Multiple Classifier Systems for Adversarial Classification Tasks

Battista Biggio, Giorgio Fumera, and Fabio Roli

Dept. of Electrical and Electronic Eng., Univ. of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{battista.biggio,fumera,roli}@diee.unica.it
http://prag.diee.unica.it

Abstract. Pattern classification systems are currently used in security applications like intrusion detection in computer networks, spam filtering and biometric identity recognition. These are *adversarial* classification problems, since the classifier faces an intelligent adversary who adaptively modifies patterns (e.g., spam e-mails) to evade it. In these tasks the goal of a classifier is to attain both a high classification accuracy and a high *hardness of evasion*, but this issue has not been deeply investigated yet in the literature. We address it under the viewpoint of the choice of the architecture of a multiple classifier system. We propose a measure of the hardness of evasion of a classifier architecture, and give an analytical evaluation and comparison of an individual classifier and a classifier ensemble architecture. We finally report an experimental evaluation on a spam filtering task.

1 Introduction

Pattern recognition systems, and in particular multiple classifier systems, are currently used in several security applications like biometric identity recognition, intrusion detection in computer networks and spam filtering, in which the task is to discriminate "attack" samples (e.g., a spam e-mail) from "legitimate" samples (e.g., legitimate e-mails). These kinds of tasks are named *adversarial* classification problems, since there is an intelligent, adaptive adversary who tries to camouflage patterns (like spam e-mails) to evade the security system. Accordingly, in these applications the goal is to attain both a high classification accuracy and a high *hardness of evasion*, which is intuitively related to the effort required to the adversary to evade the system. However in the machine learning and pattern recognition literature the issue of the hardness of evasion in adversarial classification problems has not been deeply and formally investigated yet. Most of the works proposed countermeasures against specific kinds of attacks for spam filtering and intrusion detection tasks (see for instance [1,2,3]), and only few of them proposed formal models of adversarial classification tasks [4,5], or analysed the main issues raised by the application of machine learning techniques [6]. Therefore, from an engineering viewpoint the design of accurate

J.A. Benediktsson, J. Kittler, and F. Roli (Eds.): MCS 2009, LNCS 5519, pp. 132–141, 2009.

and hard to evade classification systems for security applications is still an open problem.

In this work we argue that the hardness of evasion has to be taken into account in two distinct aspects of the design of a pattern recognition system: the choice of the features and the choice of the classifier architecture. Here we focus on the latter, and propose a quantitative measure of the hardness of evasion of a classifier architecture. We then analytically evaluate and compare the hardness of evasion and the accuracy of two specific single classifier and multiple classifier architectures which are used in many real security systems, and were supported so far only by intuitive arguments and empirical evidence. In light of our theoretical findings, we give an experimental evaluation of the accuracy and hardness of evasion of the considered classifier architectures on a spam filtering task, using the well known SpamAssassin open source spam filter.

2 Analysis of Multiple Classifier Systems for Adversarial Classification Tasks

In many classification systems used in security applications, like multimodal biometric authentication and verification, and intrusion detection in computer networks, the input features come from heterogeneous sources (for instance, images of faces and fingerprints). In these cases combining classifiers trained on the different feature subsets has been proposed as a natural way to design a simpler and more accurate classification system than a single classifier trained on all the available features [7,8,9,10]. Few authors proposed the use of MCSs with the explicit goal of improving the hardness of evasion (see for instance [2]). MCS architectures turn out to be used also in commercial and open source security systems, like the SpamAssassin spam filter (`http://spamassassin.apache.org`) and the Snort intrusion detection system (`http://www.snort.org`). However, with the only exception of a previous work by the authors [11], the use of MCSs for improving the hardness of evasion is supported only by intuitive and qualitative motivations, besides experimental evidences, and lack of a clear and sound theoretical support. In this section we propose a quantitative measure to evaluate the hardness of evasion of pattern classification systems, and apply it to analyse two different classifier architectures which are used in real adversarial classification tasks and are simple enough to allow for an analytical investigation.

2.1 The Concept of Hardness of Evasion

In security tasks there is a formal and agreed definition of classification accuracy in terms of the false positive (FP) and the false negative (FN) error rates. Instead, there is no formal and agreed definition of hardness of evasion. Intuitively, it depends on the “difficulty” for an adversary to evade the security system, but its evaluation depends on the specific task and on the kind of security system. Our aim is to propose a quantitative definition related to pattern classification systems. We first point out that in such systems the hardness of evasion can be analysed under two distinct aspects: the feature set, and the way in which features are combined (namely, the classifier architecture). Indeed, given an "attack" sample, an adversary has to consider two distinct issues: first, which features have to be modified, to evade the classifier? Second, how can patterns be camouflaged, so that the values of the targeted features are modified as desired? The latter issue is related to the nature of the individual features: the designer of the classifier should select features that are robust against pattern camouflage. However, in the design of a security system it is safer to assume that the adversary knows which features are used, and that he can always devise a way to evade them, although with some effort. Moreover, in practice quantifying the relative effort that is needed to modify different features is often very difficult. Accordingly, the hardness of evasion should also rely on forcing the adversary to modify as many features as possible to evade the system. This clearly depends on how the individual features are combined by the classifier architecture, which directly leads to the former issue above, namely, which (and how many) features have to be modified to evade the classifier. Accordingly, the hardness of evasion of a pattern classifier can be pursued at two distinct levels: the choice of the individual features, which should be not trivial to modify by pattern camouflage, and the choice of the classifier architecture, which should force the adversary to modify as many features as possible to evade the classifier. Although these choices are not necessarily independent on each other, they can nevertheless be addressed separately (perhaps in a closed-loop design cycle). In this work, we focus on the latter issue, namely designing a hard to evade classifier architecture in the sense defined above. To this aim, we give the following quantitative definition of the hardness of evasion of a classifier architecture:

> *For a given feature set, the hardness of evasion is defined as the expected value of the minimum number of features which have to be modified to evade the classifier.*

Accordingly, given two different classifiers A and B trained on the same feature set, A is harder to evade than B, if the expected minimum number of features that need to be modified to evade A is higher than the one needed to evade B.

2.2 A Theoretical Analysis of Multiple Classifier Systems for Adversarial Classification Tasks

In this section we focus on two classifier architectures (a single classifier and a MCS) used in multimodal biometric systems, in the SpamAssassin anti-spam filter, and in the Snort intrusion detection system. We will analytically evaluate and compare their hardness of evasion, defined as in Sect. 2.1, and classification accuracy. We first construct a model of the classification problem and of the two architectures, suitable to an analytical investigation. We consider n binary-valued features taking on the values 0 and 1, denoting respectively the absence and the presence of a given "attack" characteristic (as happens in Snort, while in SpamAssassin there are also features related to "legitimate" characteristics, which take on the values 0 and -1). The classifier architectures are shown in

Fig. 1. The first one is a "monolithic" classifier: a linear combination of the features with a decision threshold, as in SpamAssassin. A variant of this architecture is used by Snort: the logical OR between all features (viewed as boolean values), where 1 corresponds to *true* (accordingly, a pattern is labelled as "attack" if at least one feature detects an attack characteristic). The second one is an ensemble of classifiers trained on disjoint feature subsets, as in multimodal biometric systems [7,8]. To allow a direct comparison with the monolithic architecture, we consider an implementation in which the individual ensemble members are linear classifiers and are combined with the OR logical function. We denote the class labels as A ("attack") and L ("legitimate"), and the random feature vector as $X = (x_1, ..., x_n) \in \{0, 1\}^n$. To make an analytical evaluation possible, we assume that features are i.i.d. The (common) class-conditional distribution of each feature will be denoted as p_{1A}, p_{0A}, p_{1L} and p_{0L}, where $p_{1A} = P(X_i = 1|X \in A)$ for any $i = 1, ..., n$, and so on (obviously, $p_{1A} = 1 - p_{0A}$ and $p_{1L} = 1 - p_{0L}$). We also consider all the weights of the monolithic linear classifier to be identical. This is reasonable, given that all features are assumed to have the same discriminant capability. Without loosing generality, we normalise the weight values to 1 and consider only a variable threshold $t > 0$. The decision function $s_{\mathrm{M}}(x)$ of the monolithic classifier can then be written as follows (see Fig. 1, left):

$$s_{\mathrm{M}}(x) = \begin{cases} 1, \text{ if } \sum_{i=1}^{n} x_i - t \geq 0, \\ 0, \text{ otherwise } . \end{cases} \tag{1}$$

Note that also the OR decision function used by Snort can be written as (1), provided that $t \in (0, 1]$. These architectures can also be viewed as MCSs, if features are the decisions of individual classifiers. We also consider the weights of the individual classifiers of the MCS to be all identical and normalised to 1, and a common value also for the decision thresholds, denoted with t'. Assuming further that the n features are uniformly subdivided among the N classifiers (this requires n to be multiple of N), the decision function of the m-th individual linear classifier of the MCS (Fig. 1, right) can be written as:

$$s_{\mathrm{M}}^{m}(x) = \begin{cases} 1, \text{ if } \sum_{i=1}^{n/N} x_i^m - t' \geq 0, \\ 0, \text{ otherwise } , \end{cases} \tag{2}$$

where x_i^m is the i-th feature of the m-th classifier. The MCS architecture is shown in Fig. 1, right.

We now compute the accuracy of the two classifiers above in terms of the FP and FN rates, as functions of n, N, t, t', and of the class-conditional feature distribution. The FP rate is the probability that a legitimate sample is misclassified as an attack, $FP = \mathrm{P}\left(s_{\mathrm{M}}(X) = 1|X \in L\right)$. For the monolithic classifier, from the definition of $s_{\mathrm{M}}(x)$ in (1), this happens if at least $\lceil t \rceil$ features equal 1 for a legitimate pattern. Being the features i.i.d., the corresponding probability is:

$$FP = \sum_{k=\lceil t \rceil}^{n} \binom{n}{k} p_{1L}^{k} \times p_{0L}^{n-k} \quad . \tag{3}$$

The FN rate equals 1 minus the true positive (TP) rate, which is defined as $\mathrm{P}\left(s_{\mathrm{M}}(X) = 1|X \in A\right)$. This equals the probability that at least $\lceil t \rceil$ features equal

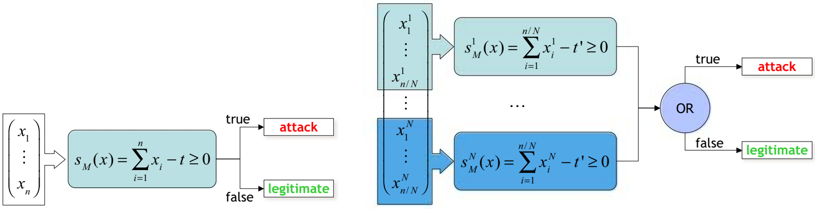
x_1
x_n
$s_M(x) = \sum_{i=1}^{n} x_i - t \geq 0$
true
attack
false
legitimate
x_1^1
$x_{n/N}^1$
x_1^N
$x_{n/N}^N$
$s_M^1(x) = \sum_{i=1}^{n/N} x_i^1 - t' \geq 0$
...
$s_M^N(x) = \sum_{i=1}^{n/N} x_i^N - t' \geq 0$
OR
true
attack
false
legitimate

Fig. 1. The two classifier architectures considered in this work. A single, linear classifier (left), and an ensemble of linear classifiers combined by the OR logical function (right). In both cases the weights of the linear combination are assumed to be all identical, and a normalised value of 1 is considered.

1 for an attack pattern, and can be computed analogously to the FP rate. For the monolithic classifier one obtains:

$$TP = \sum_{k=\lceil t \rceil}^{n} \binom{n}{k} p_{1A}^{k} \times p_{0A}^{n-k} \ . \tag{4}$$

Using the OR decision function instead of a linear combination, the expressions of FP and TP are the same ones above, with k ranging from 1 to n.

For the MCS, FP is the probability that at least one individual classifier outputs 1 for a legitimate sample. Each individual classifier is trained on n/N different i.i.d. features and has the same decision function (2). Their decisions are thus i.i.d. Denoting the common decision function as $s(x)$, one obtains:

$$\begin{aligned} FP &= \sum_{m=1}^{N} \binom{N}{m} \ \mathrm{P}\left(m \text{ classifiers say A } \wedge \ N-m \text{ say L}|X \in L\right) \\ &= \sum_{m=1}^{N} \binom{N}{m} \ \left[\mathrm{P}\left(s(x) = 1 | X \in L\right)\right]^{m} \times \left[\mathrm{P}\left(s(x) = 0 | X \in L\right)\right]^{N-m} \\ &= \sum_{m=1}^{N} \binom{N}{m} \left[\sum_{k=\lceil t' \rceil}^{n/N} \binom{n/N}{k} p_{1L}^{k} \times p_{0L}^{n/N-k}\right]^{m} \times \\ & \qquad \left[\sum_{k=n-\lceil t' \rceil}^{n/N} \binom{n/N}{k} p_{0L}^{k} \times p_{1L}^{n/N-k}\right]^{N-m} \ . \end{aligned} \tag{5}$$

The TP rate of the MCS is the probability that at least one individual classifier outputs 1 for an attack sample. This can be computed analogously to (5):

$$\begin{aligned} TP = \sum_{m=1}^{N} \binom{N}{m} & \left[\sum_{k=\lceil t' \rceil}^{n/N} \binom{n/N}{k} p_{1A}^{k} \times p_{0A}^{n/N-k}\right]^{m} \times \\ & \left[\sum_{k=n-\lceil t' \rceil}^{n/N} \binom{n/N}{k} p_{0A}^{k} \times p_{1A}^{n/N-k}\right]^{N-m} \ . \end{aligned} \tag{6}$$

The hardness of evasion was defined as the expected value over the distribution $\mathrm{P}(X|X \in A)$ of the minimum number of features that have to be modified in an attack sample to evade the classifier. We denote with $n_{\min}(x)$ such value for any sample x, for the monolithic classifier (Fig. 1, left). Since x is labelled as A when at least $\lceil t \rceil$ features equal 1, denoting with $k(x)$ the number of features equal to 1 it follows that:

$$n_{\min}(x) = \begin{cases} k(x) - \lceil t \rceil + 1, & \text{if } k(x) \geq \lceil t \rceil, \\ 0, & \text{otherwise} \ . \end{cases}$$

The expected value of $n_{\min}(x)$, denoted as $\overline{n}_{\min}$, can be computed as follows:

$$\begin{aligned}\overline{n}_{\min} &= \sum_{k=\lceil t\rceil}^{n} \left[k - \lceil t\rceil + 1\right] \times \mathrm{P}\left(k \text{ features equal } 1 | X \in A\right) \\ &= \sum_{k=\lceil t\rceil}^{n} \left[k - \lceil t\rceil + 1\right] \times \binom{n}{k} \times p_{1A}^{k} \times p_{0A}^{n-k} \ .\end{aligned} \tag{7}$$

To evade the MCS with the OR decision function (Fig. 1, right) it is necessary to evade all individual classifiers whose output is 1. Denoting with $n_{\min,m}(x)$ the minimum number of features that have to be modified in the m-th classifier, for a given attack sample x, the overall minimum number of features to modify is $n_{\min}(x) = \sum_{m=1}^{N} n_{\min,m}(x)$. The expectation is thus given by $\overline{n}_{\min} = \sum_{m=1}^{N} \mathrm{E}_{n_{\min,m}(X)|X \in A}\left[n_{\min,m}(X)\right]$. Since all classifiers are trained on disjoint subsets of i.i.d. features of the same size n/N and have the same decision function, the N random variables $n_{\min,m}(X), m = 1, ..., N$ are i.i.d. as well. Their expectation can be computed exactly as in (7), and thus we obtain:

$$\overline{n}_{\min} = N \times \sum_{k=\lceil t'\rceil}^{n/N} \left[k - \lceil t'\rceil + 1\right] \times \binom{n/N}{k} \times p_{1A}^{k} \times p_{0A}^{n/N-k} \ . \tag{8}$$

Since an analytical comparison between the above expressions of accuracy and hardness of evasion is not possible, we give a numerical comparison. To this aim, we first fix the class-conditional distribution of the features to values that can be representative of a real adversarial task like spam filtering (taking into account that features are assumed to be i.i.d.). We chose the values $p_{1A} = 0.25$ and $p_{1L} = 0.15$, namely, each individual feature detects 25% of the attacks and also erroneously identifies 15% of legitimate samples as attacks. We then evaluate the accuracy of the monolithic classifier and of the MCS using the receiver operating characteristic (ROC) curve (namely, the TP rate as a function of the FP rate, obtained by varying the decision thresholds t and t'). For the monolithic classifier (Fig. 2, left) we consider different values of the number of features n. As expected (being the features i.i.d.), the discriminant capability increases for increasing n. For the chosen values of p_{1A} and p_{1L}, $n = 600$ is sufficient to obtain nearly zero FP and FN rates. A realistic accuracy for spam filters is the one for n equal to about 300. The accuracy of the MCS was evaluated for different values of the ensemble size N, with n fixed to 300 (Fig. 2, right). It can be seen that the MCS discriminant capability is lower than that of the monolithic classifier (the MCS ROC curves are always below the one of the monolithic classifier for $n = 300$). The reason is that the individual classifiers of the MCS are much less accurate than the monolithic one, since they are trained on a lower number (n/N) of i.i.d. features. This turns out to be true also for different class-conditional feature distributions. We point out however that this result holds for the case in which the classifiers are *not* under attack.

We finally evaluate and compare the hardness of evasion (7) and (8) for $n = 300$ features. For a fair comparison between the monolithic classifier and the MCS we consider a fixed working point on the ROC curve defined by choosing classifier parameters (the decision thresholds t and t') that minimise a classification cost given by $FP + \frac{1}{C}FN$, where C denotes the relative cost of FP and FN errors. Since in security applications FP errors are more harmful than FN ones,

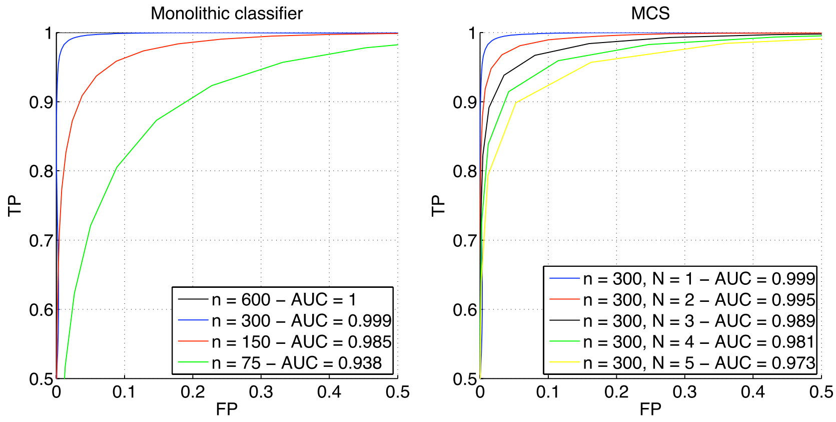

Fig. 2. Top-left part of the ROC curves of the monolithic linear classifier for different feature set sizes n (left), and of the MCS for $n = 300$ and different ensemble sizes N (right), for i.i.d. features with class-conditional distribution given by $p_{1A} = 0.25$ and $p_{1L} = 0.15$. The area under the ROC curve (AUC) is also reported.

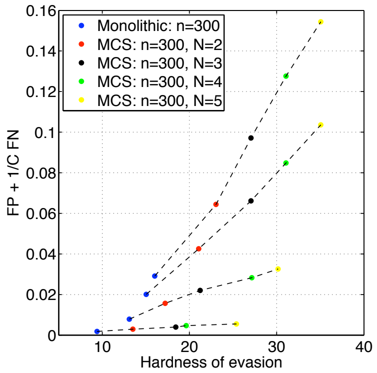

Fig. 3. Classification cost $FP + \frac{1}{C}FN$ as a function of the hardness of evasion for the monolithic classifier and four MCS with ensemble size $N = 2, ..., 5$, $n = 300$ features, and four C values. Each dashed line corresponds to a different C value: from top to bottom, $C = 1, 2, 10, 100$.

we consider $C > 1$. The comparison was made for four C values and four MCS ensemble sizes: $C = 1, 2, 10, 100$, and $N = 2, 3, 4, 5$. The corresponding classification cost and hardness of evasion are reported in Fig. 3. The comparison between the monolithic classifier and the MCSs, for any fixed C value, clearly shows that the monolithic classifier is more accurate at any given operating point when the adversary does not attack, but it is also easier to evade. Moreover, while the MCS accuracy decreases for increasing ensemble sizes, the hardness of evasion increases. Therefore, in the considered classifier architectures there is a trade-off between the accuracy when the classifier is not under attack, and the hardness of evasion. Note also that for increasing C values (namely, when a smaller FP rate is required), the accuracy of the MCS approaches that of the monolithic classifier, while the hardness of evasion remains significantly higher.

The analytical results in this section are limited to two classifier architectures, and hold only under rather strict conditions on the class-conditional feature distribution. Nevertheless, they allow to provide a first, formal evaluation and comparison of monolithic classifiers and MCSs in terms of both classification accuracy and hardness of evasion, and suggest that MCSs can be useful to attain a higher hardness of evasion than monolithic classifiers. In the next section we will give an empirical evaluation of these architectures for a spam filtering task, in light of the analytical results above.

3 Experimental Results

We analytically found in Sect. 2.2 that, when the adversary does not attack, a linear classifier with identical weights trained on i.i.d. features is more accurate than an ensemble of linear classifiers with identical weights and decision thresholds trained on disjoint subsets of identical size of the same features, and combined with the OR logical function, but it is easier to evade. In this section we empirically evaluate whether this result holds also in a real application where the assumption of i.i.d. features could be not satisfied. To this aim we considered the SpamAssassin spam filter (version 3.2.5), and the TREC 2007 e-mail corpus, publicly available at `http://plg.uwaterloo.ca/~gvcormac/treccorpus07` and made up of 75,419 real e-mails (25,220 legitimate and 50,199 spam messages) collected between April 2007 and July 2007.

SpamAssassin can be considered as a linear classifier with several hundred binary features (rules associated to legitimate or spam e-mail characteristics take on respectively -1 and 0 values and 1 and 0 values), nine of which are actually associated to the outputs of a text classifier. The main aim of our experiments was to compare the two classification architectures of Fig. 1. To this end, we compared the SpamAssassin classifier architecture (a monolithic linear classifier) with MCSs trained on disjoint subsets of its features and combined with the OR logical function. However, even disregarding the nine tests associated to the text classifier, which exhibit a significantly higher discriminant capability, the remaining features cannot be considered i.i.d. Their correlation on the TREC legitimate e-mails ranges in $[-0.0045, 0.2821]$, with 0.0001 mean and 0.0025 std. dev., while for spam e-mails it ranges in $[-0.1867, 0.3265]$ with 0.0004 mean and 0.0069 std. dev. Their class-conditional distribution is given by $p_{1A} \in [0, 0.5588]$, with 0.0105 mean and 0.0374 std. dev., and by $p_{1L} \in [0, 0.0600]$ with 0.0003 mean and 0.0029 std. dev. To take this into account, we used different weights in the linear classifiers. Moreover, the text classifier of SpamAssassin (which has a continuous-valued output) was considered as one of the individual classifiers of the MCSs. The experiments were carried out as follows. We considered only the $n = 549$ features whose value was not constant over all e-mails of the TREC corpus. The text classifier was trained on the first $10,000$ e-mails in

Table 1. Mean and standard deviation of the FP and FN rates and of the hardness of evasion of the monolithic classifier trained on the SpamAssassin features, and of two MCSs with ensemble size $N = 3, 10$, trained on disjoint subsets of the same features

	FP rate	FN rate	hardness of evasion
monolithic	0.0061(±0.0024)	0.0363(±0.0084)	1.37(±0.09)
MCS, $N = 3$	0.0062(±0.0013)	0.0520(±0.0060)	3.01(±0.13)
MCS, $N = 10$	0.0097(±0.0020)	0.0569(±0.0052)	3.25(±0.22)

chronological order. The next $10,000$ e-mails were used to train the linear classifiers, using a support vector machine (SVM) with the linear kernel (the publicly available `libsvm` software was used [12]). The operating point of all individual classifiers was set by keeping the FP rate below 1%. Two ensemble sizes for the MCS were considered: $N = 3, 10$. The 549 features were randomly and uniformly subdivided among the individual classifiers of the MCS. The accuracy (FP and FN rates) and the hardness of evasion at the chosen operating point were then computed on the remaining $55,419$ e-mails, and are reported in Table 1.

Table 1 shows that the considered MCS architecture provides a lower classification accuracy than the monolithic architecture, when they are not under attack (both the FP and FN rates of the MCS are slightly higher, and increase for increasing values of the ensemble size). However the hardness of evasion of the MCS is higher than the one of the monolithic classifier, and increases for increasing ensemble size. It is worth noting that this qualitative behaviour is the same found by our theoretical analysis of Sect. 2.2, although the assumption of i.i.d. features is violated, and the experimental setup does not match the one considered in Sect. 2.2 since the weights of the individual classifiers are not identical. In particular, the considered classifier architectures are characterised by a trade-off between classification accuracy and hardness of evasion: the MCS architecture can allow to improve the hardness of evasion, although its accuracy when the system is not under attack can be lower.

4 Conclusions

In this work we addressed for the first time the issue of quantitatively evaluating the hardness of evasion of a pattern classifier for security applications, and in particular of multiple classifier systems. We argued that the hardness of evasion has to be evaluated in two distinct steps of classifier design, namely the choice of the features and of the classifier architecture. We focused on the latter step, and proposed a quantitative measure of the hardness of evasion of a classifier architecture, related to the number of features that should be modified by the adversary to evade the whole classifier. This allowed us to give an analytical evaluation and comparison of two classifier architectures which are used in real security systems, but were motivated so far only by intuitive arguments and empirical evidence. The analytical results were exploited to give an experimental evaluation of these architectures in a real case study related to a spam filtering task. Our theoretical and experimental results suggest that MCSs can allow to improve the hardness of evasion, although their classification accuracy can be lower than that of a single classifier, when the system is not under attack. Moreover, the experimental results suggest that the validity of our theoretical conclusions can go beyond the assumptions under which they have been derived. We believe that the framework proposed in this work can be a starting point to derive principled guidelines for the design of pattern classifiers for adversarial classification problems.

References

1. Globerson, A., Roweis, S.T.: Nightmare at test time: robust learning by feature deletion. In: Cohen, W.W., Moore, A. (eds.) ICML. ACM International Conference Proceeding Series, vol. 148, pp. 353–360. ACM, New York (2006)
2. Perdisci, R., Gu, G., Lee, W.: Using an ensemble of one-class svm classifiers to harden payload-based anomaly detection systems. In: International Conference on Data Mining (ICDM), pp. 488–498. IEEE Computer Society, Los Alamitos (2006)
3. Jorgensen, Z., Zhou, Y., Inge, M.: A multiple instance learning strategy for combating good word attacks on spam filters. Journal of Machine Learning Research 9, 1115–1146 (2008)
4. Lowd, D., Meek, C.: Adversarial learning. In: Press, A. (ed.) Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Chicago, IL (2005)
5. Dalvi, N., Domingos, P., Mausam, S.S., Verma, D.: Adversarial classification. In: Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Seattle, pp. 99–108 (2004)
6. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: ASIACCS 2006: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pp. 16–25. ACM, New York (2006)
7. Kittler, J., Hatef, M., Duin, R.P., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(3), 226–239 (1998)
8. Ross, A.A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer Publishers, Heidelberg (2006)
9. Haindl, M., Kittler, J., Roli, F. (eds.): MCS 2007. LNCS, vol. 4472. Springer, Heidelberg (2007)
10. Giacinto, G., Roli, F., Didaci, L.: Fusion of multiple classifiers for intrusion detection in computer networks. Pattern Recognition Letters 24, 1795–1803 (2003)
11. Biggio, B., Fumera, G., Roli, F.: Evade hard multiple classifier systems. In: Okun, O., Valentini, G. (eds.) Supervised and Unsupervised Ensemble Methods and Their Applications. Studies in Computational Intelligence. Springer, Heidelberg (2009) (in press)
12. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`