Design of Robust Classifiers for Adversarial Environments

Battista Biggio, Giorgio Fumera, Fabio Roli
Dept. of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{battista.biggio,fumera,roli}@diee.unica.it

***Abstract*—In *adversarial classification* tasks like spam filtering, intrusion detection in computer networks, and biometric identity verification, malicious adversaries can design attacks which exploit vulnerabilities of machine learning algorithms to evade detection, or to force a classification system to generate many false alarms, making it useless. Several works have addressed the problem of designing *robust* classifiers against these threats, although mainly focusing on specific applications and kinds of attacks. In this work, we propose a model of data distribution for adversarial classification tasks, and exploit it to devise a general method for designing robust classifiers, focusing on generative classifiers. Our method is then evaluated on two case studies concerning biometric identity verification and spam filtering.**

***Index Terms*—Pattern classification, adversarial classification, robust classifiers.**

I. Introduction

Adversarial classification tasks like spam filtering, intrusion detection in computer networks, and biometric identity verification have been typically faced as two-class classification problems, in which a classifier aims to discriminate between "malicious" and "legitimate" samples. Malicious samples are generated by an intelligent and adaptive adversary who can manipulate them to mislead the classifier [1]. For example, a well-known attack against spam filters consists in modifying a spam email by *inserting* words which are likely to appear in legitimate emails but not in spam, and by *obfuscating* typical "spammy" words (e.g., "cheap" can be misspelled as "che4p"). These attacks are respectively referred to as *good word insertion* (GWI) and *bad word obfuscation* (BWO) [2]. For another instance, consider a biometric system that should verify the identity of users by analysing their fingerprints. In this case an impostor may provide a "fake" fingerprint of a genuine user to the sensor (e.g., a "gummy" finger obtained from a latent fingerprint), to gain access to the system as that genuine user. This is typically referred to as *spoof* attack in biometrics [3].

The presence of malicious adversaries causes a specific kind of non-stationarity [2], [4], and raises several open issues with respect to state-of-the-art design methods of pattern recognition systems. To date, a systematic and unifying treatment of adversarial classification problems is still lacking, and this is attracting a growing interest from the pattern recognition and machine learning communities, how witnessed by a workshop held in the context of the NIPS 2007 conference,[1] and by a subsequent special issue of the *Machine Learning* Journal [5]. So far, works in the adversarial classification field have been focused on three main open problems:

1) identifying and categorising vulnerabilities of machine learning algorithms, which potentially expose them to adversarial attacks;
2) evaluating their performance under attack;
3) developing defence strategies to counteract attacks, and secure classifiers (i.e., designing *robust* classifiers).

Only a few attempts to develop general frameworks have been made, related to the first two issues above. Most of the works related to the third issue focused instead on specific applications, classifiers and attack scenarios.

In this work we focus on the problem of designing classifiers that are robust to attacks. To this aim, we first need to identify the potential attacks that may target a machine learning algorithm. A taxonomy of attacks was proposed in [4], where attacks were categorised according to three main axes: their *influence* on the classifier, the *security violation* they cause, and their *specificity*. The influence can be **causative**, if the attack aims to introduce vulnerabilities (to be exploited at classification phase) by manipulating training data; or **exploratory**, if the attack aims to find and subsequently exploit vulnerabilities at classification phase. The security violation can be an **integrity** violation, if it aims to get malicious samples misclassified as legitimate; or an **availability** violation, if it aims to increase the misclassification rate of legitimate samples, making the classifier unusable (e.g., a denial of service). The *specificity* ranges from **targeted**, if specific samples are considered (e.g., the adversary wants a given spam email to get past a spam filter), to **indiscriminate**.

As previously mentioned, most of the works which proposed defence strategies were tied to specific applications, classifiers and attack scenarios; moreover, they were mainly focused on counteracting exploratory integrity attacks. Besides this, they all follow a general approach which was first proposed in [1]: in order to build a robust (so-called adversary-aware) classifier, a designer should try to *anticipate* the attacks which may occur at operation phase. For instance, several works proposed different countermeasures to the common GWI and BWO attacks against text-based spam filters, mainly by modifying known classification algorithms, or their training phase [2], [6]–[13]. In [3] a similar rationale was exploited to counteract spoof attacks against multi-modal biometric

[1] http://mls-nips07.first.fraunhofer.de

systems for identity verification, which are indeed exploratory integrity attacks. In particular, a modification of the well-known likelihood ratio rule (LLR) [14] was proposed. Finally, we point out that less works have been devoted to causative attacks, which are less common than exploratory ones in real applications (see, for instance [4], [15], [16]).

In this work we first propose a generative model of data distribution for adversarial classification problems, which takes explicitly into account the presence of a malicious adversary, described in Sect. II. Based on such model, we then propose a method for robust classifier design, for generative classifiers. Differently from previous works, our method is not tied to a specific application, classification algorithm or attack scenario. Then, in Sect. III we report two examples of application of our method: a biometric identity verification task, to counteract spoof attacks; and a spam filtering task, to counteract GWI and BWO attacks. Finally, in Sect. IV we draw conclusions and sketch possible future works.

II. Robust classifier design

In this section we present a model of data distribution in presence of attacks, and show how it naturally suggests a general method for robust design of generative classifiers.

A. A model of data distribution in adversarial environments

All previous works assume, more or less explicitly, that adversarial classification problems are non-stationary [2], [4]. The reason is that the adversary can generate samples at operation phase which are different from the ones seen at design phase, causing training data to be not representative of testing data. Formally, let us denote with $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$ the data set collected for classifier design, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes an n-dimensional feature vector, and $y \in \{\mathrm{M}, \mathrm{L}\}$ a class label which can be either malicious (M) or legitimate (L). The elements of $\mathcal{D}$ are typically assumed to be i.i.d. samples of some unknown distribution, which we denote as $P_{\mathcal{D}}(\mathbf{X}, Y)$. When the classifier is subject to an attack after its deployment, either the training data (if an online learning algorithm is used), the testing data, or both (depending on the attack) may contain samples manipulated by the adversary which are not present in $\mathcal{D}$. In this case training and testing data follow two different distributions, and the ones affected by the attack are also different from $P_{\mathcal{D}}(\mathbf{X}, Y)$.

We propose a generative model to account for the presence of attacks in training and testing data. We denote their distributions respectively as $P_{\mathrm{tr}}(\mathbf{X}, Y)$ and $P_{\mathrm{ts}}(\mathbf{X}, Y)$ (we will simply use the symbol P to refer to both of them). We introduce a Boolean random variable $A \in \{\mathrm{T}, \mathrm{F}\}$ which determines whether the sample being generated is subject to the attack ($A = \mathrm{T}$) or not ($A = \mathrm{F}$). We also assume that A is independent of $\mathbf{X}$ and Y, and thus rewrite P_{tr} and P_{ts} as:

$$\begin{aligned} P(\mathbf{X}, Y) = \quad & P(\mathbf{X}, Y | A = \mathrm{F}) P(A = \mathrm{F}) \\ + \quad & P(\mathbf{X}, Y | A = \mathrm{T}) P(A = \mathrm{T}). \end{aligned} \quad (1)$$

We further assume that $\mathbf{X}$ is conditionally independent on Y, given A, which leads to

$$P(\mathbf{X}, Y, A) = P(A) P(Y|A) P(\mathbf{X}|Y, A) \ , \quad (2)$$

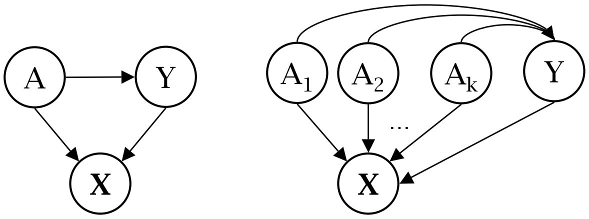

Fig. 1. Bayesian network for the generative model of eqs. 2 (left) and 3 (right).

and to the Bayesian network of Fig. 1 (left).

We finally assume that, when $A = \mathrm{F}$, the distribution $P(\mathbf{X}|Y, A)$ is identical for training and testing data, and equals $P_{\mathcal{D}}(\mathbf{X}|Y)$, which corresponds to the classical stationarity assumption: $P_{\mathrm{tr}}(\mathbf{X}|Y, A = \mathrm{F}) = P_{\mathrm{ts}}(\mathbf{X}|Y, A = \mathrm{F}) = P_{\mathcal{D}}(\mathbf{X}|Y)$. Note that $P(A = \mathrm{T}) = 1 - P(A = \mathrm{F})$ is the probability that a sample will be subject to the attack. Accordingly, if the attack does not affect training samples, then $P_{\mathrm{tr}}(A = \mathrm{T}) = 0$, and thus $P_{\mathrm{tr}}(\mathbf{X}, Y) = P_{\mathrm{tr}}(\mathbf{X}, Y | A = \mathrm{F})$. Moreover, in this case P_{tr} equals $P_{\mathcal{D}}$ by construction. We thus obtain: $P_{\mathrm{tr}}(\mathbf{X}, Y) = P_{\mathrm{tr}}(\mathbf{X}, Y | A = \mathrm{F}) = P_{\mathcal{D}}(\mathbf{X}, Y)$. Analogously, if testing samples are not affected by the attack, we have $P_{\mathrm{ts}}(\mathbf{X}, Y) = P_{\mathrm{ts}}(\mathbf{X}, Y | A = \mathrm{F}) = P_{\mathcal{D}}(\mathbf{X}, Y)$.

This generative model allows us to model attacks which affect either the class priors, through the term $P(Y|A)$, the class-conditional distribution of feature vectors $P(\mathbf{X}|Y, A)$, or both. Similarly, it can model attacks that affect either training or testing samples, or both. Our model can also be extended to account for the presence of different, (say, k) concurrent attacks, due for instance to the presence of different adversaries. To this aim, a different Boolean random variable A_i can be associated to the i-th attack. The corresponding Bayesian network is shown in Fig. 1 (right), while the distribution $P(\mathbf{X}, Y)$ is given by:

$$\sum_{A_1, \ldots, A_k \in \{\mathrm{T}, \mathrm{F}\}^k} P(\mathbf{X}, Y | A_1, \ldots, A_k) P(A_1, \ldots, A_k) \ . \quad (3)$$

B. Exploiting our model for designing robust classifiers

In the context of robust classifier design, our model can be exploited by making assumptions on the distributions $P(A = \mathrm{T})$ and $P(\mathbf{X}, Y | A = \mathrm{T})$ (either for training and testing samples), which allows one to anticipate the effects of the attack by taking them into account when the classifier is being designed. This embeds the general idea of adversary-aware classifier proposed in [1], and can be implemented in several ways, depending on the application scenarios.

We focus here on the design of robust classifiers against *exploratory integrity* attacks. In general, to counteract these attacks, a classifier has to be learnt on an hypothesised distribution $P_{\mathrm{ts}}(\mathbf{X}, Y)$, trying to prevent unknown (i.e., not present in training data) attacks. For simplicity, in this work we only consider *generative* classifiers, as they can be directly learnt on the assumed $P_{\mathrm{ts}}(\mathbf{X}, Y)$.

According to our model, to define $P_{\mathrm{ts}}(\mathbf{X}, Y)$, one has to set the probability distributions $P_{\mathrm{ts}}(A = \mathrm{T})$, $P_{\mathrm{ts}}(Y | A = \mathrm{T})$

and $P_{\rm ts}(\mathbf{X}|Y, A = {\rm T})$. The distributions $P_{\rm ts}(Y|A = {\rm F})$ and $P_{\rm ts}(\mathbf{X}, Y|A = {\rm F})$ are instead identical to the corresponding distributions of the data $\mathcal{D}$ collected for classifier training (see Sect. II-A), and can thus be estimated from $\mathcal{D}$. The distributions of $P_{\rm ts}(A = {\rm T})$ and $P_{\rm ts}(Y|A = {\rm T})$, namely, the prior probability of the attack and the class priors under attack, can be viewed as parameters in our model, while $P_{\rm ts}(\mathbf{X}|Y, A = {\rm T})$ depends on the considered kind of attack. Since a typical, reasonable assumption in the case of exploratory integrity attacks is that the adversary can only modify malicious testing samples, without affecting the class priors, we can set $P_{\rm ts}(Y) = P_{\mathcal{D}}(Y)$, and only assume a distribution for $P_{\rm ts}(\mathbf{X}|Y = {\rm M}, A = {\rm T})$. The only parameter of our method remains thus $P_{\rm ts}(A = {\rm T})$.

In general, when one has no idea of the kind of attack that may be incurred at operation phase, the distribution of $P_{\rm ts}(\mathbf{X}|Y = {\rm M}, A = {\rm T})$ can be assumed uniform. Note that this is implicitly done in several one-class classification tasks like anomaly-based intrusion detection [17]. However, this assumption might be too pessimistic, and lead to poor performances in other applications. In particular, when independence can be assumed among the features, and each feature x_i is attacked independently, a less pessimistic choice is possible. The model of Fig. 1 (right) can be exploited, indeed, using n random variables $A_1, \ldots, A_n$ (one for each feature), and further assuming that each feature x_i depends only on the corresponding A_i. Then, the distribution of each individual feature $P_{\rm ts}(x_i|Y = {\rm M}, A_i = {\rm T})$ can be modelled as uniform, instead of the joint distribution $P_{\rm ts}(\mathbf{X}|Y = {\rm M}, A_i = {\rm T})$.

For instance, consider the case of multi-modal biometric identity verification, in which a user is identified based on the analysis of different biometric traits. In this case, $\mathbf{x} = (x_1, \ldots, x_n)$ is the vector of matching scores, each representing the similarity between the submitted trait and a template corresponding to the claimed identity. The value of $P_{\rm ts}(A = {\rm T})$ can be interpreted here as the probability that at least one biometric trait is spoofed. However, it would be more realistic (and less pessimistic) to assume that spoofing a higher number of biometric traits is less probable, as it should be more difficult for the adversary. To this aim, we can more properly assume that each matching score x_i only depends on a different random variable A_i (besides the class label), and that the matching scores are independent, given Y (which is a common assumption in multi-modal biometric systems). Consequently, each $P_{\rm ts}(A_i = {\rm T})$ can be interpreted as the probability that a given biometric trait is spoofed. This allows to model spoof attacks against different matchers independently, and with different probabilities. Further assuming that $A_1, \ldots, A_n$ are i.i.d. random variables means that each matcher is attacked with the same probability, and that the probability of attacking k matchers is given by

$$P_{\rm ts}(A_1, \ldots, A_n) = \prod_{i=1}^{n} P_{\rm ts}(A_i) = \binom{n}{k} p^k (1-p)^{n-k}, \quad (4)$$

where $p = P_{\rm ts}(A_i = {\rm T})$, $i = 1, \ldots, n$, and k is the number of A_is whose value is T (i.e., the number of attacked matchers). Within these assumptions, one can more properly model each individual distribution $P_{\rm ts}(x_i|Y = {\rm M}, A_i = {\rm T})$ as uniform, assuming that the effect of a spoof attack against a given matcher may not be known in advance (indeed, this depends on the kind of spoof and of matching algorithm used).

In the next section we show how this implementation of our method can be applied in two distinct application scenarios to improve classifier robustness.

III. Application examples

We consider here two different applications, involving generative classifiers: a well-known fusion rule for multi-modal biometric identity verification tasks, and a text classifier for spam filtering. We apply the model described in the previous section, assuming a uniform probability distribution for each x_i, under attack, i.e., $P_{\rm ts}(x_i|Y = {\rm M}, A_i = {\rm T}) = {\rm const.}$, and assuming that the A_is are i.i.d. random variables with $P_{\rm ts}(A_i = {\rm T}) = {\rm const.}$, $i = 1, \ldots, n$.

A. Biometric identity verification

As our first application example, we consider the design of a typical multi-modal biometric system for identity verification, made up of a fingerprint and a face matcher, similarly to [3]. To access this system, a user has to provide the requested biometric traits and his identity. Then, each matcher provides a real-valued score which represents the similarity between the biometric trait of the user and the template associated to the claimed identity. The matching scores are subsequently fused, and the result is eventually compared with a decision threshold to classify the user either as "genuine" (legitimate) or "impostor" (malicious). Note that this corresponds to a classification problem in which the matching scores represent a feature vector $\mathbf{x} = (x_1, x_2)$, and the fusion rule is the classification algorithm.

As mentioned in Sect. I, a widely used fusion rule is the LLR [14]. It is defined as $f(\mathbf{x}) = P(\mathbf{x}|Y = {\rm L})/P(\mathbf{x}|Y = {\rm M})$, where $P(\mathbf{X}|Y)$, $Y \in \{{\rm L}, {\rm M}\}$ represents the class-conditional probability distribution of the matching scores, for genuine users (L) and impostors (M). If $f(\mathbf{x})$ exceeds (or, at least equals) a decision threshold, the user is accepted as a genuine one, otherwise he is rejected as an impostor. The two score distributions and the decision threshold are estimated on an available data set of scores, coming from known genuine and impostor users, typically through Maximum Likelihood Estimation (MLE). Since the available set of impostor scores is not likely to include any spoof attack, the impostor distribution $P(\mathbf{X}|Y = {\rm M})$ estimated on training data will be different from that incurred at operation phase (which, on the contrary, is likely to include spoof attacks). According to our model, $P(\mathbf{X}|Y = {\rm M})$ will be indeed equal to $P_{\mathcal{D}}(\mathbf{X}|Y = {\rm M})$, instead of $P_{\rm ts}(\mathbf{X}|Y = {\rm M})$. This makes a multi-modal biometric system, in principle, vulnerable to spoof attacks.

To overcome this issue, in [3] a modified version of the LLR was proposed, called *Extended* LLR. The underlying idea was to define an analytical model for the impostor distribution, which takes into account the probability of incurring spoof attacks at operation phase. The Extended LLR can be recast into our model, considering independence between the matching scores x_1 and x_2, between the corresponding A_is, and assuming that each A_i only affects x_i. In particular, in [3] it was also assumed that $P_{\text{ts}}(x_i|Y = \text{M}, A_i = \text{T}) = P_{\text{ts}}(x_i|Y = \text{L}, A_i = \text{F})$, namely, that a successful spoof attack is assigned a genuine score by the spoofed matcher. The values of $P_{\text{ts}}(A_i = \text{T})$ of our model can be computed from the parameters of the Extended LLR, which are: (1) the probability of attempting a spoof attack at least against one of the matchers, α; and (2) the *security* of each matcher, c_i, which is related to the probability of successfully spoofing that matcher (given by $1 - c_i$). In particular, it can be shown that

$$P_{\text{ts}}(A_i = \text{T}) = \alpha/(2^n - 1) \times (1 - c_i)\,, \tag{5}$$

where n is the number of matchers.

We show here that the straightforward implementation of the LLR suggested by our model of data distribution, with $P_{\text{ts}}(x_i|Y = \text{M}, A_i = \text{T})$ uniform, can yield similar performances with respect to the Extended LLR (with the main advantage that we assume a generic, uniform distribution for the spoof attacks, instead of a particular model), and investigate its performance for varying values of $P_{\text{ts}}(A_i = \text{T})$. We point out also that when $P_{\text{ts}}(A_i = \text{T}) = 0$, our approach is exactly equal to the standard LLR.

We experiment the standard LLR, the Extended LLR and the modified version of the LLR according to our model (which we will refer to as *Uniform* LLR) on the NIST Biometric Score Set, Release 1. [2] This data set contains raw similarity scores obtained on a set of 517 users from two different face matchers (named 'G' and 'C'), and from one fingerprint matcher using left and right index. For each user, one genuine score and 516 impostor scores are available for each matcher and each modality, for a total of 517 genuine and 266,772 impostor samples. We only consider here the scores of the 'G' face matcher and the ones of the fingerprint matcher for the left index, normalized in $[0, 1]$ using the min-max technique. We randomly subdivided the resulting data set into a training (TR) and testing set (TS) containing respectively 50% and 50% of the score pairs. To estimate the class-conditional distributions of the matching scores, we made the usual assumption of conditional independence between the matching scores given Y, and computed an MLE of $P(x_1, x_2|Y)$ from TR, using a product of Gamma distributions, as in [3].

Performance is evaluated using the Receiver Operating Characteristic (ROC) curve, which shows the percentage of accepted genuine users (Genuine Acceptance Rate, GAR) as a function of the percentage of accepted impostors (False Acceptance Rate, FAR), for all values of the decision threshold. The FAR can also be interpreted as the probability that an impostor is accepted as a genuine user through a "zero-effort" attack, namely, by providing his original traits. We evaluate the ROC curves for the different LLR implementations when no attack is performed, and in the case of fingerprint spoofs. We simulate spoof attacks against the fingerprint matcher by reproducing the same worst-case scenario investigated in [3], in which the score distribution of the spoofed traits is assumed to be equal to the one of the genuine traits. More precisely, we assume that *all* impostors in TS attempt a spoof attack, and that each impostor spoofs the fingerprint of a randomly-chosen genuine user in TS. Consequently, the effect of spoof attacks can be directly simulated by replacing the fingerprint matching score x_1 of each impostor in TS with that of the corresponding spoofed genuine user, avoiding the cumbersome task of constructing real fake biometric traits. Note that, since we assume that *all* impostors in TS attempt a spoof attack, the FAR can be still interpreted as the probability for an impostor to access the system, but, this time, with a spoof attack.

Results are shown in Fig. 2. On the left plot, we compare the standard LLR with our implementation, for different values of $P_{\text{ts}}(A_i = \text{T})$, with and without simulating spoof attacks. When no attack is considered, the Uniform LLR is slightly less accurate than the LLR, and its performance worsen as $P_{\text{ts}}(A_i = \text{T})$ increases. On the contrary, the Uniform LLR shows significantly better performances under attack. The underlying reason is that modifying $P_{\text{ts}}(A_i = \text{T})$ corresponds to obtain more or less wider decision regions for the legitimate (genuine) class in the feature space. On one hand, this allows to improve robustness against "unknown" attacks (i.e., malicious samples not present in training data, like spoof attacks) but, on the other hand, the performance when no attack is incurred degrades. Moreover, a too high value of $P_{\text{ts}}(A_i = \text{T})$ may worsen the performance even under attack, since too many genuine users are rejected.

On the right plot of Fig. 2, we compared the Extended LLR configured as suggested in [3], namely, with $\alpha = 0.01$, and the security parameters c_i for the fingerprint and face matcher respectively set to 0.7 and 0.3, with the Uniform LLR with $P_{\text{ts}}(A_i = \text{T}) = 1\text{E} - 3$. Note that, according to Eq. 5, the corresponding values of $P_{\text{ts}}(A_i = \text{T})$ were only slightly greater than $1\text{E} - 3$. This application example shows that the Uniform LLR can perform quite similarly to the Extended LLR, either in the presence of spoof attacks or not, with the advantage of assuming a simpler, generic model for spoof attacks (the uniform distribution) instead of a more specific one (as assumed by the Extended LLR).

B. Spam filtering

As our second application example, we consider a text classifier for spam filtering. Several real spam filters like the widely used SpamAssassin, SpamBayes and Bogofilter, include text classification algorithms to help discriminating between legitimate emails and spam on the basis of their textual content. These classifiers exploit the *bag-of-words* feature representation, in which any email is represented by a feature vector $\mathbf{x} = (x_1, \ldots, x_n)$ with Boolean values, each denoting whether a given word occurs in the considered email ($x_i = 1$) or not ($x_i = 0$).

To make a straightforward example of application of our method, we consider here a Naïve Bayes text classifier, which,

[2] Publicly available at http://www.itl.nist.gov/iad/894.03/biometricscores/

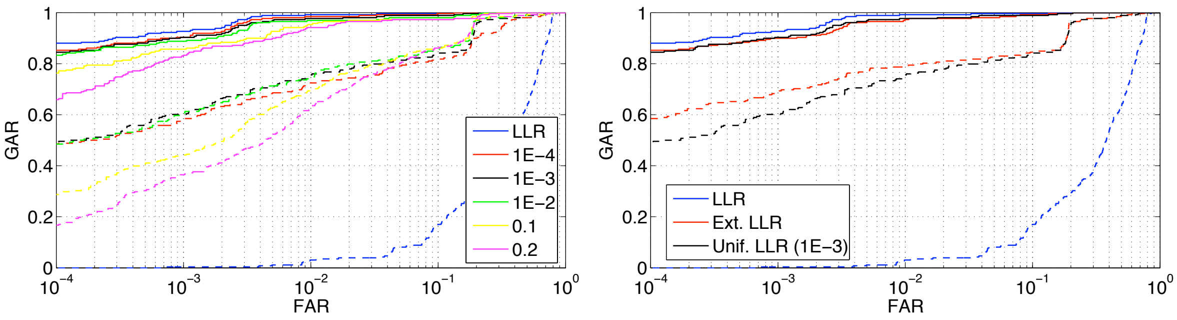

Fig. 2. ROC curves for the considered multi-modal biometric system, without attack (solid lines), or under fingerprint spoof attacks (dashed lines). *Left:* Comparison of the standard LLR with the Uniform LLR, for different values of $P_{\mathrm{ts}}(A_i = \mathrm{T})$, reported in the legend. *Right:* Comparison of the standard LLR, Extended LLR, and Uniform LLR with $P_{\mathrm{ts}}(A_i = \mathrm{T}) = 1\mathrm{E} - 3$.

besides its simplicity, has shown good performances on both text categorization and spam filtering tasks [18], [19]. The Naïve Bayes classifier assumes independence among the features, and its discriminant function can be written as

$$f(\mathbf{x}) = \log \frac{P(Y = \mathrm{M}) \times \prod_{i=1}^{n} p(x_i|Y = \mathrm{M})}{P(Y = \mathrm{L}) \times \prod_{i=1}^{n} p(x_i|Y = \mathrm{L})} \,. \qquad (6)$$

In the case of Boolean features, $f(\mathbf{x})$ turns out to be a linear discriminant function, given by $\sum_{i=1}^{n} w_i x_i + b$, where

$$w_i = \log \frac{P(x_i = 1|Y = \mathrm{M})}{P(x_i = 0|Y = \mathrm{M})} - \log \frac{P(x_i = 1|Y = \mathrm{L})}{P(x_i = 0|Y = \mathrm{L})} \qquad (7)$$

and

$$b = \log \frac{P(Y = \mathrm{M})}{P(Y = \mathrm{L})} + \sum_{i=1}^{n} \log \frac{P(x_i = 0|Y = \mathrm{M})}{P(x_i = 0|Y = \mathrm{L})} \,. \qquad (8)$$

The probability distributions $P(Y)$ and $P(x_i|Y)$ are often estimated through MLE over training data; moreover, $P(x_i|Y)$ is typically smoothed with a Laplacian prior (i.e., starting the word counts from 1) to avoid extreme probability values of 0 or 1 [18]. According to our model, the estimation of $P(x_i|Y = \mathrm{M})$ can be computed in a slightly different way, taking into account that the spam distribution can change at testing phase due to adversarial attacks. In particular,

$$P(x_i|Y = \mathrm{M}) = \sum_{A_i \in \{\mathrm{T},\mathrm{F}\}} P_{\mathrm{ts}}(x_i|Y = \mathrm{M}, A_i) P_{\mathrm{ts}}(A_i) \,. \qquad (9)$$

Similarly to the biometric identity verification task, the distributions $P_{\mathrm{ts}}(x_i|Y = \mathrm{M}, A_i = \mathrm{F})$, $i = 1, \ldots, n$ have to be estimated from training data, while $P_{\mathrm{ts}}(x_i|Y = \mathrm{M}, A_i = \mathrm{T})$ is set to 0.5 (the uniform distribution for Boolean features), and $P_{\mathrm{ts}}(A_i = \mathrm{T})$ is a parameter of our model.

We experiment on the benchmark TREC 2007 email corpus [20], which is made up of 75,419 real emails (25,220 legitimate and 50,199 spam messages), collected between April and July 2007. [3] The first 10,000 emails of this corpus (in chronological order) are used as training set (TR), and the next 20,000 emails as testing set (TS). We only consider a subset of TREC 2007 since we noted that further increasing the size of TR or TS does not significantly affect the results. We first extract the features (words) from training emails using the tokenization method of SpamAssassin, and then select $n = 20,000$ distinct features using the information gain criterion.

We consider GWI and BWO attacks in this experiment, and simulate them by modifying up to n_{MAX} words in each spam email, as in [2], [13]. More precisely, the feature vectors of all the spam emails in the testing set were modified directly, since changing a feature value from 1 (0) to 0 (1) correctly simulates a BWO (GWI) attack. Furthermore, we consider the following worst-case scenario. For a given feature vector $\mathbf{x}$ of a spam email, the features to modify (up to n_{MAX}) to get a new feature vector $\mathbf{x}'$ are the ones which minimise the discriminant function $f(\mathbf{x}')$. This leads to the maximum decrease in performance for the considered n_{MAX}. For linear classifiers with Boolean features, $\mathbf{x}'$ can be found by first sorting the weights w_1, w_2, …, w_n in descending order of their absolute value, and by sorting the features accordingly (note that this step can be carried out only once). We denote the sorted weights and features respectively as $w_{(1)}, w_{(2)}, \ldots w_{(n)}$, and $x_{(1)}, x_{(2)}, \ldots, x_{(n)}$, where $|w_{(1)}| \geq |w_{(2)}| \geq \ldots \geq |w_{(n)}|$. Then, for $i = 1, 2, \ldots, n$, and until the number of modified features does not exceed n_{MAX}: if $x_{(i)} = 1$ and $w_{(i)} > 0$, $x_{(i)}$ is set to 0; if $x_{(i)} = 0$ and $w_{(i)} < 0$, $x_{(i)}$ is set to 1; otherwise, $x_{(i)}$ is left unchanged.

Performance is evaluated with a measure derived from the area under the ROC curve (AUC). Since in tasks like spam filtering false positive (FP) errors are typically more harmful than false negative (FN) ones, the region of interest of the ROC curve is restricted to low FP rate values. As in [2], we used a more informative measure than the AUC, defined as the area of the region of the ROC curve corresponding to FP rates in $[0, 0.1]$: $\mathrm{AUC}_{10\%} = \int_0^{0.1} \mathrm{TP}(\mathrm{FP})\mathrm{dFP} \in [0, 0.1]$.

In Fig. 3 we report the results for the above experimental setup. In particular, we show how the value of $\mathrm{AUC}_{10\%}$ decreases for increasing n_{MAX}, i.e., with respect to the maximum number of words which can be modified in each spam. Note that the $\mathrm{AUC}_{10\%}$ value when no attack is performed corresponds to $\mathrm{AUC}_{10\%}$ when $n_{\mathrm{MAX}} = 0$, while it tends to 0 when $n_{\mathrm{MAX}} \approx 50$: this means that all the spam emails

[3] Publicly available at http://plg.uwaterloo.ca/~gvcormac/treccorpus07

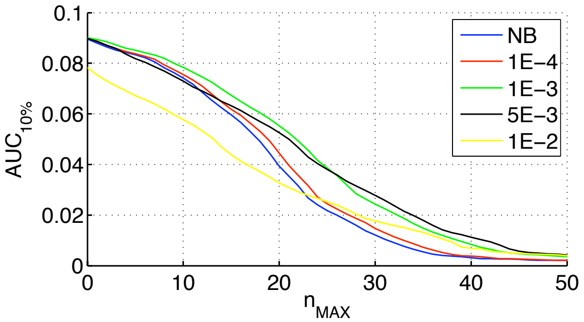

Fig. 3. $\mathrm{AUC}_{10\%}$ against different values of n_{MAX} for the Naïve Bayes classifier (NB), and the Naïve Bayes classifier modified according to our method, for different values of $P_{\mathrm{ts}}(A_i = \mathrm{T})$, reported in the legend.

in TS are misclassified as legitimate ones. Consequently, the more gracefully the $\mathrm{AUC}_{10\%}$ value degrades for increasing n_{MAX}, the higher the classifier robustness. As for the biometric identity verification task, we consider different values of $P_{\mathrm{ts}}(A_i = \mathrm{T})$ for our method, and compare it with the standard Naïve Bayes implementation. Note first that the performance not under attack (when $n_{\mathrm{MAX}} = 0$) is similar among all the considered classifiers, except for our method with $P(A_i = \mathrm{T}) = 1\mathrm{E} - 2$. Then, similarly to the previous case, the robustness of our method improves with respect to the standard Naïve Bayes, for increasing values of $P(A_i = \mathrm{T})$, but up to a critical value (1E-2 in this case) after that the exhibited classifier performance, both under attack and not, results worsened. This is due to the fact that, in this case, the model assumed for training data deviates too much from the model of testing data, and, thus, the classifier is no longer able to generalise well.

IV. Conclusions and future works

We developed a *general* strategy (i.e., not tied to a specific application or attack) to design classifiers which are robust against adversarial manipulation of input samples at testing phase. Our strategy is based on a generative model of data distribution for adversarial classification tasks that we developed to account for the intrinsic non-stationarity of this kind of problem. We provided a preliminary experimental investigation of our strategy in two different application scenarios: to improve robustness of multi-modal biometric systems for identity verification against spoof attacks, when the LLR score fusion rule is used; and to improve robustness of a spam filter against the BWO and GWI attacks, when the Naïve Bayes text classifier is used. The experimental results showed that our strategy can improve the robustness of a classifier to exploratory integrity attacks, without requiring specific assumptions on the feature vectors of testing samples under attack.

These promising results suggest to further attempt to exploit the proposed model of data distribution for the design of robust classifiers against other kinds of potential attacks, like causative availability ones. Moreover, we are also planning to investigate whether our model can be exploited to improve robustness of discriminative classifiers, like SVMs.

Acknowledgement

This work was partly supported by a grant awarded to B. Biggio by Regione Autonoma della Sardegna, PO Sardegna FSE 2007-2013, L.R. 7/2007 “Promotion of the scientific research and technological innovation in Sardinia”, and by the PRIN 2008 project “Biometric Guards - Electronic guards for protection and security of biometric systems” funded by the Italian Ministry of University and Scientific Research (MIUR).

References

[1] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, “Adversarial classification,” in *10th ACM SIGKDD Int.'l Conf. Knowledge Discovery and Data Mining (KDD)*, Seattle, 2004, pp. 99–108.

[2] A. Kolcz and C. H. Teo, “Feature weighting for improved classifier robustness,” in *6th Conf. Email and Anti-Spam (CEAS)*, CA, USA, 2009.

[3] R. N. Rodrigues, L. L. Ling, and V. Govindaraju, “Robustness of multimodal biometric fusion methods against spoof attacks,” *J. Vis. Lang. Comput.*, vol. 20, no. 3, pp. 169–179, 2009.

[4] M. Barreno, B. Nelson, A. Joseph, and J. Tygar, “The security of machine learning,” *Machine Learning*, vol. 81, pp. 121–148, 2010.

[5] P. Laskov and R. Lippmann, “Machine learning in adversarial environments,” *Machine Learning*, vol. 81, pp. 115–119, 2010.

[6] G. L. Wittel and S. F. Wu, “On attacking statistical spam filters,” in *1st Conf. Email and Anti-Spam (CEAS)*, CA, USA, 2004.

[7] D. Lowd and C. Meek, “Good word attacks on statistical spam filters,” in *2nd Conf. Email and Anti-Spam (CEAS)*, CA, USA, 2005.

[8] D. Sculley, G. Wachman, and C. E. Brodley, “Spam filtering using inexact string matching in explicit feature space with on-line linear classifiers,” in *TREC*, E. M. Voorhees and L. P. Buckland, Eds., vol. Special Publication 500-272. NIST, 2006.

[9] Z. Jorgensen, Y. Zhou, and M. Inge, “A multiple instance learning strategy for combating good word attacks on spam filters,” *Journal of Machine Learning Research*, vol. 9, pp. 1115–1146, June 2008.

[10] A. Globerson and S. T. Roweis, “Nightmare at test time: robust learning by feature deletion,” in *ICML*, ser. ACM International Conference Proceeding Series, W. W. Cohen and A. Moore, Eds., vol. 148. ACM, 2006, pp. 353–360.

[11] C. H. Teo, A. Globerson, S. Roweis, and A. Smola, “Convex learning with invariances,” in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008, pp. 1489–1496.

[12] O. Dekel, O. Shamir, and L. Xiao, “Learning to classify with missing and corrupted features,” *Machine Learning*, vol. 81, pp. 149–178, 2010.

[13] B. Biggio, G. Fumera, and F. Roli, “Multiple classifier systems for robust classifier design in adversarial environments,” *Int.'l Journal of Machine Learning and Cybernetics*, vol. 1, no. 1, pp. 27–41, 2010.

[14] K. Nandakumar, Y. Chen, S. C. Dass, and A. Jain, “Likelihood ratio-based biometric score fusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 342–347, February 2008.

[15] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, “Casting out demons: Sanitizing training data for anomaly sensors,” *Security and Privacy, IEEE Symposium on*, vol. 0, pp. 81–95, 2008.

[16] B. I. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S.-h. Lau, S. Rao, N. Taft, and J. D. Tygar, “Antidote: understanding and defending against poisoning of anomaly detectors,” in *Proc. 9th ACM SIGCOMM Internet Measurement Conf.*, ser. IMC '09. NY, USA: ACM, 2009, pp. 1–14.

[17] D. M. J. Tax, “One-class classification,” Ph.D. dissertation, Advanced School for Computing and Imaging (ASCI), 2001.

[18] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” in *Proc. AAAI Workshop on learning for text categorization*, 1998, pp. 41–48.

[19] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A bayesian approach to filtering junk e-mail,” *AAAI Technical Report WS-98-05, Madison, Wisconsin*, 1998.

[20] G. V. Cormack, “Trec 2007 spam track overview,” in *TREC*, E. M. Voorhees and L. P. Buckland, Eds., vol. Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.