).
3. The CLASSIFIER uses Algorithm 5 based on NB_{i-1} to classify $\mathcal{T}_i$ (i.e., $\mathcal{Y}_i = \mathcal{C}_i(\mathcal{T}_i)$). It is important to note that the new naïve Bayes model NB_i is trained on $(\mathcal{T}_i, \mathcal{Y}_i)$, generating the new classifier to be considered as input for the next iteration of the repeated game.

FINDMCC(i, w) is a function that computes the minimum cost needed to change the log-odds ratio of $\mathbf{x}$ ($LO_c(\mathbf{x})$, Eq. 31) by W using only the first i features. This method returns the pair $(MinCost, MinList)$ where $MinCost$ is the minimum cost and $MinList$ is a feature-value mapping that represents all changes that are to be made to instance $\mathbf{x}$. To compute the optimal adversarial strategy, as shown in Algorithm 4, the goal is to compute the minimum cost that changing n features of $\mathbf{x}$ will not change the log-odds of $\mathbf{x}$ in more than $gap(\mathbf{x})$ (Eq. 30),

$$gap(\mathbf{x}) = LO_c(\mathbf{x}) - \frac{U_C(-1,-1) - U_C(+1,-1)}{U_C(+1,+1) - U_C(-1,+1)} \tag{30}$$

$$LO_c(\mathbf{x}) = log\left(\frac{P(+1)}{P(-1)}\right) + \sum_{x_i \in \mathcal{X}_C} \log\left(\frac{P(x_i|+1)}{P(x_i|-1)}\right) \tag{31}$$

and

$$LO_c(x_i) = \log\left(\frac{P(x_i|+1)}{P(x_i|-1)}\right) \tag{32}$$

where, $\mathcal{X}_C$ is the set of features of $\mathbf{x}$ considered and $U_C(y_C, y)$ is the CLASSIFIER's utility when identifying as y_C an instance of class y. It is important to notice that the log-odds ratio $LO_c(x_i)$ is also considered to be the contribution of the ith attribute of $\mathbf{x}$ (i.e., x_i) to the CLASSIFIER.

Finally, in Algorithm 4, $\Delta U_A = U_A(-1,+1) - U_A(+1,+1)$ is the ADVERSARY's utility gain, where $U_A(y_C, y)$ is the ADVERSARY's utility when the CLASSIFIER identifies as y_C an instance of class y.

Algorithm 5 is specifically adversary-aware as it considers as input $\mathcal{AD}(\mathbf{x}) = \mathbf{x}'$ which is the ADVERSARY's optimal strategy.

$\hat{P}(\cdot)$ is used to denote the probability estimated using the training data $\mathcal{T}$. Similarly, $\hat{P}_A(\mathbf{x}'|+1)$ is obtained by the approximation in Eq. 33.[10] Considering that $\mathbf{x}_{[i=x_i]}$ represents the instance identical to $\mathbf{x}$ except its ith feature changed to x_i, then,

$$\hat{P}_A(\mathbf{x}'|+1) \simeq \left(\sum_{(i,x_i)\in GV} P(\mathbf{x}'_{[i=x_i]}|+1)\right) + I(\mathbf{x}')\hat{P}(\mathbf{x}'|+1) \tag{33}$$

where $I(\mathbf{x}') = 1$ if $NB((\mathbf{x}')) = -1$ and $I(\mathbf{x}') = 0$ otherwise. Let $\mathbf{x}_A$ be an instance labeled as $+1$, $\mathbf{x}' = \mathcal{AD}(\mathbf{x})$, $\mathcal{X}_A(\mathbf{x}') = \{\mathbf{x} : \mathbf{x}' = \mathcal{AD}(\mathbf{x})\}$, and $\mathcal{X}'_A(\mathbf{x}') = \mathcal{X}_A(\mathbf{x}')\backslash\{\mathbf{x}'\}$. It can be shown that, $\forall i$:

$$(\mathbf{x}_A)_i \neq x'_i \Rightarrow gap(\mathbf{x}') + LO_C((\mathbf{x}_A)_i) - LO_C(x'_i) > 0 \tag{34}$$

Let FV be defined as a set of feature-value pairs that satisfy Eq. 34, then $GV \subset FV$ will be defined as all feature-value pairs that $\mathbf{x}'_{i=x_i} \in \mathcal{X}'_A(\mathbf{x}')$.

To evaluate this algorithm as a benchmark, the values considered for the utility functions are $(U_A(+1,-1), U_A(+1,+1), U_A(-1,-1), U_A(-1,+1)) = (0, 0, 0, 20)$ and $(U_C(+1,-1), U_C(+1,+1), U_C(-1,-1), U_C(-1,+1)) = (1, -10, -1, 1)$ for the ADVERSARY and the CLASSIFIER, respectively.[11]

For the evaluation of such an algorithm, the "Add Words" adversarial strategy was originally considered by Dalvi et al. (2004). This strategy consists of adding words that were not used before to the message. However, in order to keep a consistent adversarial behavior across all the experiments, a different adversarial strategy is considered. Without affecting the framework presented in this section, the ADVERSARY is able to either keep a given message or modify it to a message type extracted from the data collected at that period of time (as discussed in Sects. 4 and 5). In such scenario, the $MinList$ from Algorithm 4 is determined by all the feature-value mapping that

[10] This expression represents a lower bound for $\hat{P}_A(x'|+1)$. It has been shown to have a good performance in practice (Dalvi et al. 2004).

[11] These are the same values considered in Dalvi et al. (2004) which showed less variability in the overall utility functions of both players.

represents all changes to be made to instance $\mathbf{x}$ to transform it into one of the available types of messages in $T_{\mathcal{A}}$.

6.3.2 Criteria for classifiers' performance evaluation

This binary classification task can be described using four possible outcomes: Correctly classified phishing messages, or True Positives (TP); correctly classified ham messages, or True Negatives (TN); wrongly classified ham messages as phishing, or False Positives (FP); and wrongly classified phishing messages as ham, or False Negatives (FN). Using these values, the Accuracy, False Positive rate, False Negative rate, Precision, Recall, and F-measure were calculated as shown in the following formulas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{35}$$

$$FP - Rate = \frac{FP}{FP + TN} \tag{36}$$

$$FN - Rate = \frac{FN}{FN + TP} \tag{37}$$

$$Precision = \frac{TP}{TP + FP} \tag{38}$$

$$Recall = \frac{TP}{TP + FN} \tag{39}$$

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{40}$$

$$\text{Error}(t) = 1 - Accuracy(t) \tag{41}$$

where $Accuracy(t)$ is the Accuracy after t messages evaluated for each algorithm. This measure is used to determine the dynamics of the incremental algorithms.

6.3.3 Sensitivity analysis and robust testing

To determine the parameters of the proposed classification algorithms, performance measures of different scenarios were calculated. The S-QRE and P-QRE algorithms were tested for three different cost scenarios:

1. Cost Scenario 1: Payoffs are defined with equal values, in order to represent a zero-sum game.
2. Cost Scenario 2 considers the same payoff structure presented in the numerical example introduced in Sect. 3.1.1 adapted with equations introduced in Sect. 4.1.
3. Cost Scenario 3 is similar to Scenario 2, with the exception that classification errors have a more severe penalization.

Furthermore, for the S-QRE algorithm, a parameter p representing the training/test ratio was evaluated, with values for $p \in \{0.1, 0.3, 0.5, 0.7\}$. For P-QRE a parameter w was considered which determines the window frame (i.e., number of observations)

for the re-evaluation of the strategic interaction between agents with values for $w \in \{500; 1000; 3000\}$.

Finally, we propose the following five scenarios to analyze sensitivity of the utility functions used in the proposed AAO-SVM classifier. These analyses are performed by varying the parameters γ_R, γ_M, ϵ_R, and ϵ_M presented in Table 1 in Sect. 5.1.

1. The first scenario represents the situation in which each case of correct classification provides the same payoff (i.e., $\epsilon_R = \epsilon_M$) and each misclassification leads to the same (negative) payoff (i.e., $\gamma_R = \gamma_M$).
2. The second scenario is based on the idea that correctly classifying a regular message is more important for the CLASSIFIER than correctly classifying a malicious one. Furthermore, it is assumed that incorrectly classifying a regular message as malicious is even more costly than misclassifying a malicious message. To achieve this we use $\frac{\gamma_R}{\gamma_M} = 4$ and $\frac{\epsilon_R}{\epsilon_M} = 2$.
3. The third scenario—inversely to the second one - assumes that the correct classification of malicious messages is more important for the CLASSIFIER than correctly classifying regular ones. Along the same line, incorrectly classifying a malicious message as regular is more costly than misclassifying a regular message. This is attained by $\frac{\gamma_R}{\gamma_M} = 1/4$ and $\frac{\epsilon_R}{\epsilon_M} = 1/2$.
4. The fourth scenario assumes that misclassifying ham messages is extremely costly in comparison with misclassifying phishing messages. This is $\frac{\gamma_R}{\gamma_M} = 100$. Classifying phishing messages correctly reports very high benefits, represented by $\frac{\epsilon_R}{\epsilon_M} = 1/100$.
5. The fifth case is the opposite of the latter, in which classifying ham messages correctly is better rewarded, and misclassifying phishing messages is much more costly than misclassifying ham messages. This is achieved by $\frac{\gamma_R}{\gamma_M} = 1/100$ and $\frac{\epsilon_R}{\epsilon_M} = 100$.

The ranges of the proposed values were determined by preliminary sensitivity testing over the utility function, and the classification performance of the proposed algorithms. In this case, small variations of these parameters ($\times 100$) showed better performance than high variations on parameters ($\times 1000$). Therefore, we selected small ranges for parameters in this work.

6.4 Results and discussion

In Sect. 6.4.1, we first present results provided by the different classification algorithms applied to the data sets introduced in Sect. 6.1. Subsequently, in Sect. 6.4.2 we show clearly which kinds of additional insights beyond improved classification performance we could expect from the proposed combination of Game Theory models with advanced Data Mining techniques. Finally, in Sect. 6.4.3, we perform sensitivity analysis and robust testing in which different parameter settings were used within the presented adversary-aware classification algorithms. How to determine the ADVERSARIES' strategies and types using clustering techniques has already been mentioned in Sects. 6.2.1 and 6.2.2, respectively. A more detailed study of these issues would go beyond the scope of this paper.

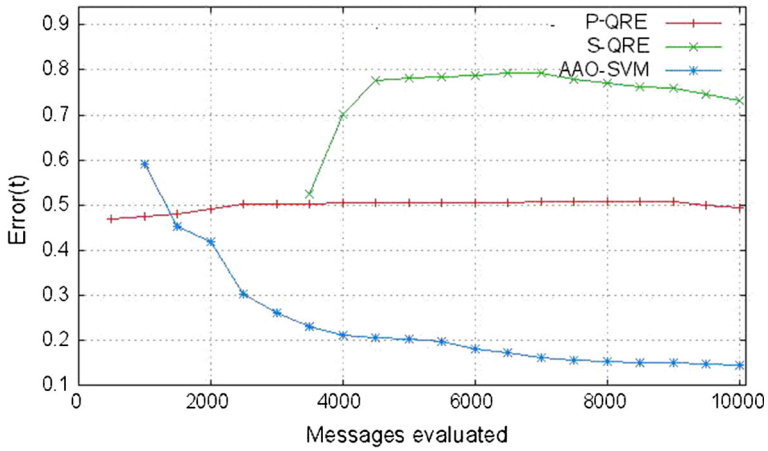

Fig. 3 Error calculated each 500 messages presented in the online evaluation setup for S-QRE, P-QRE, and AAO-SVM

6.4.1 Results obtained by classification algorithms

We first present the results obtained by applying QRE-based algorithms (S-QRE, P-QRE, and AAO-SVM). Then we turn our attention to benchmark algorithms, which in this work are RO-SVM, RGAC, and incremental naïve Bayes. All methods were evaluated according to the performance measures introduced in Sect. 6.3.2.

QRE-based learning algorithms

Figure 3 exhibits the results obtained by S-QRE, P-QRE, and AAO-SVM. These algorithms were evaluated estimating, first, all parameters for both payoffs and structural properties of the algorithms. P-QRE presents an error which is greater than the respective values for S-QRE and AAO-SVM with the latter being the most competitive algorithm.

Online learning algorithms

Figure 4 presents the complete evaluation sequence on messages presented from the dataset. In general, the performance for all incremental algorithms improves as the messages are presented, as exhibited in Fig. 4. However, the rate at which algorithms are improved, differs showing better performance for both AAO-SVM and RO-SVM, whose results outperformed the incremental version of naïve Bayes. Also, all algorithms outperform RGAC.

Tables 2 and 3 show the classification performance of the proposed algorithms using the performance measures introduced in Sect. 6.3.2. As can be seen, the proposed AAO-SVM outperforms all benchmark algorithms, with RO-SVM achieving only slightly worse results. Algorithms were evaluated incrementally, repeating the experiment 10 times and then computing the average and standard deviation of the respective performance measures.

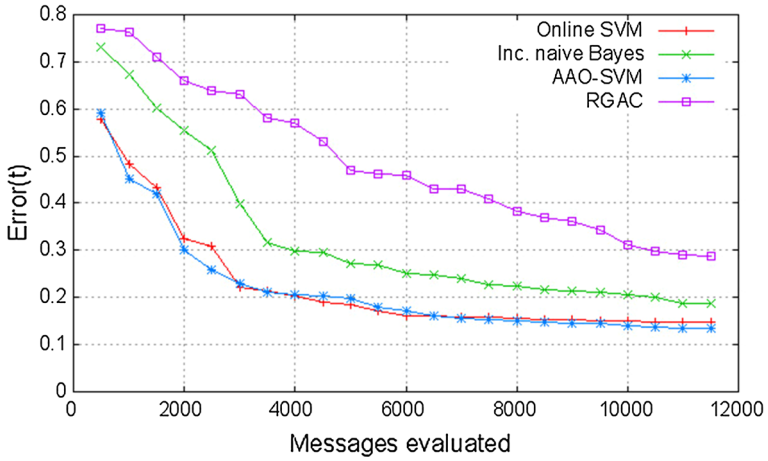

Fig. 4 Error calculated each 500 messages for AAO-SVM, using RO-SVM, RGAC, and incremental naïve Bayes classification algorithms as a benchmark

Table 2 Results for Accuracy, FP-Rate, and FN-Rate, represented by their mean values and standard deviation for the proposed adversary-aware classifiers (S-QRE, P-QRE, AAO-SVM)

Model	Accuracy	FP-Rate	FN-Rate
S-QRE	32.12 % (±6 %)	63.75 % (±8 %)	74.06 % (±6 %)
P-QRE	42.09 % (±12 %)	54.17 % (±8 %)	62.01 % (±13 %)
AAO-SVM	**86.63 %** (±5 %)	**9.43 %** (±3 %)	16.35 % (±4 %)
RO-SVM	85.52 % (±3 %)	10.19 % (±5 %)	17.87 % (±3 %)
Inc. NB	80.69 % (±1 %)	25.19 % (±2 %)	**10.11 %** (±2 %)
RGAC	71.10 % (±3 %)	25.17 % (±4 %)	34.69 % (±4 %)

Best result in bold
RGAC repeated game for adversarial classification, *Inc. NB* incremental Naïve Bayes, *RO-SVM* relaxed online-SVM as benchmark algorithms

Table 3 Results for Precision, Recall, and F-measure, represented by their mean values and standard deviation for the proposed adversary-aware classifiers (S-QRE, P-QRE, AAO-SVM)

Model	Precision	Recall	F-measure
S-QRE	25.94 % (±11 %)	21.36 % (±12 %)	0.2343 (±0.0171)
P-QRE	37.99 % (±5 %)	39.08 % (±5 %)	0.3853 (±0.1033)
AAO-SVM	**83.65 %** (±9%)	**92.12 %** (±4 %)	**0.8768** (±0.0111)
RO-SVM	82.13 % (±6 %)	91.05 % (±3 %)	0.8636 (±0.0091)
Inc. NB	69.55 % (±1 %)	92.03 % (±2 %)	0.7923 (±0.0044)
RGAC	62.41 % (±3 %)	65.30 % (±2 %)	0.6382 (±0.0031)

Best result in bold
RGAC repeated game for adversarial classification, *Inc. NB* incremental Naïve Bayes, *RO-SVM* relaxed online-SVM as benchmark algorithms

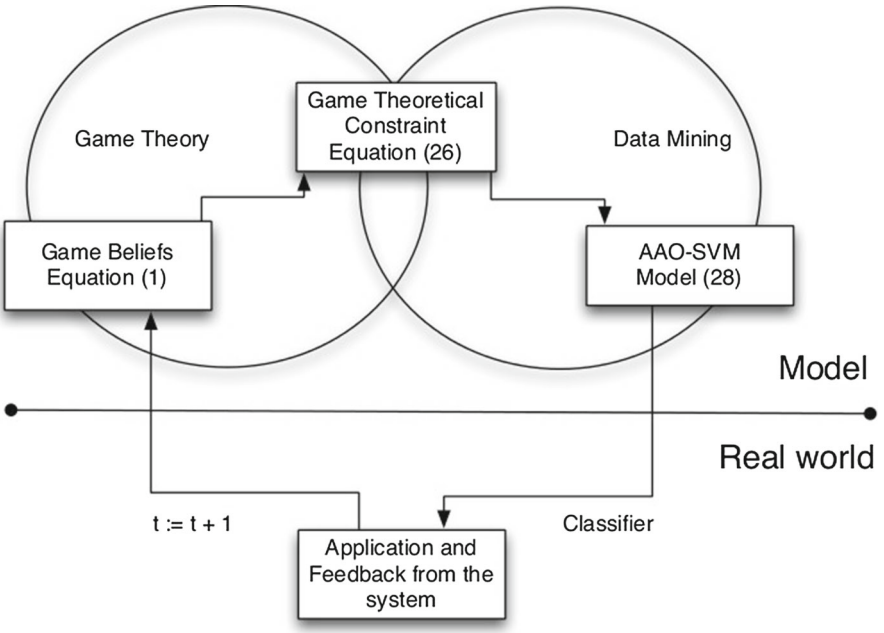

Fig. 5 Interaction between Game Theory and Data Mining

As can be seen, AAO-SVM provides slightly better results than RO-SVM if only classification performance is considered.

6.4.2 Results obtained by combining game theory models with classification algorithms

Explicitly modeling the interaction between the players in adversarial classification, combining approaches from Game Theory with advanced Data Mining techniques, provides results which go beyond a mere improvement in classification performance, by also providing additional insights into the respective phenomena as will be shown clearly in this subsection.

Figure 5 depicts the interaction between the proposed approaches from Game Theory and Data Mining showing one iteration and emphasizing the role of parameter Ψ (used in Condition 25) which links the basic knowledge obtained from game-theoretic modeling to classification models using SVM.

The cycle presented in Fig. 5 starts with the belief $\mu(t|\mathbf{x}^j)$ which is the probability that a message $\mathbf{x}^j$ will be classified as t. Using these beliefs and the respective misclassification costs, we determine $\Psi(\mathbf{x}^j)$ which is then used in model 27 adding explicitly knowledge derived by Game Theory to the respective classification approach. The subsequent application to the respective domain (here: Phishing detection) modifies the beliefs in the following iteration.

Next, we analyze the development of value $\Psi(\mathbf{x}^j)$ as introduced in formula (23) during the iterations of one particular experiment. Fig. 6 shows this value over the

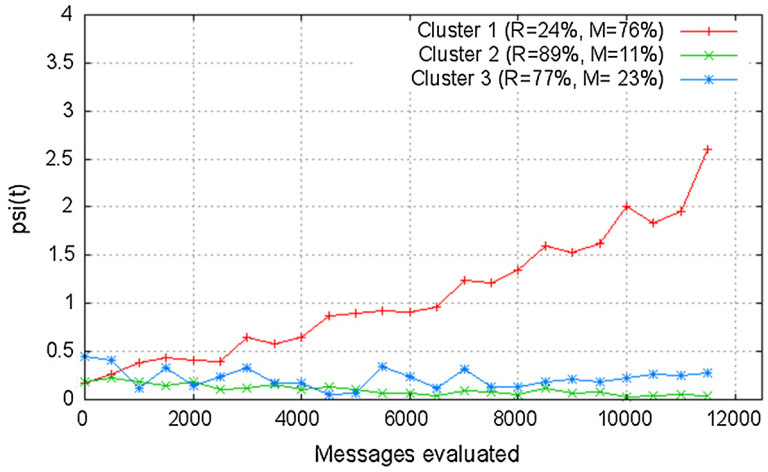

Fig. 6 Ψ in function of the messages evaluated considering 3 clusters for Regular and Malicious types

number of iterations for three classes found by clustering as explained in Sect. 6.2.2. As can be seen, the proposed procedure identifies the class which contains most of the M-type messages, since this is the class where value $\Psi(\mathbf{x}^j)$ increases whereas the other two classes maintain a low value for the incoming messages.

This result suggests that Game Theory allows obtaining an additional parameter which is used within the classification model, but which, by itself, also provides important information.

6.4.3 Sensitivity analysis and robust testing

In the following, a sensitivity analysis and robust testing for each adversary-aware classification algorithm (S-QRE, P-QRE, AAO-SVM) are presented and discussed. All of the following evaluations are based on the experimental setup described in Sect. 6.3.3.

To test the robustness of the classifier **S-QRE**, the percentage of objects used for training/test datasets p was analyzed using values $p \in \{0.1, 0.3, 0.5, 0.7\}$. Best results were obtained for $p = 0.3$, whereas $p = 0.7$ led to the worst results; see Fig. 7.

We also tested sensitivity regarding the utility function's parameters used in S-QRE as is shown in Fig. 8. Best results were obtained for Cost Scenario 1, where both CLASSIFIER and ADVERSARY have a balanced distribution of costs (see Sect. 6.3.3). In this case, the classification error for Scenarios 1 and 3 are similar, and Scenario 2 presents a higher error. This was evaluated using 30 % of the dataset for training, and 70 % for testing and calculating the error ($p = 0.3$).

In the case of **P-QRE**, we analyzed the sensitivity of parameter w which is the size of the window used to update the game-theoretical parameters of the proposed model (utility functions and agents' strategies). Three different window sizes were evaluated ($w \in \{500; 1000; 3000\}$); see Fig. 9. The lowest error was obtained for the wider window strategy ($w = 3000$), but results for alternative values of w are

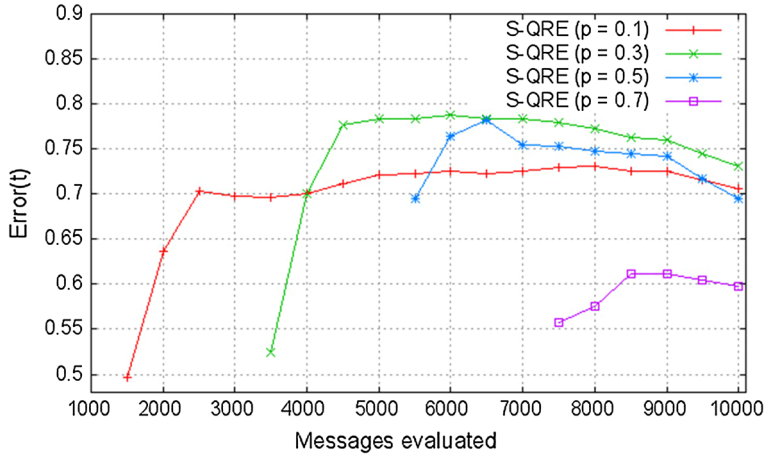

Fig. 7 Error for the S-QRE classifier using $p \in \{0.1, 0.3, 0.5, 0.7\}$

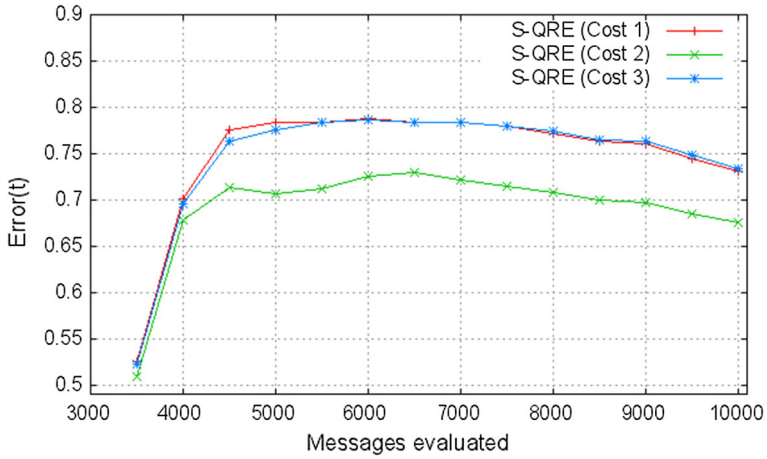

Fig. 8 Error for the S-QRE classifier based on Cost Scenarios 1, 2, and 3

very similar, indicating low sensitivity regarding this parameter. In terms of the cost scenarios, Scenario 1 shows better classification performance than Scenarios 2 and 3 for P-QRE; see Fig. 10.

Finally, sensitivity analysis for the **AAO-SVM** algorithm was performed considering the relationship between different parameters presented in the CLASSIFIER'S and ADVERSARY'S utility functions. Figure 11 exhibits the evaluation of the **AAO-SVM** algorithm for $Gp \in \{100; 500; 1000\}$. Results show that $Gp = 500$ leads to the least error. For $Gp = 1000$, the classifier behaves only slightly worse in terms of the error, indicating low sensitivity regarding parameter Gp.

Figure 12 shows the error of classified messages using the five scenarios introduced in Sect. 6.3.3. In the first place, a linearly decreasing error over time can be identified

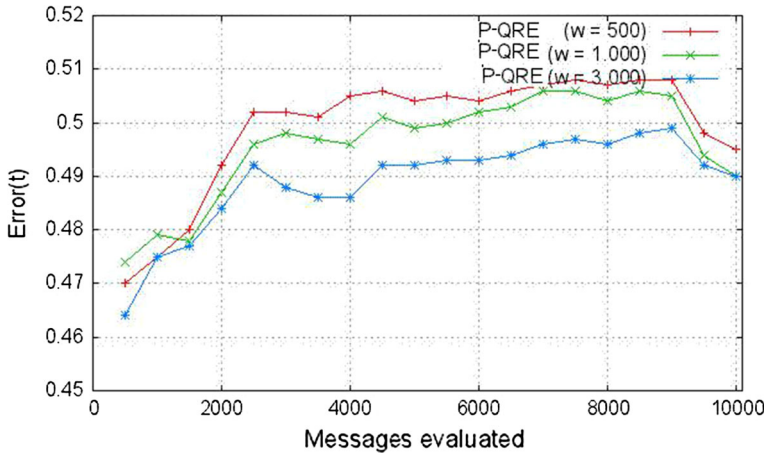

Fig. 9 Error for the P-QRE classifier using $w \in \{500, 1000, 3000\}$

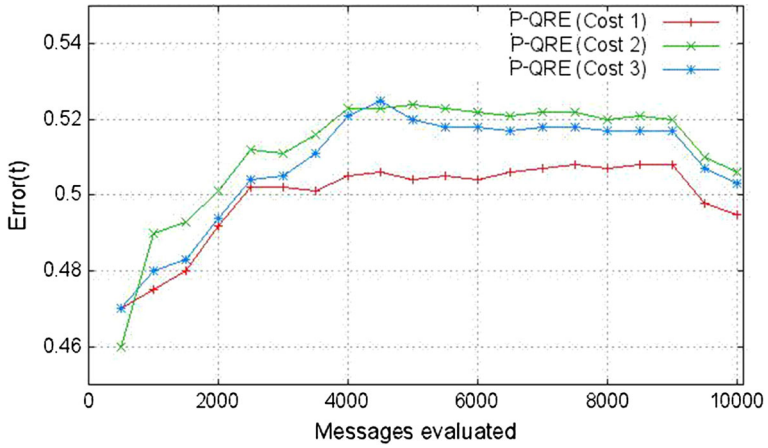

Fig. 10 Error for the P-QRE classifier based on Cost Scenarios 1, 2, and 3

in Scenarios 5 and 3 whereas Scenarios 1, 2, and 4 present faster adaptation. The best overall results were reached for scenario 2.

7 Summary, future work, and conclusions

In this section, we summarize first the most relevant results presented in this paper, and then propose possible lines for future research. The subsequent Sect. 7.2 concludes this paper by identifying our main contributions.

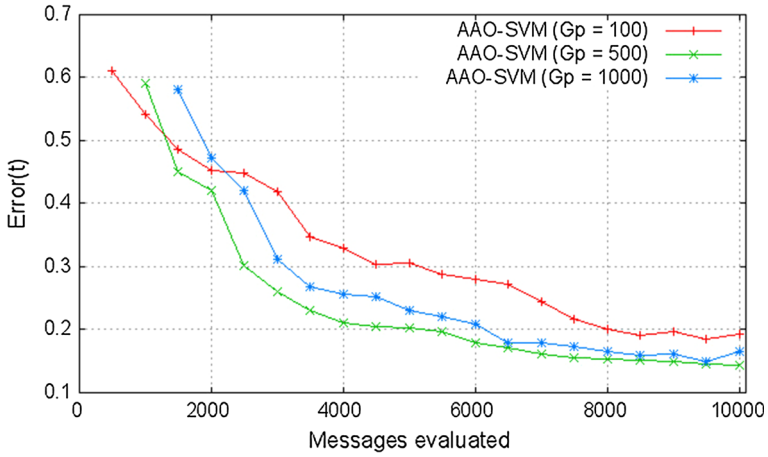

Fig. 11 Error calculated for the AAO-SVM classifier for $Gp \in \{100; 500; 1000\}$

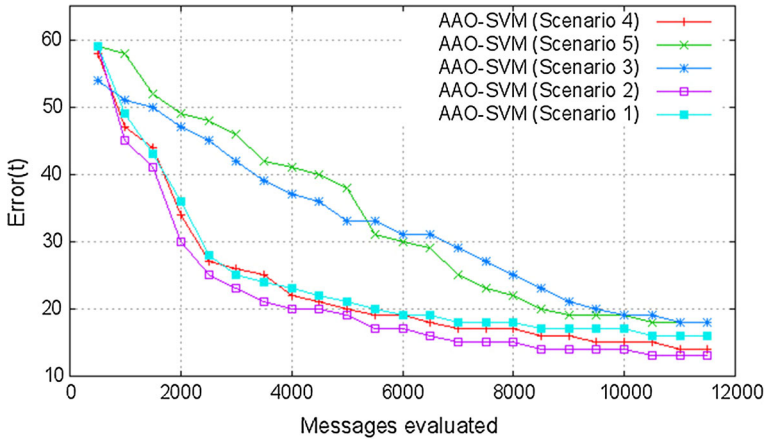

Fig. 12 Error calculated for the AAO-SVM classifier for scenarios in Sect. 6.3.3

7.1 Summary and future work

The following list provides a short summary of each single achievement followed by ideas for related future research.

1. We presented various extensions to the Adversarial Classification framework, considering **dynamic games of incomplete information (signaling games)** as a new approach to having CLASSIFIERS improve their performance in adversarial environments using game-theoretic approaches. Interesting results were obtained for the proposed S-QRE and P-QRE classifiers based on some assumptions on ADVERSARY strategies, the utility function modeling for both agents, and the experimental setup related to the database processing.

Future work regarding game-theoretical assumptions such as the QRE promises better approximations in various game modeling scenarios. In our current approach we use QRE to approximate the agents' strategies. Further considerations and refinements on incomplete information games could lead to better approximations of game modeling scenarios.

2. **Feature determination** is a key component in identifying strategies and types of ADVERSARIES for the proposed game-theoretical framework. Whereas there might be some general rules, we believe that feature determination (i.e., selection and extraction) should be done context-dependently given a particular adversarial classification task. In the case of the application presented for phishing fraud detection, we proposed a holistic view on the respective email messages using different kinds of features. On the one hand, we characterize each email by structural properties. On the other hand, text mining techniques are used to extract text-related features from the respective email messages. Results showed that the proposed feature extraction and selection techniques are highly competitive with previously presented work on feature extraction.
   Future work should be oriented towards considering a mixture of previous and present feature extraction techniques. This could result in estimating a better strategy space for the ADVERSARY, and therefore the ADVERSARY types. This is an important topic that directly affects the definition of the signaling game between the ADVERSARY and the CLASSIFIER, and thus the CLASSIFIER'S performance.
3. The proposed game-theoretic setting requires the identification of so-called *information sets* which are sets of indistinguishable messages. In this work we propose using clustering approaches in order to determine such information sets. In the application presented for phishing fraud detection we applied standard k-means and determined the cluster number using the Davis–Bouldin index.
   In *future work* the identification of such information sets can be improved, for example by using more sophisticated clustering techniques. Interesting alternatives to the proposed k-means clustering would be to add uncertainty modeling as well as dynamic elements to the clustering process since especially adversarial environments may be characterized by uncertain and dynamic behavior. Adequate techniques to identify the respective clusters have been presented e.g., by Crespo and Weber (2005) and Peters et al. (2012, 2013) among others. Another related issue is the game's drift concept. In this work, the ADVERSARY'S strategies are observed from a given probability distribution defined for the observed data, which is assumed to be modified intentionally by a centralized adversarial agent. An experimental setup for the CLASSIFIER for showing the impact related to the inclusion of new ADVERSARY strategies within an already defined set of strategies (or features) might move us further in this direction.
4. The main contribution of this paper is the proposed AAO-SVM Classifier. It considers a signaling game where beliefs, mixed strategies, and probabilities for the message types are updated and incorporated into the optimization problem via a game-theoretic parameter, as new messages are presented. This enables the classifier to change the margin error dynamically as the game evolves, including an embedded awareness of the adversarial environment. More specifically, this is considered in the misclassification constraint of the optimization problem for the

SVM algorithm, given the utility function modeling for the CLASSIFIER. Results obtained show a promising interaction with the data set for the proposed classifiers over benchmark algorithms used in this experimental setup and previous work. In our application the proposed AAO-SVM classifier outperformed the plain online SVM algorithm.
Future work on several possible extensions would allow enriching the proposed AAO-SVM model.

- The ADVERSARIES can adapt their behavior strategically using a strategy finding algorithm (e.g., Adversarial Learning, proposed in Lowd and Meek (2005)). Here, data mining issues related to the adversarial learning problem must be considered, which to the best of our knowledge, have been studied only in complete information game modeling to date.
- The ADVERSARY, whose only strategic reaction is based on utility function maximization assumptions, must deal with multiple strategic CLASSIFIERS in an incomplete information interaction. To date, only complete information modeling has been considered for a strategic multi-classifier system (Biggio et al. 2009); incomplete strategic information interaction is still an open question in multi-classifier systems.
- Another interesting future development would be the case where one strategic CLASSIFIER must interact with multiple strategic ADVERSARIES. Here, several complexities can be associated with the game modeling, for which the assumption that there is one central strategic ADVERSARY controlled by NATURE does not hold. Given this, further revision into Game Theory modeling for incomplete information games should be considered.
- Finally, the natural extension of all previous problems would occur when all agents, without any assumptions on a central agent for both ADVERSARY as well as CLASSIFIER, strategically modify their behavior using strategy finding algorithms for an incomplete information interaction.

5. The **applications on phishing detection** connect our developments to a real-world problem and underline the strengths of the proposed models.
Future work on various applications for adversarial classification, which could vary from cyber-security to fraud detection, and from crime analytics to privacy-preserving systems, would provide more insights into the potential of the models presented and could indicate further necessary developments.

7.2 Conclusions

The contribution of this paper is threefold:

- First, we define a game-theoretical framework that is proposed to include adversarial interaction between two agents in an adversarial classification game.
- Second, we show how to integrate the findings on strategic interactions provided by Game Theory into classical Data Mining models, in particular SVM, arriving at the proposed AO-SVM. This idea offers a huge potential for creating hybrid models that take "the best of both worlds" explicitly into account: the solid basis

of theoretically well-formulated behavioral models, and powerful state-of-the-art Data Mining techniques.
– Third, the phishing classification application shows the use of the proposed framework in a real-world context, and its potential for improving existing approaches for adversarial classification. The proposed AAO-SVM outperformed Relaxed Online Support Vector Machines slightly, and provided additional insights regarding the behavior of the agents.

The first two items are generic and could be used in any kind of adversarial classification. The detection of phishing messages presented underscores the potential of our developments in a real-world application.

These contributions, together with the above mentioned suggestions for future work, are the results of our research so far and, at the same time, a rich starting point for future work in the area of adversarial classification using combinations of Game Theory and Data Mining.

Acknowledgements Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FBO16; www.isci.cl), the Chilean Anillo project ACT87 "Quantitative Methods in Security" (www.ceamos.cl), and the Master's Degree program in Operations Management at the University of Chile is gratefully acknowledged. The third author acknowledges support from Fondecyt project 1140831.

Appendix 1: Text mining for feature extraction

This appendix contains methods from text mining which have been used to extract useful features from the emails we analyzed in our application as presented in Sect. 6.

First, documents are tokenized in order to extract every word used in the message. Both a stopword removal process, and stemming of the words in the messages are realized. Second, various feature extraction methodologies are executed: Keyword extraction, SVD, and LDA. Third, the extracted features are combined in order to extract the features that fully characterize a given phishing message (see Fig. 13). Finally, feature selection methodologies are required in order to minimize the complexity of the classification task.

Let,

– Ω be the set of features determined by keyword extraction algorithm.
– Υ be the set of features determined by SVD.
– Γ be the set of features determined by LDA
– Ξ be the set of basic structural features.
– Then the final set of features $\mathcal{F}$ is given by,

$$\mathcal{F} = \Xi \cup ((\Gamma \cap \Upsilon) \cup (\Omega \cap \Upsilon)) = \Xi \cup ((\Gamma \cup \Omega) \cap \Upsilon) \tag{42}$$

As shown in Eq. (42), the final set of features $\mathcal{F}$ is a combination of structural basic features Ξ, independent of the other content-based feature sets Γ, Ω, and Υ. However, these feature sets are not independent of each other. They are represented by binary features, indicating whether or not a word is present in a given message.

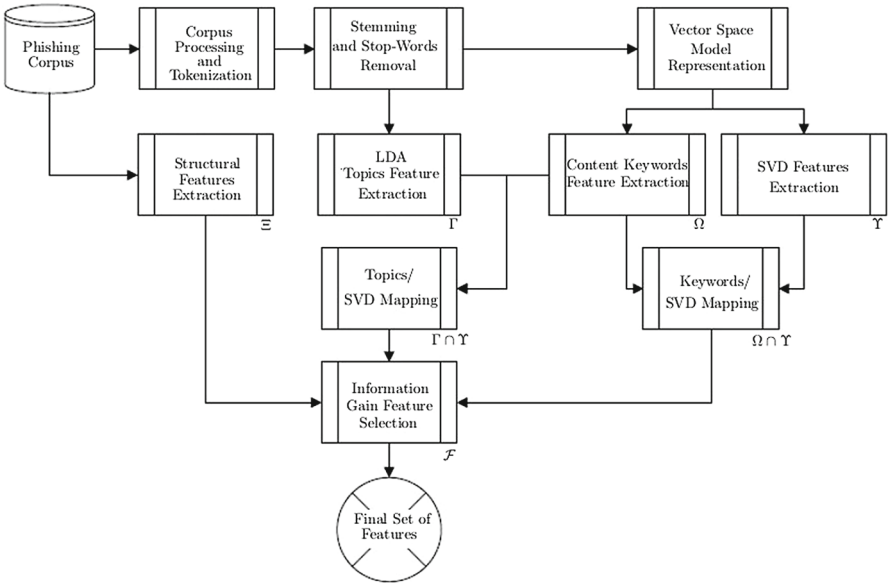

Fig. 13 Feature extraction flow diagram process for phishing messages

Keyword extraction

Based on a TF−IDF representation for the set of messages, the following keyword extraction technique was applied in order to define a set of relevant features: Using clustering, the entire collection of phishing emails is grouped into segments, and a geometric mean of the weights of each word within the messages contained in a given segment is computed. Then, these words are ranked and selected for each set of messages within their respective clusters. This procedure is based on work described by Velasquez et al. (2005).

Singular value decomposition

Using the TF−IDF matrix (Salton et al. 1975), its singular value decomposition (SVD) reduces the dimension of the $Term \times Document$-space. SVD uses a new representation of the feature space, in which the underlying semantic relationship between terms and documents is revealed.

As described by Papadimitriou et al. (1998), SVD preserves the relative distances in the vector space model matrix (e.g., TF−IDF), while projecting it into a lower-dimensional semantic space model. This is similar to what principal component analysis achieves by projecting features into their principal components, maintaining the information needed for an appropriate representation of the dataset.

Probabilistic topic models

A topic model can be viewed as a probabilistic model that relates documents and words through variables which represent the main topics inferred from the text itself. In this context, a document can be looked at as a mixture of topics, represented by probability distributions which generate the words in a document given these topics. The inferring process of the latent variables, or topics, is the key component of this model, whose main objective is to learn the distribution of the underlying topics from text in a given corpus of text documents.

A leading topic model is the LDA (Bíró et al. 2009; Blei et al. 2003). LDA is a Bayesian model in which latent topics of documents are inferred from estimated probability distributions over a training data set. In LDA every topic is modeled as a probability distribution over the set of words represented by the vocabulary ($w \in \mathcal{V}$), and every document as a probability distribution over a set of topics ($\mathcal{T}$). These distributions are sampled from multinomial Dirichlet distributions.

Structural features

As presented in Basne et al. (2008), Bergholz et al. (2010) and Fette et al. (2007), the extraction of basic structural features is needed for a minimum representation of phishing messages given a certain application. In the case of phishing emails, these features are associated with structural properties of email messages, such as their links, embedded codes, and the output of traditional Spam filters. It is important to note that all these features are extracted directly from email messages.

Appendix 2: List of symbols

Throughout the paper, we used the following notation:

Notation	Explanation
A	Total number of features
N	Total number of instances or objects
$\mathbf{x}$	Set of features $\mathbf{x} = \{x_1, \ldots, x_A\}$
x_a	ath feature of an object $\mathbf{x}$, $a \in \{1, \ldots, A\}$
y	Dependent variable
$(\mathbf{x}^i, y_i)$	Instance or object i, $i \in \{1, \ldots, N\}$
$\mathcal{F}_{x_a}$	Set of possible values for feature x_a
$\mathcal{F}_{\mathbf{x}}$	All possible values for all features $x_a \in \mathbf{x}$
$\mathcal{F}_y$	Set of target values
$\mathcal{T}$	Set of instances $\mathcal{T} = \{\mathbf{x}^i, y_i\}_{i=1}^N$
$\mathcal{T}_r$	Training dataset, subset of $\mathcal{T}$
$\mathcal{T}_e$	Testing dataset, subset of $\mathcal{T}$

Notation	Explanation
$\mathcal{N}$	Agent NATURE
$\mathcal{A}$	Agent ADVERSARY
$\mathcal{C}$	Agent CLASSIFIER
R	Regular adversary type (non-malicious)
M	Malicious adversary type
k	Number of information sets. Total number of type of messages that can be drawn by nature
$\{I_i\}_{i=1}^{k}$	Information sets for types of messages
$T_{\mathcal{A}}$	ADVERSARY types, $t \in T_{\mathcal{A}} = \{R, M\} \times \{1, \ldots, k\}$, also denoted as $T_{\mathcal{A}} = \{t_{R,j}\}_{j=1}^{k} \times \{t_{M,j}\}_{j=1}^{k}$
T_M	$\{t_{M,j}\}_{j=1}^{k}$
T_R	$\{t_{R,j}\}_{j=1}^{k}$
$p(t)$	Probability distribution for each type $t \in T_{\mathcal{A}}$
$\{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$	Set of all possible types of messages
Δ_k	Simplex of dimension $k-1$
$\mathcal{C} : \mathbb{R}^A \to \Delta_2$	Classification function
$\phi : \mathbb{R}^A \to \Delta_k$	Mimetic function that changes $\mathbf{x}$ to $\mathbf{x}'$, interpreted as the probability of sending a given message i when preferred message is l, $\forall i, l \in \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$
$\mu(t\|\mathbf{x})$	CLASSIFIER's belief that message is type t when observed instance is $\mathbf{x}$
$\sigma_{\mathcal{A}}^*$	Optimal strategy for the ADVERSARY
$\sigma_{\mathcal{C}}^*$	Optimal strategy for the CLASSIFIER
$d : \mathbb{R}^A \to \{+1, -1\}$	Decision function
$h_\tau \in \{+1, -1\}$	Hypothesis classification for message τ
$\{c_i\}_{i=1}^{k}$	Centroids associated to all types of messages
$\delta : \mathbb{R}^A \times \mathbb{R}^A \to \mathbb{R}$	Distance function between messages' centroids
$I(a)$	Information set for action a
π	Strategy profile
π_a	Probability that an action a is chosen if information set $I(a)$ is reached
λ	Parameter that controls the A-QRE algorithm
η_ϵ, η_A	Factors which amplifies the distance function between messages' centroids
$U_{\mathcal{A}} : T_{\mathcal{A}} \times \mathbb{R}^A \times \{+1, -1\} \to \mathbb{R}$	Utility function for the ADVERSARY
$u_{\mathcal{A}}$	Baseline utility for the ADVERSARY
$U_{\mathcal{C}} : T_{\mathcal{A}} \times \mathbb{R}^A \times \{+1, -1\} \to \mathbb{R}$	Utility function for the CLASSIFIER
$u_{\mathcal{C}}$	Baseline utility for the CLASSIFIER
θ_1	Cases where messages are correctly classified
θ_2	Cases where messages are incorrectly classified
$\mathcal{E}rr : \{+1, 1\} \times \{+1, -1\} \to \mathbb{R}$	Error function
ν	Classification threshold
ϵ_M	Classification gain when classifying correctly a malicious message
ϵ_R	Classification gain when classifying correctly a regular message
γ_M	Classification cost when classifying incorrectly a malicious message
γ_R	Classification cost when classifying incorrectly a regular message

References

APWG (2012) Phishing activity trends report, 2nd quarter 2012. Tech. rep., APWG
Basne R, Mukkamala S, Sung AH (2008) Detection of phishing attacks: a machine learning approach. Chapter Studies in fuzziness and soft computing. Springer, Berlin, pp 373–383
Bergholz A (2009) Antiphish: lessons learnt. In: Proceedings of the ACM SIGKDD workshop on CyberSecurity and intelligence informatics, ACM, CSI-KDD '09, New York, pp 1–2
Bergholz A, Beer JD, Glahn S, Moens MF, Paass G, Strobel S (2010) New filtering approaches for phishing email. J Comput Secur 18(1):7–35
Biggio B, Fumera G, Roli F (2008) Adversarial pattern classification using multiple classifiers and randomisation. In: SSPR & SPR '08: Proceedings of the 2008 joint IAPR international workshop on structural, syntactic, and statistical pattern recognition. Springer, Berlin, pp 500–509
Biggio B, Fumera G, Roli F (2009) Multiple classifier systems for adversarial classification tasks. In: MCS '09: Proceedings of the 8th international workshop on multiple classifier systems. Springer, Berlin, pp 132–141
Biggio B, Nelson B, Laskov P (2011) Support vector machines under adversarial label noise. In: Journal of Machine Learning Research—Proceedings of the 3rd Asian conference on machine learning (ACML 2011), vol 20. Taoyuan, Taiwan, pp 97–112
Bíró I, Siklósi D, Szabó J, Benczúr AA (2009) Linked latent dirichlet allocation in web spam filtering. In: AIRWeb '09: Proceedings of the 5th international workshop on adversarial information retrieval on the web, ACM, New York, pp 37–40
Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
Bravo C, Thomas LC, Weber R (2015) Improving credit scoring by differentiating defaulter behaviour. J Oper Res Soc 66(5):771–781. doi:10.1057/jors.2014.50
Brückner M, Scheffer T (2011) Stackelberg games for adversarial prediction problems. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, KDD '11, New York, pp 547–555
Brückner M, Kanzow C, Scheffer T (2012) Static prediction games for adversarial learning problems. J Mach Learn Res 13:2617–2654
Cho IK, Kreps DM (1987) Signaling games and stable equilibria. Q J Econ 102(2):179–221
Crespo F, Weber R (2005) A methodology for dynamic data mining based on fuzzy clustering. Fuzzy Sets Syst 150(2):267–284
Dalvi N, Domingos P, Mausam, Sanghai S, Verma D (2004) Adversarial classification. In: Proceedings of the tenth international conference on knowledge discovery and data mining, ACM Press, Seattle, vol 1, pp 99–108
Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41:391–407
Fette I, Sadeh N, Tomasic A (2007) Learning to detect phishing emails. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, ACM, New York, pp 649–656
Fudenberg D, Tirole J (1991) Game theory. MIT Press, Cambridge
Gibbons R (1992) Game theory for applied economists. Princeton University Press, Princeton
Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Intell Inf Syst 17(2/3):107–145
Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software: an update. SIGKDD Explor Newsl 11(1):10–18
Hamming R (1950) Error detecting and error correcting codes. Games Econ Behav 29(2):147–160
Harsanyi JC (1968) Games with incomplete information played by bayesian players. The basic probability distribution of the game. Manag Sci 14(7):486–502
Kantarcioglu M, Xi B, Clifton C (2011) Classifier evaluation and attribute selection against active adversaries. Data Min Knowl Discov 22:291–335
L'Huillier G, Hevia A, Weber R, Rios S (2010) Latent semantic analysis and keyword extraction for phishing classification. In: ISI'10: Proceedings of the IEEE international conference on intelligence and security informatics, Vancouver, pp 129–131
Liu W, Chawla S (2010) Mining adversarial patterns via regularized loss minimization. Mach Learn 81:69–83
Lowd D, Meek C (2005) Adversarial learning. In: KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, New York, pp 641–647

McKelvey R, Palfrey T (1998) Quantal response equilibria for extensive form games. Exp Econ 1(1):9–41
McKelvey RD, McLennan AM, Turocy TL (2010) Gambit: Software tools for game theory, version 0.2010.09.01. [online: Accessed 25 Nov 2012], http://www.gambit-project.org
Nazario J (2007) Phishing corpus. [online: Accessed 25 Nov 2012], http://bit.ly/jnazariophishing
Papadimitriou CH, Tamaki H, Raghavan P, Vempala S (1998) Latent semantic indexing: a probabilistic analysis. In: PODS '98: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, ACM, New York, pp 159–168
Peters G, Weber R, Nowatzke R (2012) Dynamic rough clustering and its applications. Appl Soft Comput 12(10):3193–3207
Peters G, Crespo F, Lingras P, Weber R (2013) Soft clustering: fuzzy and rough approaches and their extensions and derivatives. Int J Approx Reason 54(2):307–322
Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Advances in kernel methods, MIT Press, Cambridge, pp 185–208
Rosenblatt F (1962) Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Spartan Books, Washington
Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620
Sculley D, Wachman GM (2007) 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, ACM, New York, pp 415–422
Tambe M (2011) Security and game theory: algorithms, deployed systems, lessons learned. Cambridge University Press, New York
Turocy TL (2005) A dynamic homotopy interpretation of the logistic quantal response equilibrium correspondence. Games Econ Behav 51(2):243–263
Turocy TL (2010) Using quantal response to compute nash and sequential equilibria. Econ Theory 42(1):255–269
Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
Velasquez JD, Rios SA, Bassi A, Yasuda H, Aoki T (2005) Towards the identification of keywords in the web site text content: a methodological approach. Int J Web Inf Syst 1(1):53–57
Wu X, Srihari R (2004) 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, ACM, New York, pp 326–333
Xu R, Wunsch DC (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
Zareapoor M, Seeja K (2015) Text mining for phishing e-mail detection. In: Jain LC, Patnaik S, Ichalkaranje N (eds) Intelligent computing, communication and devices, advances in intelligent systems and computing. Springer India, pp 65–71. doi:10.1007/978-81-322-2012-1_8
Zhou Y, Kantarcioglu M, Thuraisingham B, Xi B (2012) Adversarial support vector machine learning. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, KDD '12, New York, pp 1059–1067