Gradient Correlation: Are Ensemble Classifiers More Robust Against Evasion Attacks in Practical Settings?

Fuyong Zhang[1], Yi Wang[1](✉), and Hua Wang[2]

[1] Dongguang University of Technology, Dongguan, Guangdong, China
{zhangfy,wangyi}@dgut.edu.cn
[2] Institute for Sustainable Industries and Liveable Cities,
VU Research, Victoria University, Melbourne, Australia
hua.wang@vu.edu.au

Abstract. Pattern recognition is an essential part of modern security systems for malware detection, intrusion detection, and spam filtering. Conventional classifiers widely used in these applications are found vulnerable themselves to adversarial machine learning attacks. Existing studies argued that ensemble classifiers are more robust than a single classifier under evasion attacks due to more uniform weights produced on the basis of training data. In this paper, we investigate the problem in a more practical setting where attackers do not know the classifier details. Instead, attackers may acquire only a portion of the labeled data or a replacement dataset for learning the target decision boundary. In this case, we show that ensemble classifiers are not necessarily more robust under a least effort attack based on gradient descent. Our experiments are conducted with both linear and kernel SVMs on real datasets for spam filtering and malware detection.

Keywords: Adversarial machine learning · Ensemble classifiers
Evasion attacks

1 Introduction

Learning-based classifiers are increasingly accepted as a versatile tool for data-intensive security tasks [7,13,14,23–26]. They have been successfully deployed in many cyber security applications such as biometric authentication, intrusion detection, malware detection, spam filtering, detection of malicious Web page and so on [16,28,29,32,34,36]. In these applications, binary classifiers are essential for the task of discriminating a malicious instance from a legitimate one. To boost the accuracy performance, an ensemble approach may be adopted by combining multiple classifiers together to form an integrated output [9,12,17].

This work was supported in part by National Natural Science Foundation of China (61403324) and Dongguan University of Technology (KCYKYQD2017003).

H. Hacid et al. (Eds.): WISE 2018, LNCS 11233, pp. 96–110, 2018.
https://doi.org/10.1007/978-3-030-02922-7_7

Unlike in other applications where the operating environment is static, these security-related tasks involve intelligent adversaries who are able to analyse vulnerabilities of learning-based models and adapt their attacks in response to system outputs. In such an adversarial setting, conventional learning-based classifiers are found to be susceptible to evasion attacks among other security issues [2, 33]. In evasion attacks, the attacker is able to manipulate samples carefully to circumvent system detections. For example, in spam filtering, attackers can disguise their email behavior by misspelling bad words or adding normal words [35]. The PDFrate[1], a real-world deployed, well-known PDF malware detection system, can suffer substantial drops of detection accuracy when exposed to simple attacks [20].

The growing evidence of adversarial learning in different application domains has drawn significant attention of the research community in related fields [6, 8, 31]. There are several theoretical attempts to understand the rationale of inherent vulnerabilities in machine learning systems [6, 22]. It was pointed out that the success of attacks against learning algorithms crucially depends on the amount and type of knowledge exposed to an attacker [6]. Regarding the targeted system, there are four level of knowledge [6]: (1) the training data $\mathcal{D}$; (2) the feature set $\mathcal{X}$; (3) the learning algorithm f, along with the objective function $\mathcal{L}$ minimized during training; and, possibly, (4) the targeted model parameters $\mathbf{w}$. Thus, the attacker's knowledge can be characterized in terms of $\theta = (\mathcal{D}, \mathcal{X}, f, \mathbf{w})$.

Most of previously reported successful attacks assume that the attacker has full knowledge of the targeted model, known as the "white-box" attack [6] or the worst-case attack [5]. Recently, there are studies discussing evasion attacks with limited knowledge of θ, mainly focusing on improving the robustness of a *single* classifier in specific application domains [8, 35]. These methods argued that reducing the amount of knowledge available to the attacker or a proactive response to potential exploitation of such knowledge should provide adequate protection against adversarial data manipulation. Accordingly, several security evaluation measures were proposed to indicate the robustness of a learning-based classifier against evasion attacks. For example, hardness of evasion measure was defined as the average minimum number of features that have to be modified in a malicious sample to evade detection [35]. Another measure called *weight evenness* was proposed in [5] based on the observation that some features are highly discriminant than the others and if the adversary can identify them, e.g., by the associated weight values, it is not difficult to modify and get the malicious sample misclassified as a legitimate one. Under these security measures, it was shown that multiple classifier systems by averaging simpler classifiers such as classic SVMs can be exploited to improve the robustness against evasion attacks because more evenly distributed feature weights should require the adversary to manipulate a higher number of features [4, 5, 27].

In this paper, we re-investigate the security evaluation problem from another perspective. Our intuition is that, with small subsets or even zero knowledge of the target training data $\mathcal{D}$, ensemble learning may lead to a surrogate classifier

[1] http://pdfrate.com/.

with less variation thus more accurate estimation of gradients when approximating the target decision boundary. Accordingly, we introduce a new security measure called *Gradient Correlation* to evaluate the similarity of gradient estimation between the surrogate and the targeted systems. We build the ensemble on linear and kernel SVMs with averaging and voting strategies, respectively. Our experimental results on real-world datasets indicate that, unlike expected previously, ensembling base classifiers such as linear SVMs do *not* necessarily improve the robustness of classifiers against evasion attacks under all circumstances.

2 Related Work

The problem of evasion attack at test time has been considered in the literature [6,22]. Most of the studies are focusing on individual classifiers, either convex-inducing classifiers including SVM with simple decision functions [2,35] or more complex neural networks [10,33], and defence methods in specific application domains. There are relatively fewer discussions on the security of ensemble classifiers. Ensemble classifiers were originally proposed to improve the classification accuracy by combining multiple weak classifiers to cope with more complex hypothesis and nonlinear decision boundaries [11,30]. Because the feature weights can be more evenness through the combination of multiple single classifiers and the decision boundary is hard to find [5,27]. From this perspective, previous studies showed that ensemble classifiers are more robust than a single classifier under white-box attacks with full knowledge of the targeted system [4,5].

The white-box attack is extended in [5] to more general attacks on multiple classifiers. The paper proposed two limited knowledge attacks by assuming the feature set in an attacker's hand is not the same as the original one $\mathcal{X}$. To simulate the limited knowledge scenarios, the feature set $\hat{\mathcal{X}}$ assumed available to the attacker was generated by shuffling half or all features in $\mathcal{X}$ at random. However, it is not clear how attackers can obtain shuffled features in practical classifier systems. The security evaluation therein is based on the weight evenness which considers a classifier is more robust if the weights in $\mathbf{w}$ are more evenly distributed so that the attacker cannot easily discover the most salient features.

In more realistic settings, the attacker's knowledge is limited by restricting the training data $\mathcal{D}$ available to the attacker [2,3,35]. It was assumed that the available $\hat{\mathcal{D}}$ is either subsets of the original $\mathcal{D}$ or a surrogate collected from alternative sources with the same data distribution as the target. Gradient descent attacks were proposed to increase the probability of successful evasion by exploiting knowledge of the (estimated) decision boundary gained from the discriminant function of the target classifier. However, again, these methods are evaluated on a single SVM rather than ensemble classifiers and the security measures are in general based on the hardness of evasion and the weight evenness.

Gradient information was exploited to attack tree ensemble classifiers in [15,27]. It was shown that both gradient boosted trees and random forests are extremely susceptible to evasions under white-box attacks with full knowledge of the original training dataset $\mathcal{D}$ [15]. On the other hand, the study in [27] shows that an ensemble of classifiers, either decision trees or SVMs, can be used to detect evasion attacks by checking diversity in the ensembles themselves. Due to diversity and adjusting the voting threshold accordingly, ensemble trees are considered more robust than a single classifier against evasion attacks [27].

3 Background

Before proceeding to the proposed approach, here we briefly review SVM-based classifiers and introduce relevant notations used in this paper followed by the general formulation of evasion attacks.

3.1 Support Vector Machines

Given a training dataset $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$, the primal problem of linear SVM is to solve the following quadratic program:

$$\begin{aligned} &\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ &s.t. \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0, \end{aligned} \tag{1}$$

where $\mathbf{w}$ is the weight vector, b is the displacement, $\{\xi_i\}$ are slack variables defining the soft margin [11], and the regularization term C tunes the trade-off between the classification error and margin maximization. Once the parameters are solved, the discriminant function defining the decision boundary is given by

$$g_{linear}(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b. \tag{2}$$

The linear SVM can be extended to a more complex feature space by introducing some kernel function on $\mathbf{x}$. The discriminant function written in its dual form is

$$g_{kernel}(\mathbf{x}) = \sum_{i=1}^{N} a_i y_i K(\mathbf{x}, \mathbf{x}_i) + b. \tag{3}$$

where $\{\alpha_i\}$ are Lagrange multipliers in KKT conditions. We consider a kernel SVM with radial-basis function (RBF) where $K(\mathbf{x}, \mathbf{x}_i) = exp(-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2)$.

To build ensemble SVMs, we follow previous studies [5,27] by bagging subspace features as it was reported to be more effective than bagging training subsets [27]. We adopt two aggregation methods, namely *averaging* and *voting*, to make the final decision over classification results of the independently trained SVMs. The voting method aggregates the results of individual base classifiers by majority votes. The averaging method has discriminant function as

$$\begin{aligned} g_{ensem}(\mathbf{x}) &= \frac{1}{M}\sum_{m=1}^{M}\Big[\sum_{i=1}^{N} a_i^m y_i^m K(\mathbf{x}^m, \mathbf{x}_i^m) + b^m\Big] \\ &= \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N} a_i^m y_i^m K(\mathbf{x}^m, \mathbf{x}_i^m) + b^{avg} \end{aligned} \tag{4}$$

where M is the number of base classifiers and b^{avg} is the averaged displacement. Note that the linear kernel is given by $K(\mathbf{x}^m, \mathbf{x}_i^m) =< \mathbf{x}^m, \mathbf{x}_i^m >$ in this case.

3.2 Evasion Attacks

In this attack mode, the attacker's goal is to have a malicious sample misclassified as benign at test time. To this end, the attacker needs to know the decision boundary or similar boundary of the targeted system. Without loss of generality, suppose that the discriminant function $g(\mathbf{x}) > 0$ for detecting a malicious sample $\mathbf{x}$ otherwise passing a legitimate one. The attack rationale is to find a sample $\mathbf{x}'$ that yields $g(\mathbf{x}) < 0$ by minimally manipulating the initial malicious sample $\mathbf{x}$, where the amount of manipulations is characterized by some distance function $d(\mathbf{x}, \mathbf{x}')$ in feature space. This general formula can be written as [35]

$$A(\mathbf{x}) = \text{argmin}_{\mathbf{x}'} \;\; d(\mathbf{x}, \mathbf{x}'), \quad s.t. \;\; g(\mathbf{x}') < 0. \tag{5}$$

In the case when the features are Boolean as used in the paper, $d(\cdot, \cdot)$ corresponds to the Hamming distance which indicates the number of features that must be added (i.e., flipped from 0 to 1) or deleted (i.e., flipped from 1 to 0) from the initial attack sample $\mathbf{x}$.

Recall that the attacker's knowledge regarding the target classifier can be characterized in terms of $\theta = (\mathcal{D}, \mathcal{X}, f, \mathbf{w})$ as introduced in Sect. 1. It should be noted that only in white-box attacks the target $g(\mathbf{x})$ will be known to the attacker and the required $d(\mathbf{x}, \mathbf{x}')$ is minimal. In more realistic settings, the attacker can only obtain an estimated $\hat{g}(\mathbf{x})$ by constructing an approximated learner $\hat{f}$ with estimated parameters $\mathbf{w}$ trained on a surrogate training set $\hat{\mathcal{D}} = \{(\hat{\mathbf{x}}_i, \hat{y}_i)\}_{i=1}^{N_s}$ of N_s samples. The surrogate training data may be collected by the adversary via sniffing network traffic or augmenting from other sources. Sometimes, the attacker may obtain some true labels and/or subsets of the original training samples. In any case, there must be bias in the estimation should there be knowledge discrepancy about the target. It is intuitive that a better approximation of the surrogate will make the attacker manipulate fewer features to evade the detection and thus less secure/robust of the target classifier against the attack.

4 The Proposed Attack Approach

In this paper, we evaluate the robustness of ensemble SVMs under gradient attacks by assuming limited knowledge of the target's training data available to an attacker. As the surrogate dataset $\hat{\mathcal{D}}$ differs from $\mathcal{D}$, more or less there must be a distribution drift in attacker's learning the surrogate classifier $\hat{f}$ for simulating attacks. It is anticipated that the more drift from $\mathcal{D}$ the more bias in the learner estimate $\hat{f}$ and parameter estimates in $\hat{\mathbf{w}}$ that are trained on the surrogate dataset $\hat{\mathcal{D}}$. In any case, if the attacker can obtain from θ a better discriminant function $\hat{g}(\mathbf{x})$ that closely approximates the target $g(\mathbf{x})$ he is able to manipulate an evasion sample more effectively with fewer efforts.

Accordingly, we consider two attack scenarios with respect to the attacker's knowledge on $\mathcal{D}$. The first scenario is called *the subset scenario* which assumes the attacker knows a subset of training data, i.e., $\hat{\mathcal{D}} \subset \mathcal{D}$. We gradually vary the size of $\hat{\mathcal{D}}$ to evaluate the classifier's robustness under evasion attacks with respect to the distribution drift between the surrogate and the target datasets. In particular, when $\hat{\mathcal{D}} = \mathcal{D}$ it is equivalent to the "white-box" attack in which the surrogate classifier can be regarded as a reproduction of the target and $\hat{g}(\mathbf{x}) = g(\mathbf{x})$ yields the worst-case scenario for evasion with least efforts. The second scenario is called *the surrogate data scenario* which assumes the attacker does not know any instance of the original training data but is able to collect a surrogate dataset $\hat{\mathcal{D}}$ resemble the data distribution of $\mathcal{D}$.

To solve the optimization problem in (5), we assume $\hat{g}(\mathbf{x})$ to be differentiable almost everywhere and adopt the gradient descent attacks which were shown to be effective against single SVMs [2,8]. For classifiers with binary features, the procedure of gradient descent attacks can be found as follows. Firstly, the gradients in $\mathbf{g}$ have to be sorted in descending order of their absolute values, and feature values $\mathbf{x}$ of the malicious sample have to be sorted accordingly. We denote the sorted gradients as $g_1, g_2, ..., g_n$, and the features as $x_1, x_2, ..., x_n$, where $|g_1| \geq |g_2| \geq ... \geq |g_n|$. Then, for $i = 1, 2, ..., d$:

- If $g_i > 0$ and $x_i = 1$, set x_i to 0;
- If $g_i < 0$ and $x_i = 0$, set x_i to 1 (if it is possible);
- otherwise, x_i is left unmodified.

The following subsections explicitly give the gradient formular of discriminant functions for the comparing classifiers. We also propose a new evaluation measure called *gradient correlation* to indicate the similarity of gradients between the surrogate and the targeted systems for constructing evasion samples offline with limited knowledge on θ.

4.1 Gradients of Single SVMs

The most important thing in gradient decent attack is to know the gradient of a classifier. In this section, we give the gradient of single SVMs. The gradient of ensemble SVMs is given in Sect. 4.2.

The gradient of linear-SVM is quite simple which is

$$\nabla g(\mathbf{x}) = \mathbf{w} \tag{6}$$

For kernel-SVM, the gradient is

$$\nabla g(\mathbf{x}) = \sum_{i=1}^{N} a_i y_i \nabla K(\mathbf{x}, \mathbf{x}_i) \tag{7}$$

For RBF kernel, $\nabla K(\mathbf{x}, \mathbf{x}_i) = -2\gamma exp(-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2)(\mathbf{x} - \mathbf{x}_i)$, so the gradient of RBF-SVM is

$$\nabla g(\mathbf{x}) = \sum_{i=1}^{N} a_i y_i [-2\gamma exp(-\gamma\|\mathbf{x} - \mathbf{x}_i\|^2)(\mathbf{x} - \mathbf{x}_i)] \tag{8}$$

4.2 Gradients of Ensemble SVMs

For averaging linear-SVMs, we can see that its discriminant function is still a linear function. Its gradient is just like linear-SVM which is the averaged weight vector $\mathbf{w}^{avg}$. We use the same gradient in voting linear-SVMs.

For averaging RBF-SVMs, the gradient is

$$\nabla g(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \sum_{i=1}^{N} a_i^m y_i^m \nabla K(\mathbf{x}^m, \mathbf{x}_i^m) \tag{9}$$

where

$$\nabla K(\mathbf{x}, \mathbf{x}_i) = -2\gamma exp(-\gamma \|\mathbf{x}^m - \mathbf{x}_i^m\|^2)(\mathbf{x}^m - \mathbf{x}_i^m) \tag{10}$$

This gradient is also used by voting RBF-SVMs.

4.3 The Gradient Correlation Measure

Kolcz and Teo [18] proposed a measure to evaluate the weight evenness of a classifier which is

$$F(k) = \frac{\sum_{i=1}^{k} w_i}{\sum_{j=1}^{n} w_j} \tag{11}$$

where $k = 1, 2, ..., n$, w_i is the absolute value of its original weight, and $w_1, w_2, ..., w_n$, denote the weights sorted in descending order of their absolute value.

However, this measure is not a scalar. The weight distribution is most even when every weight is identical, which corresponds to $F(k) = k/n$. The most uneven distribution is when only one weight is not zero where $F(k) = 1$ for each k. Accordingly, Biggio et al. [5] proposed a normalized measure $(E) \in [0, 1]$, called weight evenness, based on $F(k)$.

The weight evenness measure was used in addition to hardness of evasion to indicate the robustness of a linear classifier. It is worth noting that in more practical settings the weight eveness measured on a surrogate classifier is not necessarily the same as that on the target model. To address this problem, we propose a more universal measure to evaluate the similarity of gradient estimation between the surrogate and the targeted systems. The gradient correlation (GC) measure is given by:

$$GC = \frac{\sum_{k=1}^{n} C(k)}{n} \tag{12}$$

where

$$C(k) = \frac{\sum_{i=1}^{k} g_i'}{\sum_{i=1}^{k} g_i} \tag{13}$$

Let $\mathbf{g}^+$ denotes the original gradient vector of the targeted system, $\mathbf{g}$ is the vector which sorted $|\mathbf{g}^+|$ in descending order, i.e., $g_1 \geq g_2 \geq ... \geq g_n$. $\mathbf{g}'$ is the gradient vector of surrogate system with the absolute gradient value of targeted system

Algorithm 1. Gradient Correlation

```
Input: $[\mathbf{g}^{+}, \mathbf{f}^{+}]$, $\mathbf{g}^{+}$: the original gradient vector of targeted system, $\mathbf{f}^{+}$: the features used in the targeted system; $[\mathbf{g}^{-}, \mathbf{f}^{-}]$, $\mathbf{g}^{-}$: the original gradient vector of surrogate system, $\mathbf{f}^{-}$: the features used in the surrogate system, $\mathbf{f}^{-} \subseteq \mathbf{f}^{+}$; $n$: the number of features used in the targeted system; $m$: the number of features used in the surrogate system.
Output: $GC$
$[\mathbf{g}, \mathbf{f}] \leftarrow$ sort $|\mathbf{g}^{+}|$ in descending order;
$[\mathbf{g}^{*}, \mathbf{f}^{*}] \leftarrow$ sort $|\mathbf{g}^{-}|$ in descending order;
$j \leftarrow 1$;
while $j \leq m$ do
    $p \leftarrow$ find the position of $f_j^{*}$ in $\mathbf{f}$ if exist, otherwise $p \leftarrow 0$;
    if $p > 0$ then
        $g_j' \leftarrow g_p$
    else
        $g_j' \leftarrow 0$
    end if
    $j \leftarrow j + 1$;
end while
if $m < n$ then
    $g_j' \leftarrow 0, j = m, m+1, ..., n$;
end if
$C(k) = \frac{\sum_{i=1}^{k} g_i'}{\sum_{i=1}^{k} g_i}, k = 1, 2, ..., n$;
$GC = \frac{\sum_{k=1}^{n} C(k)}{n}$.
```

for the same features between the targeted and the surrogate systems. n is the number of features used in the targeted system. The detailed procedure is given by Algorithm 1.

Form Algorithm 1, we can see that $GC \in [0, 1]$, $GC = 0$ and $GC = 1$ correspond respectively to the most uncorrelated and the most correlated gradient distribution. Larger GC means attacker knows more about the gradient of targeted system and the attacks are more effective.

5 Evaluation

In this section, we evaluate the robustness of ensemble SVMs and a single SVM trained on the same dataset for spam email filtering and malware detection tasks. In the subset attack scenario, we gradually increase the size of $\hat{\mathcal{D}}$ by 10%, 20%, ..., 100% of the original training dataset. The samples in $\hat{\mathcal{D}}$ are randomly selected from $\mathcal{D}$. In the second attack scenario of surrogate datasets, we also vary the amount of attacker's knowledge by portions but the training data is from an alternative source rather than the targeted system. Each experiment was run 30 times and the results were averaged to produce the figures.

It is worth noting that for ensemble learning, the surrogate and the targeted systems are different in each run for the features were selected randomly, albeit the target was trained on exactly the same dataset. Each ensemble classifier contains 100 independent base classifiers. As suggested by Ho [12], using half of the features resulting in the best or very close to the best accuracies. We set the feature bagging ratio to 50% for all ensemble classifiers.

In our evaluation tasks, we do not restrict the attack ability which means the attacker can manipulate a malicious sample using whatever computing and time resources needed based on the available knowledge. For security evaluation, we adopt both the conventional hardness of evasion [35] and the proposed gradient correlation measures. The following shows our results performed on two real application datasets.

5.1 Spam Email Filtering

Experimental Setup: The PU3 dataset was considered in spam email classification task [1,21]. There are 11 subdirectories in PU3 (part1, ..., part10, unused) and the first 10 subdirectories, which consists of 1820 spam and 3310 legitimate emails, were used in our experiments. In PU3 dataset, the messages are "encoded" with digital numbers. For the evaluation task, we first extracted words (i.e., features) from emails in the first 5 subdirectories and more than 30,000 features were extracted. Then we reconstructed every email with binary features, which is 1 or 0 represent a feature presence or absence respectively in an email. For keeping computational complexity manageable, we used a feature selection approach, information gain [19], to reduce the feature space to 200 features without loss the classification accuracy significantly.

After turned every email in PU3 to the new feature space, we split the 4130 emails into 3 different subsets. Subset 1 included 608 spam and 769 legitimate emails, which was used as the training dataset. Subset 2 included 604 spam and 773 legitimate emails, which was used as the surrogate dataset. Subset 3 included 608 spam and 768 legitimate emails, which was used as the test dataset. Each email in 3 subsets was different. The training data was used by the targeted systems to train classifiers. The evaluation was carried out on the test data. For linear-SVM and linear-SVM ensemble, the SVM regularization parameter C was set to $C = 1$. For RBF-SVM and RBF-SVM ensemble, we set the SVM regularization parameter $C = 100$ and the kernel parameter $\gamma = 0.01$.

Experimental Results: Table 1 shows classification accuracies achieved by the targeted systems. We observe that ensemble classifiers improved the classification accuracy as expected and that the base classifier using RBF-SVMs slightly outperforms that using linear-SVMs. Figure 1 shows that, under the subset attack scenario, ensemble SVMs are more robust than a single SVM when there is a significant amount (more than 30%) of the original training data are exposed to an attacker. When the available data is reduced to less than 30% ensemble classifiers become more susceptible to evasion for both linear-based and RBF-based SVMs. Under the surrogate data scenario, which is shown on the right side of

Fig. 1, the amount of surrogate data is not as critical as that in the subset scenario. In this case, the ensemble classifiers are always easier to be compromised by manipulating fewer features on average for evading the target classifier.

Table 1. Classification accuracy

Single Linear-SVM	Averaging Linear-SVMs	Voting Linear-SVMs	Single RBF-SVM	Averaging RBF-SVMs	Voting RBF-SVMs
0.9440	0.9565	0.9573	0.9448	0.9599	0.9592

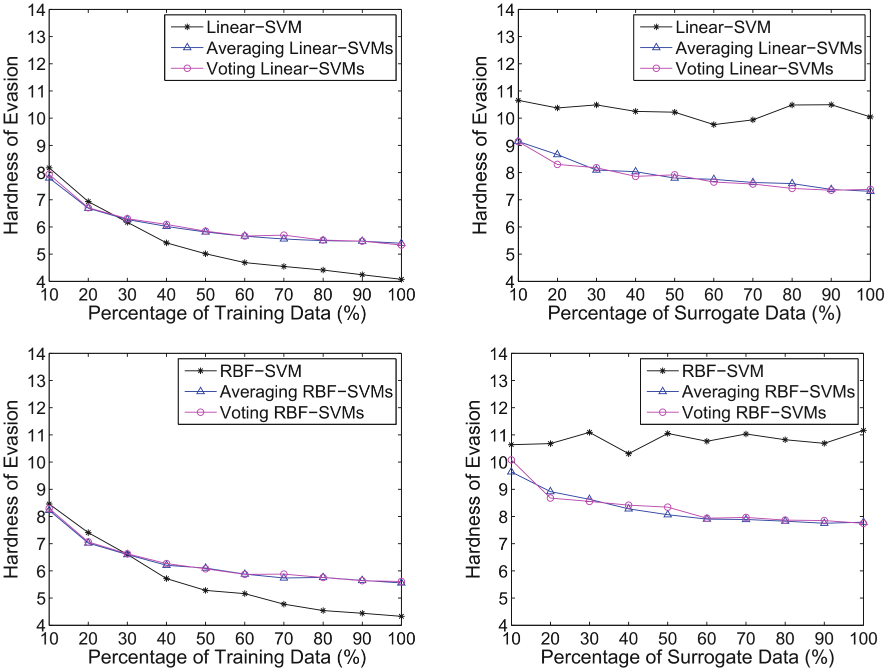

Fig. 1. Hardness of evasion (i.e. average minimum number of modified words to let all spam emails classified as legitimate) in the subset scenario (left) and the surrogate data scenario (right).

Figure 2 plots gradient correlation measures for the two attack scenarios. It can be seen that ensemble SVMs always have higher gradient correlation scores than single SVMs, which supports the observation in Fig. 1. A higher correlation score indicates a higher similarity level between gradient estimates of the surrogate and the targeted systems, and thus more prone to be compromised by evasion attack. In the subset data scenario, the gradient correlation score rises with an increasing percentage of training data exposed to the attacker for both single and ensemble classifiers. In the surrogate data scenario, however, the change is rather flat as the distribution drift is determined by the nature of the two data sources rather than the surrogate data size.

Another interesting observation in Fig. 2 is that an attacker can use less than 50% of the original training dataset to build a surrogate system of ensemble classifiers that closely mimics the targeted system performance and acquire a proactive response to a malicious input. Whereas for a single SVM system it will require more than 80% of the target training dataset to build a resembled surrogate system.

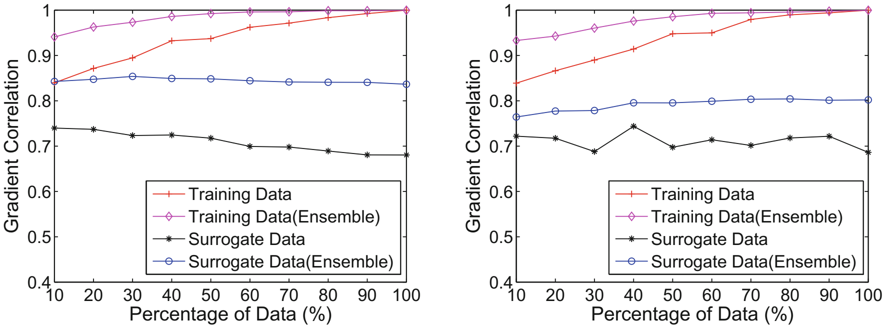

Fig. 2. Gradient correlation GC for linear-based classifiers (left) and RBF-based classifiers (right).

5.2 Malware Detection in PDF

Experimental Setup: The other real-world task we considered is malware detection. The PDF dataset used in [2] was considered in these experiments. In their released library adversarialib v1.0[2], there are 514 malicious samples and 486 benign samples. The feature space includes 114 features (keywords) and the feature value $x \in [0, 1]$ which is the occurrence of a given keyword in a PDF divided by 100. For less confusing, we simply modified the feature value to 1 if the original value $x > 0$, or 0 if the original value $x = 0$. In this case, $x = 1$ means a given keyword is present in a PDF, and $x = 0$ means it is absent.

As discussed in [2], it is hard to remove an embedded object (keywords) from a PDF file without corrupting its structure. But inserting new objects (keywords) through adding a new version to a PDF file is quite easy [2,35]. In our experiments, keywords only can be added cannot be removed which means a feature value only can be modified from 0 to 1, cannot be modified from 1 to 0. For this reason, only the original gradient value $g_i < 0$ need to be considered when calculating GC.

Following the experimental setup used in spam email filtering task, we also split the 1000 samples into 3 different subsets. Subset 1 included 162 malicious and 171 benign samples, which was used as the training dataset.

[2] http://pralab.diee.unica.it/.

Subset 2 included 182 malicious and 152 benign samples, which was used as the surrogate dataset. Subset 3 included 170 malicious and 164 benign samples, which was used as the test dataset. Also, each sample in 3 subsets was different. For linear-SVM and linear-SVM ensembles, the SVM regularization parameter C was set to $C = 1$. For RBF-SVM and RBF-SVM ensembles, we set the SVM regularization parameter $C = 100$ and the kernel parameter $\gamma = 0.01$.

Experimental Results: For this dataset, the security measures in terms of hardness of evasion are very close between the single and the ensemble classifiers in Fig. 3. Nevertheless, it can still be observed that ensemble SVMs tend to require fewer features modified on average for evasion when an attacker has less knowledge on $\mathcal{D}$. This is more obvious by the gradient correlation scores shown in Fig. 4 where gradient estimations are more accurate by ensemble classifiers for both linear- and kernel-based SVMs. This indicates that ensemble SVMs are more susceptible to gradient descent attacks with limited knowledge. In all cases, there is no much difference in security performance between the two aggregation methods of averaging and voting.

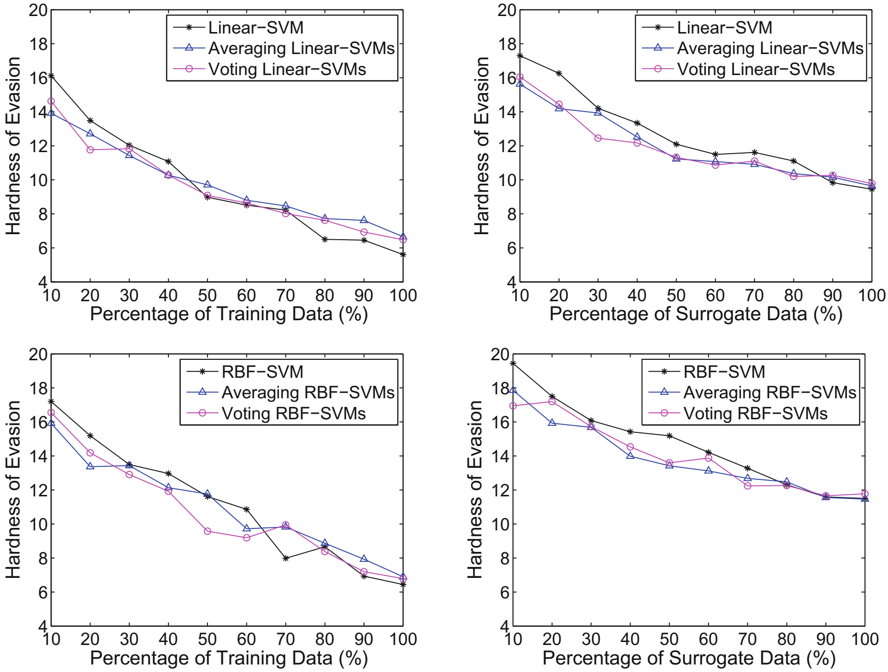

Fig. 3. Hardness of evasion (i.e. average minimum number of added keywords to make every malicious PDF classified as benign) in the subset scenario (left) and the surrogate scenario (right).

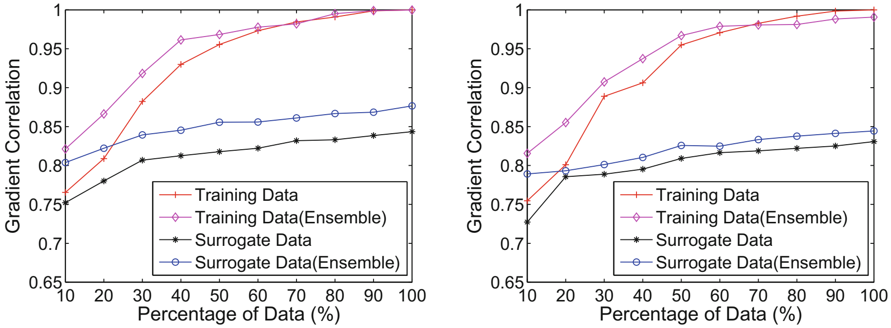

Fig. 4. Gradient correlation *GC* for linear-based classifiers (left) and RBF-based classifiers (right).

6 Conclusion and Future Work

In this paper, we investigated the robustness of ensemble classifiers comparing with single classifiers under evasion attacks with limited knowledge. We propose a new security evaluation measure called gradient correlation to indicate the accuracy of gradient estimation when building a surrogate system for simulating proactive responses to malicious samples. Our experimental results showed that ensemble classifiers require much less knowledge of the original training dataset to build a surrogate classifier closely resembled the targeted system and thus more susceptible to evasion attacks with limited knowledge in more practical scenarios.

Our future work will focus on finding novel defence methods for ensemble approaches. We also intend to extend the proposed gradient correlation measure to study the security performance of other learning-based classifiers.

References

1. Androutsopoulos, I., Paliouras, G., Michelakis, E.: Learning to filter unsolicited commercial e-mail (2004)
2. Biggio, B., et al.: Evasion attacks against machine learning at test time. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8190, pp. 387–402. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_25
3. Biggio, B.: Security evaluation of support vector machines in adversarial environments. In: Ma, Y., Guo, G. (eds.) Support Vector Machines Applications, pp. 105–153. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02300-7_4
4. Biggio, B., Fumera, G., Roli, F.: Evade hard multiple classifier systems. In: Okun, O., Valentini, G. (eds.) Applications of Supervised and Unsupervised Ensemble Methods, pp. 15–38. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03999-7_2

5. Biggio, B., Fumera, G., Roli, F.: Multiple classifier systems for robust classifier design in adversarial environments. Int. J. Mach. Learn. Cybern. **1**(1–4), 27–41 (2010)
6. Biggio, B., Roli, F.: Wild patterns: ten years after the rise of adversarial machine learning. arXiv Preprint (2017). http://arxiv.org/abs/1712.03141
7. Cheng, K., et al.: Secure k-NN query on encrypted cloud data with multiple keys. IEEE Trans. Big Data **1**, 1–1 (2015)
8. Demontis, A., et al.: Yes, machine learning can be more secure! A case study on android malware detection. IEEE Trans. Dependable Secur. Comput. (2017, in press). https://ieeexplore.ieee.org/document/7917369
9. Dong, Y.S., Han, K.S.: Boosting SVM classifiers by ensemble. In: The 14th International Conference on World Wide Web, pp. 1072–1073, WWW 2005. ACM (2005)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: The International Conference on Learning Representations, ICLR 2015 (2015)
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning, 2nd edn. Springer, New York (2009)
12. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. **20**(8), 832–844 (1998)
13. Kabir, E., Mahmood, A., Wang, H., Mustafa, A.: Microaggregation sorting framework for k-anonymity statistical disclosure control in cloud computing. IEEE Trans. Cloud Comput. (2015, in press). https://ieeexplore.ieee.org/document/7208829
14. Kabir, M.E., Wang, H., Bertino, E.: A role-involved purpose-based access control model. Inf. Syst. Front. **14**(3), 809–822 (2012)
15. Kantchelian, A., Tygar, J., Joseph, A.: Evasion and hardening of tree ensemble classifiers. In: International Conference on Machine Learning, pp. 2387–2396 (2016)
16. Khalil, F., Li, J., Wang, H.: An integrated model for next page access prediction. Int. J. Knowl. Web Intell. **1**(1–2), 48–80 (2009)
17. Kim, H.C., Pang, S., Je, H.M., Kim, D., Bang, S.Y.: Constructing support vector machine ensemble. Pattern Recogn. **36**(12), 2757–2767 (2003)
18. Kołcz, A., Teo, C.H.: Feature weighting for improved classifier robustness. In: Sixth Conference On Email and Anti-spam, CEAS 2009 (2009)
19. Kolter, J.Z., Maloof, M.A.: Learning to detect malicious executables in the wild. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 470–478. ACM (2004)
20. Laskov, P., et al.: Practical evasion of a learning-based classifier: a case study. In: 2014 IEEE Symposium on Security and Privacy (SP), pp. 197–211. IEEE (2014)
21. Mujtaba, G., Shuib, L., Raj, R.G., Majeed, N., Al-Garadi, M.A.: Email classification research trends: review and open issues. IEEE Access **5**, 9044–9064 (2017)
22. Papernot, N., Mcdaniel, P., Sinha, A., Wellman, M.: SoK: Towards the science of security and privacy in machine learning. arXiv Preprint, pp. 1–19 (2016). http://arxiv.org/abs/1611.03814
23. Peng, M., Zeng, G., Sun, Z., Huang, J., Wang, H., Tian, G.: Personalized app recommendation based on app permissions. World Wide Web **21**(1), 89–104 (2018)
24. Shah, Z., Mahmood, A.N., Barlow, M., Tari, Z., Yi, X., Zomaya, A.Y.: Computing hierarchical summary from two-dimensional big data streams. IEEE Trans. Parallel Distrib. Syst. **29**(4), 803–818 (2018)
25. Shen, Y., Zhang, T., Wang, Y., Wang, H., Jiang, X.: Microthings: a generic iot architecture for flexible data aggregation and scalable service cooperation. IEEE Commun. Mag. **55**(9), 86–93 (2017)

26. Shu, J., Jia, X., Yang, K., Wang, H.: Privacy-preserving task recommendation services for crowdsourcing. IEEE Trans. Serv. Comput. (2018, in press). https://ieeexplore.ieee.org/document/8253516
27. Smutz, C., Stavrou, A.: When a tree falls: using diversity in ensemble classifiers to identify evasion in malware detectors. In: NDSS (2016)
28. Sun, X., Li, M., Wang, H., Plank, A.: An efficient hash-based algorithm for minimal k-anonymity. In: Proceedings of the Thirty-First Australasian Conference on Computer Science, vol. 74, pp. 101–107. Australian Computer Society, Inc. (2008)
29. Sun, X., Wang, H., Li, J., Zhang, Y.: Injecting purpose and trust into data anonymisation. Comput. Secur. **30**(5), 332–345 (2011)
30. Vapnik, V.: The Nature of Statistical Learning, 1st edn. Springer, New York (1999)
31. Wang, G., Wang, T., Zheng, H., Zhao, B.Y.: Man vs. machine: practical adversarial detection of malicious crowdsourcing workers. In: USENIX Security Symposium, pp. 239–254 (2014)
32. Wang, H., Cao, J., Zhang, Y.: Ticket-based service access scheme for mobile users. In: Australian Computer Science Communications, vol. 24, pp. 285–292. Australian Computer Society, Inc. (2002)
33. Xu, W., Qi, Y., Evans, D.: Automatically evading classifiers. In: Proceedings of the 2016 Network and Distributed Systems Symposium (2016)
34. Yi, X., Sun, H., Jafar, S.A., Gesbert, D.: Tdma is optimal for all-unicast dof region of tim if and only if topology is chordal bipartite. IEEE Trans. Inf. Theory **64**(3), 2065–2076 (2018)
35. Zhang, F., Chan, P.P., Biggio, B., Yeung, D.S., Roli, F.: Adversarial feature selection against evasion attacks. IEEE Trans. Cybern. **46**(3), 766–777 (2016)
36. Zhang, Y., Shen, Y., Wang, H., Zhang, Y., Jiang, X.: On secure wireless communications for service oriented computing. IEEE Trans. Serv. Comput. **11**(2), 318–328 (2018)