


Article

Improving Adversarial Robustness of Ensemble Classifiers by Diversified Feature Selection and Stochastic Aggregation

Fuyong Zhang *, Kuan Li  and Ziliang Ren

School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China; likuan@dgut.edu.cn (K.L.); renzl@dgut.edu.cn (Z.R.)

* Correspondence: zhangfy@dgut.edu.cn

Abstract: Learning-based classifiers are found to be vulnerable to attacks by adversarial samples. Some works suggested that ensemble classifiers tend to be more robust than single classifiers against evasion attacks. However, recent studies have shown that this is not necessarily the case under more realistic settings of black-box attacks. In this paper, we propose a novel ensemble approach to improve the robustness of classifiers against evasion attacks by using diversified feature selection and a stochastic aggregation strategy. Our proposed scheme includes three stages. Firstly, the adversarial feature selection algorithm is used to select a feature each time that can trade-off between classification accuracy and robustness, and add it to the feature vector bank. Secondly, each feature vector in the bank is used to train a base classifier and is added to the base classifier bank. Finally, m classifiers from the classifier bank are randomly selected for decision-making. In this way, it can cause each classifier in the base classifier bank to have good performance in terms of classification accuracy and robustness, and it also makes it difficult to estimate the gradients of the ensemble accurately. Thus, the robustness of classifiers can be improved without reducing the classification accuracy. Experiments performed using both Linear and Kernel SVMs on genuine datasets for spam filtering, malware detection, and handwritten digit recognition demonstrate that our proposed approach significantly improves the classifiers' robustness against evasion attacks.

Keywords: adversarial machine learning; evasion attacks; classifier robustness; ensemble classifiers; gradient correlation

MSC: 68T50



Citation: Zhang, F.; Li, K.; Ren, Z. Improving Adversarial Robustness of Ensemble Classifiers by Diversified Feature Selection and Stochastic Aggregation. *Mathematics* **2024**, *12*, 834. <https://doi.org/10.3390/math12060834>

Academic Editors: Mingbo Zhao, Haijun Zhang and Zhou Wu

Received: 7 February 2024

Revised: 7 March 2024

Accepted: 11 March 2024

Published: 12 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid expansion of global data volumes, machine learning has become extensively adopted and serves as a principal tool for data analysis across various sectors such as transportation, computer vision, finance, and security [1,2]. However, a widely acknowledged truth is that machine learning models are susceptible to adversarial examples. Attackers can probe machine learning models and maliciously manipulate their inputs to mislead the recognition outcomes [3,4]. For instance, a machine learning-powered malware detector ingests features extracted from Portable Executable (PE) files and categorizes a test case as either malware or benign software. Here, an adversary could tamper with a malware input by introducing barely perceptible perturbations, thereby tricking the detector into classifying it as benign software [5].

Adversarial samples become serious security threats for many machine learning-based systems, e.g., by extracting sufficient knowledge to exploit Google's phishing pages filter [6] and PDFRATE system-based attack [7].

Various countermeasures against evasion attacks have been proposed. Previous research demonstrates that dimensionality reduction can be an effective defense mechanism against evasion attacks [8]. Another defensive strategy is adversarial training [9], which

incorporates adversarial examples into the training process. Some studies indicate that defensive distillation can be leveraged to enhance the robustness of neural networks against adversarial examples [10].

Furthermore, ensemble methods have been put forward as defense mechanisms. Intuitively, it is more challenging for an attacker to undermine a group of models than a single one. Strauss et al. [11] argue that compromising individual classifiers does not necessarily imply that other classifiers within the ensemble will also succumb to the attack. Their experimental findings substantiate that ensemble methods can indeed act as defensive strategies against evasion attacks. Tramèr et al. [12] propose an 'Ensemble Adversarial Training' approach, aimed at training a robust classification model that is resilient to such attacks. Previous research has also demonstrated that ensemble SVMs are generally more robust against evasion attacks compared to a single SVM [13].

However, recent studies have indicated that conventional ensemble learning methods may not necessarily enhance the robustness of learning models. Zhang et al. [14] and Kantchelian et al. [15] have demonstrated that tree ensembles can be more prone to evasion attacks than SVM classifiers, whether they are single or an ensemble. Zhang et al. [16] noted that ensemble SVMs are not necessarily more robust against evasion attacks compared to single SVMs. These studies show that it is still possible for attackers to launch evasion attacks on ensemble models. In 2019, Pang et al. [17] proposed a novel method, Adaptive Diversity Promoting (ADP), which improves the robustness of deep ensembles by promoting diversity in non-maximal predictive scores while keeping the maximal (most likely) prediction consistent with the true label for members in the ensemble. It reveals that promoting the diversity of ensemble models can improve their robustness against adversarial samples.

The main limitations of Linear and Kernel SVM classifiers lie in the fact that their designs do not inherently account for security aspects, rendering them less robust against evasion attacks. Zhang et al. [18] proposed an adversarial feature selection approach that incorporates security considerations into the training process, thereby improving the robustness of single SVM classifiers. Despite this improvement, single SVM classifiers generally exhibit lower classification accuracy when compared to ensemble methods. Smutz et al. [13] advocated for the use of ensemble SVMs to against evasion attacks; however, Zhang et al. [16] pointed out that ensemble SVMs are not necessarily more robust than single SVMs against evasion attacks in practical settings.

Consequently, we propose to improve the robustness of ensemble classifiers by employing diversified feature selection and stochastic aggregation, thus aiming to create a more resilient solution against adversarial threats. The underlying idea of our approach is to build ensemble classifiers not based on the combination of *weaker* classifiers, but the ensemble of classifiers that are robust against adversarial samples. We exploit the adversarial feature selection approach [18] to train the base classifiers because this method makes a trade-off between the generalization capability and its security against evasion attacks. Experimental results on real-world datasets demonstrate that our approach significantly enhances the classifiers' robustness against adversarial examples while maintaining comparable accuracy levels even when there is no attack.

The main contributions are summarized as follows:

- We propose a novel approach to train base classifiers using sequential feature selection, wherein each base classifier encompasses all the features of the preceding trained classifier and subsequently selects an additional new feature.
- We introduce stochastic aggregation, in which m classifiers are randomly selected from the base classifier bank to participate in decision-making, which not only improves the classification accuracy, but also improves the robustness against evasion attacks.
- We re-investigate the security evaluation problem, and update the gradient correlation measure to extend it to be suitable for any real number feature.

- To evaluate the performance of the proposed model, we launched lots of experiments, and the experimental results demonstrate that the proposed ensemble model can improve the robustness against evasion attacks.

2. Related Work

2.1. Evasion Attacks

The susceptibility of machine learning systems to attacks has been extensively researched within the academic community [19]. These studies not only aim to uncover the unknown vulnerabilities present in learning models but also to evaluate and address security concerns when these models face adversarial actions. Among the plethora of such research, evasion attacks during the testing phase have emerged as a prominent area of interest.

Previous studies have shown that the amount and types of knowledge obtained by an adversary can affect the success of the attack [16,19]. Many of the previously proposed attack strategies focus on white-box attacks [15,20–22]. Gradient descent attacks were proposed to employ the discriminant function from the targeted model to probe the decision boundary [20]. Kantchelian et al. [15] proposed two algorithms to attack tree ensemble classifiers. Their experimental results showed that tree ensembles such as random forests and gradient-boosted trees are vulnerable to evasion attacks in terms of white-box attacks.

There exist works that concentrate specifically on black-box attacks. These studies mainly aim to design a robust classifier in terms of the application domains [23]. Liu et al. [24] presented ensemble-based approaches to generate transferable adversarial samples that can be used as a black-box attack against (Available online: clarifai.com). Shokri et al. [25] studied the targeted model where ‘machine learning’ was used as a service, and presented a *shadow training* approach to launch black-box attacks using realistic datasets. Their work shows that the ‘machine learning’ service model is vulnerable to membership inference attacks. Alzantot et al. [26] argued that existing black-box attacks usually require many times more queries either in obtaining the training information or in obtaining the gradients based on the output scores. Therefore, they developed a gradient-free optimization approach to create visually imperceptible adversarial samples.

2.2. Defense Against Evasion Attacks

Adversarial training has become a widely adopted approach for training robust machine learning models [27]. The core concept behind adversarial training involves training a classifier by incorporating adversarial examples into the training data set. Goodfellow et al. [28] introduced the idea of considering an augmented objective during the training procedure, which entails adding adversarial examples to the original training dataset. Huang et al. [9], on the other hand, proposed a more efficient method for generating adversarial examples, specifically referring to them as supervised samples. They highlighted that the robustness of neural networks should be learned by utilizing these supervised adversarial samples to train a substantially improved robust model.

Defensive distillation uses the additional information extracted from the distillation to return to the training regimen to improve the robustness of networks [10]. This method is especially robust against gradient-based attacks. However, studies [29,30] showed that distillation is not robust to adversarial samples. Especially, when facing white-box attacks. Thus, Meng and Chen [31] proposed MagNet, which employs several detectors and a reformer network for defending neural networks.

As illustrated by Metzen et al. in [32], they devised subnetworks that are interconnected with the main network, designed to discern whether the input to the network is a non-adversarial sample or an adversarial one. Specifically, their process involves initially training a classification network using non-adversarial samples. Following this, adversarial samples are generated for each individual data point. Subsequently, a subnetwork is further trained using both the generated adversarial samples and the original non-adversarial ones. In another work by Zhang et al. [18], they proposed an adversarial

feature selection model which enhances the security of classifiers against evasion attacks. This model improves robustness by incorporating specific assumptions about the attacker's data manipulation strategy.

On the other hand, Smutz et al. [13] demonstrated that ensemble classifiers, such as ensemble trees or SVMs, can serve as a defense strategy against evasion attacks by leveraging the diversity within the ensembles themselves. In contrast, Huang et al. [33] employed deep ensembles as a means of adversarial defense, enhancing their resistance by promoting diversity in the high-level feature representations and gradient dispersion during the simultaneous training of deep ensemble networks. Table 1 presents the defense strategies obtained from a literature review of current defense methods against evasion attacks.

Table 1. Defense strategies against evasion attacks.

Defense Technique	Description	Publication
Adversarial training	Add adversarial samples to the original training data.	[9,28]
Defensive distillation	Employ the extra information extracted from the distillation process to feed back into the training regimen for enhancing the robustness of the network.	[10]
Several detectors	Employ several detectors alongside a reformer network to defend against attacks.	[31]
Statistical tests	Use a statistical test to differentiate adversarial examples from training data.	[34]
Binary detector network	Subnetworks that branch off from the main network are trained to discern the input to the network.	[32]
Pre-trained softmax neural classifier	A framework for identifying out-of-distribution samples and adversarial attacks.	[35]
Random perturbations	Analyze the model's responses to an input subjected to random perturbations.	[36]
Adversarial feature selection	This method enhances the security of the classifier against evasion attacks by incorporating specific assumptions about the adversary's data manipulation tactics.	[18]
Ensemble trees or SVMs	To defend against evasion attacks by examining the diversity within the ensembles themselves.	[13]
Deep ensemble	Encourage diversity in the learning of high-level feature representations and gradient dispersion during the concurrent training of deep ensemble networks.	[33]

However, an adversary can still launch attacks against an ensemble of classifiers because they generalize across classifiers. For example, Zhang et al. [16] conducted research on evasion attacks against ensembles of SVMs and highlighted that, in practical scenarios, ensemble SVMs can be more susceptible to evasion attacks compared to a single SVM. Kantchelian et al. [15] showed that it is possible to employ a Mixed Integer Linear Program solver to generate an evading instance and launch attacks against ensemble of regression trees. In this paper, we focus on improving adversarial robustness of ensemble classifiers by diversified feature selection and stochastic aggregation strategy.

3. Overview of the Proposed Model

In this section, we begin with a review of the evasion attack model, followed by a description of the motivation and structure of the proposed model.

3.1. Evasion Attack Model

In order to efficaciously evaluate the robustness of learning-based classifiers against evasion attacks, we utilize the evasion attack model initially defined in the works [19,20]. In an evasion attack, an adversary's goal is to estimate the decision boundary of the targeted system and manipulate the input sample to mislead the decision of the targeted system. Without the loss of generality, the problem of an evasion attack can be described as: given a

machine learning system M , and an input sample \mathbf{x} , where \mathbf{x} can be correctly classified by M and the output is $c(\mathbf{x})$. An adversary's goal is to try to find its classification boundary by probing the classifier. Then, an adversary can modify the content of \mathbf{x} after knowing what kinds of instances can be misclassified by the classifier. Therefore, it is possible for \mathbf{x} be modified to \mathbf{x}' by minimally manipulating \mathbf{x} , where \mathbf{x}' is classified incorrectly (i.e., $c(\mathbf{x}') \neq c(\mathbf{x})$). Suppose the amount of manipulations is characterized by distance function $d(\mathbf{x}, \mathbf{x}')$, the evasion attack problem can be written as [18]

$$E(\mathbf{x}) = \arg \min_{\mathbf{x}'} d(\mathbf{x}, \mathbf{x}'), \quad s.t. \quad c(\mathbf{x}') \neq c(\mathbf{x}). \quad (1)$$

3.2. Motivation and Architecture of the Proposed Model

Conventional ensemble classifiers, such as Random Forest [37,38], Gradient-Boosting Trees [39], Ensemble SVMs [40,41] and so on, consist of multiple *weak* classifiers, which increase the diversity of classifiers and improve classification performance [42,43], while our primary objective is to enhance the robustness of classifiers. Therefore, the first motivation of our approach is to ensemble with *strong* classifiers. The *strong* classifiers here refer to the robust classifiers without significantly reducing classification accuracy. Intuitively, the ensemble of multiple *strong* classifiers guarantees both robustness and classification accuracy.

Another motivation is that the learning procedure of conventional ensemble classifiers may enable an attacker to train a classifier with a decision boundary that closely mimics the targeted system using a minimal amount of training data, thereby facilitating the attack [16]. Thus, our idea revolves around modifying the learning process to obfuscate the decision boundary, such that even if an attacker possesses knowledge of some or all of the training data, it becomes challenging for them to accurately learn the true decision boundary, thereby enhancing the robustness of the targeted system.

The last point is about the aggregation strategy, which usually adopts voting or averaging [40]. No matter which method, all the classifiers in the ensemble are utilized to make the decision for better performance. However, we intend to use *strong* classifiers as the base classifiers of the ensemble. Any single classifier in the ensemble can achieve good performance. It does not need all classifiers to participate in decision-making. We can randomly selecting m classifiers to make the decision. There are two benefits for doing this. One is that combining the decision of m classifiers can boost the accuracy performance compared with a single classifier. Second, randomly select m classifiers can confuse classification boundaries. Even if an attacker is aware of all the parameters of the targeted system, the real decision boundary remains elusive due to the random selection of classifiers that contribute to the decision-making process.

The architecture of the proposed model is depicted in Figure 1. Within the model, we initially conduct feature selection to identify a set of feature vectors that are advantageous for both robustness and accuracy performance. All selected feature vectors are stored in a feature vector bank. Subsequently, base classifiers are trained using the chosen feature vectors from this bank, and these classifiers are then placed in a classifier bank. Ultimately, the decision made by the proposed model is determined through the consensus of m randomly selected classifiers from the classifier bank.

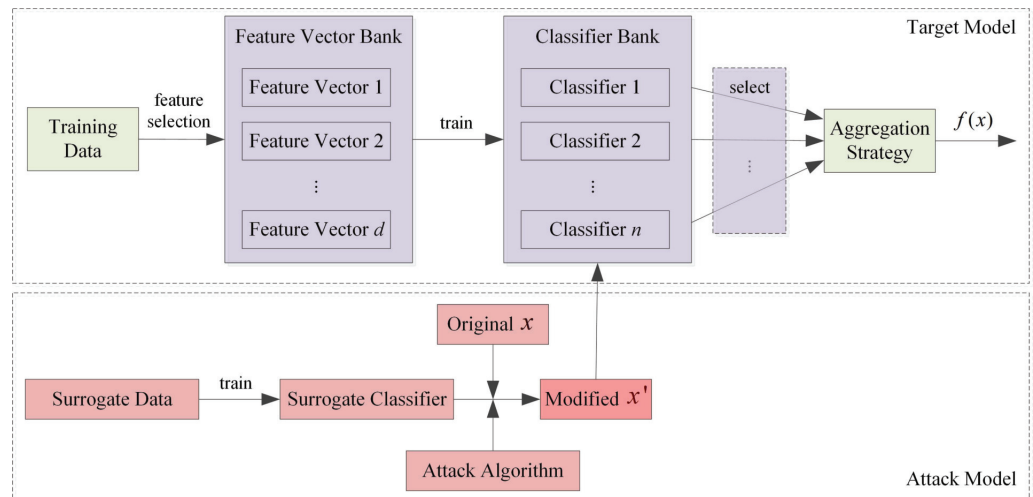


Figure 1. Architecture of the proposed model.

4. Proposed Model

In this section, we present the proposed model for improving adversarial robustness of ensemble classifiers (AREC). In evasion attacks, the more an attacker knows about the decision boundary of the targeted classifier, the easier it becomes to execute a successful evasion. Our fundamental approach is to obfuscate the decision boundary, making it unpredictable, while concurrently maximizing the generalization capacity of the classifier, thus enhancing its resilience against evasion attacks.

4.1. Training Procedure of AREC

As discussed in Section 3, each classifier used in our model should not only optimize accuracy—it is more appropriate to optimize a trade-off between accuracy and robustness. We apply the adversarial feature selection approach to optimize the trade-off [18]. Given a d -dimensional sample x , the criterion can be formalized as

$$k^* = \arg \max_{\hat{x}_k} G(\hat{x}_k) + \lambda S(\hat{x}_k) \tag{2}$$

where G symbolizes an estimation of the classifier’s generalization ability in the absence of attack. S represents the classifier security against evasion attacks, λ is a trade-off parameter, $\hat{x}_k, k = 1, 2, \dots, d$, is the mapping of x in the subspace of k selected features, and k^* is the k th optimal feature be selected.

Let $g : \mathcal{X} \rightarrow \mathcal{Y}$ be a classifier. For a given sample $x \in \mathcal{X}$ and its label $y \in \mathcal{Y}$, G can be formalized as

$$G = \mathbb{E}_{(x,y) \sim P_{\mathcal{X} \times \mathcal{Y}}} l(g(\hat{x}_k), y) \tag{3}$$

where \mathbb{E} represents the expectation operator, $P_{\mathcal{X} \times \mathcal{Y}}$ is the data distribution, and $g(\cdot)$ is the discriminant function of classifier g . For binary classifiers $\mathcal{Y} = \{0, 1\}$ and

$$l(g(\hat{x}_k), y) = \begin{cases} 1, & \text{if } yg(\hat{x}_k) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

As discussed in Section 3, the security term S can be formalized as

$$S = \mathbb{E}_{(x,y) \sim P_{\mathcal{X} | \mathcal{Y}=1}} d(\hat{x}_k, \hat{x}'_k) \tag{5}$$

where $\mathcal{Y} = 1$ represents the label of malicious samples, and \hat{x}'_k is the optimal solution to problem (1).

As the data distribution $P_{\mathcal{X} \times \mathcal{Y}}$ and $P_{\mathcal{X} | \mathcal{Y}=1}$ are typically unknown, we can estimate G and S using a set of n fixed samples, which can be written as

$$G \approx \frac{1}{n} \sum_{i=1}^n l(g(\hat{x}_k^i), y^i) \tag{6}$$

$$S \approx \frac{1}{n^+} \sum_{i=1}^{n^+} d(\hat{x}_k, \hat{x}'_k) \tag{7}$$

where n^+ denotes the number of malicious samples within the set of n samples. It should be emphasized that the value of S varies depending on the datasets and the distance function $d(\cdot, \cdot)$. Therefore, we can use parameter λ to avoid the dependency, e.g., one may rescale λ by dividing its value by the maximum value of $d(\cdot, \cdot)$. Also, λ can be used to trade-off between G and S .

The criterion (2) can be exploited to select one optimal feature at a time. After each feature selection, the selected features are put in the feature vector bank and also used for the next feature selection step. The base classifiers of AREC are trained using the selected feature vectors in the feature vector bank. The training procedure of the proposed model is shown in Figure 2. The detailed training process is given by Algorithm 1.

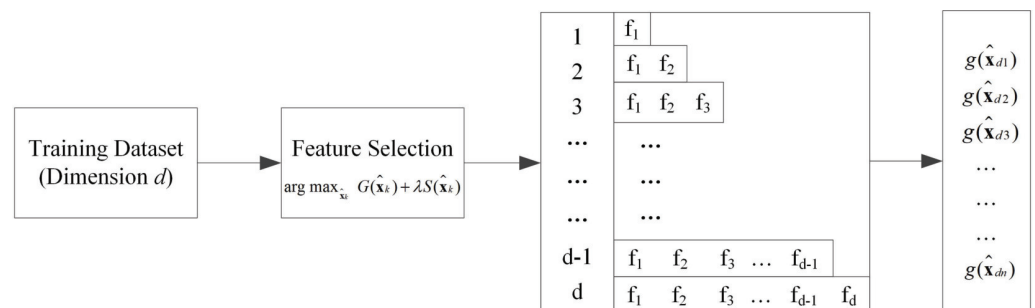


Figure 2. Training procedure of the proposed model.

Algorithm 1 Training AREC

- Input:** $\mathcal{D} = \{x^i, y^i\}_{i=1}^d$: the training dataset, λ : the trade-off parameter.
Output: $M[d, d]$: the selected feature matrix.
- 1: $M[a, b] = 0, a = 1, \dots, d$ and $b = 1, \dots, d$
 - 2: $\mathcal{S} \leftarrow \emptyset, \mathcal{U} \leftarrow \{1, \dots, d\}$
 - 3: **for** j from 1 to d **do**
 - 4: **for** each feature $k \in \mathcal{U}$ **do**
 - 5: $\mathcal{F} \leftarrow \mathcal{S} \cup \{k\}$
 - 6: $\theta = \mathbf{0}$, and then $\theta_f = 1$ for $f \in \mathcal{F}$
 - 7: Estimate $G_k(\theta)$ and $S_k(\theta)$ using cross-validation on $\mathcal{D}_{\subseteq} = \{x^i_\theta, y^i\}_{i=1}^d$
 - 8: **end for**
 - 9: $\lambda' = \lambda(\max_k S_k)^{-1}$
 - 10: $k^* = \arg \max_k G_k(\theta) + \lambda' S_k(\theta)$
 - 11: $\mathcal{S} \leftarrow \mathcal{S} \cup \{k^*\}$
 - 12: $\mathcal{F} \leftarrow \mathcal{S}$
 - 13: $\theta = \mathbf{0}$, and then $\theta_f = 1$ for $f \in \mathcal{F}$
 - 14: $M[j] = \theta$
 - 15: **end for**
 - 16: **return:** M

4.2. Aggregation Strategy of AREC

Following feature selection, each feature vector in the feature vector bank is utilized to train a base classifier and subsequently added to the base classifier bank. Ultimately, m classifiers are randomly selected from the classifier bank for decision-making purposes. This design inherently increases confusion, making it difficult for an attacker to predict which classifiers will be engaged in the decision process. Put simply, this configuration

renders evasion attacks more challenging. Moreover, it is important to note that every individual classifier within the ensemble is trained using carefully selected features, striking a balance between accuracy and robustness. Each classifier operates within a unique feature space and assigns different feature weights. Collectively, these characteristics render the proposed approach more robust against evasion attacks. The classification procedure of the proposed model is shown in Figure 3 and Algorithm 2.

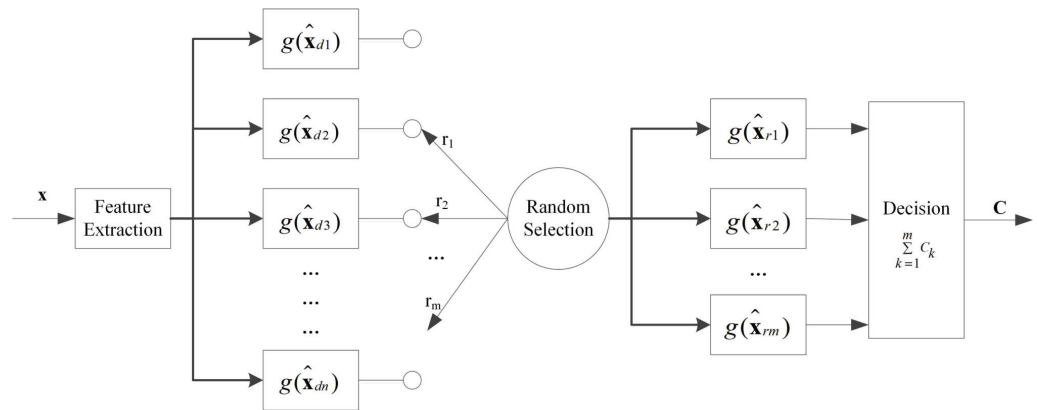


Figure 3. Classification procedure of the proposed model.

Algorithm 2 Classification Procedure of AREC

- Input:** x : the input sample.
Output: y : the label of x .
- 1: Randomly select m classifiers from $[G_{d_1}, G_{d_2}, \dots, G_{d_n}]$ to form $\mathcal{R} = \{G_{r_1}, \dots, G_{r_m}\}$, where m is an odd number and $m < n$
 - 2: $k = 1$
 - 3: **for** $G \in \mathcal{R}$ **do**
 - 4: If $G(x) > 0$ then $C_k = 1$, else if $G(x) < 0$ then $C_k = -1$
 - 5: $k = k + 1$
 - 6: **end for**
 - 7: If $\sum_{k=1}^m C_k > 0$ then $y = 1$, else $y = -1$
 - 8: **return:** y

5. Classifier Security Evaluation

Following the adversary model presented in [19,20,44], an attacker’s knowledge can be described in four levels: (1) the training data \mathcal{D} ; (2) the feature set \mathcal{X} ; (3) the learning algorithm f , and (4) the targeted model parameters \mathbf{p} . Thus, the knowledge can be characterized in terms of $\varphi = (\mathcal{D}, \mathcal{X}, f, \mathbf{p})$. According to this assumption, the knowledge of an attacker can be divided into two categories:

White-box attacks: An attacker is assumed to know all of the targeted model, namely $\varphi = (\mathcal{D}, \mathcal{X}, f, \mathbf{p})$.

Black-box attacks: In this scenario, an attacker is assumed to possess some level of knowledge about the targeted model. In this paper, we suppose that the attacker knows f and \mathcal{X} , whereas \mathcal{D} and \mathbf{p} remain unknown to the attacker. However, an attacker can estimate the parameters namely $\hat{\mathbf{p}}$ trained on a subset of \mathcal{D} or a surrogate dataset $\hat{\mathcal{D}} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{N_s}$ of N_s samples drawn from the resemble distribution of \mathcal{D} . The surrogate dataset may be collected from an alternate source. Thus, we can define this scenario as $\hat{\varphi} = (\hat{\mathcal{D}}, \mathcal{X}, f, \hat{\mathbf{p}})$.

In this paper, we examine the robustness of SVM classifiers under black-box attack scenarios (while the evaluation under white-box attacks will be given in discussion). With regard to the attacker’s knowledge about the dataset \mathcal{D} , we consider two distinct attack situations. One is *subset scenario* which assumes a subset of \mathcal{D} is able to be collected by an

attacker, i.e., $\hat{\mathcal{D}} \subset \mathcal{D}$. The other is *surrogate data scenario* which assumes a surrogate dataset $\hat{\mathcal{D}}$ drawn from the resemble distribution of \mathcal{D} can be collected by an attacker.

The gradient descent evasion attack is adopted to solve the optimization problem in Equation (1), which was shown to be effective against SVM-based classifiers [16,20]. The process of the gradient descent evasion attack is detailed in Algorithm 3.

The gradients of single SVMs and ensemble SVMs can be found in [16]. Here, we give the gradients of the proposed ensemble SVMs used in this paper followed by the updated gradient correlation measure.

Algorithm 3 Gradient Descent Evasion Attack

Input: \mathbf{x}^0 : the initial attack point, α : the gradient step size, d_{max} : the maximum number of iterations.
Output: \mathbf{x} : the final attack point.

- 1: Make $\nabla g(\mathbf{x}^0)$ and \mathbf{x}^0 a matrix $[\nabla g(\mathbf{x}^0), \mathbf{x}^0]$
- 2: $[\mathbf{v}, \mathbf{x}] \leftarrow$ Rearrange the matrix $[\nabla g(\mathbf{x}^0), \mathbf{x}^0]$ according to the descending order of $|\nabla g(\mathbf{x}^0)|$
- 3: $i \leftarrow 0$
- 4: **while** $g(\mathbf{x}) > 0$ && $i < d_{max}$ **do**
- 5: $i \leftarrow i + 1$
- 6: **if** $v_i > 0$ && $(x_i - \alpha) \in [0, 1]$ **then**
- 7: $x_i \leftarrow x_i - \alpha$
- 8: **else if** $v_i < 0$ && $(x_i + \alpha) \in [0, 1]$ **then**
- 9: $x_i \leftarrow x_i + \alpha$
- 10: **end if**
- 11: **end while**
- 12: **return:** \mathbf{x}

5.1. Gradients of the Proposed Ensemble SVMs

Because the classifiers that participate in decision-making are randomly selected from the classifier bank, it is difficult to find the exact gradient. In this section, we give three approaches to find approximate gradients. As discussed above, we assume an attacker knows \mathcal{X} and f . Here, we further assume that the attacker is aware of the number of classifiers selected in the ensemble, which is parameter m .

- Averaging gradient: This means we average each gradient of the classifier from the classifier bank, the gradient function is just like gradients of ensemble SVMs.
- Gradient of the minimal features: Since only one feature is added from one feature vector to the next, the shortest feature vector must be contained within all the other vectors, suggesting that these features could be the most significant in the ensemble. Typically, classifiers do not use very few features, such as just one or two. It is worthwhile to investigate the attack efficiency leveraging this gradient.
- Gradient of the maximal features: Given the difficulty in determining which specific features contribute significantly to the classification process, it is a prudent choice to employ all available features.

5.2. Updated Gradient Correlation

In [16], we proposed a gradient correlation measure to evaluate the similarity of gradient between the surrogate and targeted classifiers, which is given by

$$GC = \frac{\sum_{k=1}^n C(k)}{n} \tag{8}$$

where

$$C(k) = \frac{\sum_{i=1}^k v'_i}{\sum_{i=1}^k v_i} \tag{9}$$

Let \mathbf{v}^+ denote the original gradient vector of the targeted classifier, \mathbf{v} is the vector which sorted $|\mathbf{v}^+|$ in descending order, i.e., $v_1 \geq v_2 \geq \dots \geq v_n$. \mathbf{v}' is the gradient vector of surrogate classifier with the absolute gradient value of target classifier for the same features between the targeted and surrogate classifiers. n is the amount of features adopted in the targeted classifier.

There are two issues with this metric. First, in GC, gradients of all n features are computed, but not all features need to be modified to launch an attack. Therefore, in the updated gradient correlation (UGC), only gradients of the modified features are taken into account. The second issue is that the original GC only considers binary features. In the updated version of GC, we expand it to accommodate any real-valued feature. The updated gradient correlation is given by

$$UGC = \frac{\sum_{i=1}^l \alpha v'_i}{\sum_{i=1}^l \alpha v_i} = \frac{\sum_{i=1}^l v'_i}{\sum_{i=1}^l v_i} \quad (10)$$

where l represents the number of modified features to let $c(\mathbf{x}') \neq c(\mathbf{x})$. α denotes the step size and $\alpha \in (0, 1]$, when each feature value is normalized to $[0, 1]$. For binary features, $\alpha = 1$. The detailed procedure of updated gradient correlation measure is given by Algorithm 4. From Algorithm 4, we can see that $UGC \in [0, 1]$, $UGC = 1$ and $UGC = 0$ correspond to the most correlated and the most uncorrelated gradient distribution, respectively.

Algorithm 4 Updated Gradient Correlation

Input: $[\mathbf{v}^+, \mathbf{f}^+]$, \mathbf{v}^+ : the original gradient vector of the targeted classifier, \mathbf{f}^+ : the features adopted in the targeted classifier; $[\mathbf{v}^-, \mathbf{f}^-]$, \mathbf{v}^- : the original gradient vector of the surrogate classifier, \mathbf{f}^- : the features adopted in the surrogate classifier, $\mathbf{f}^- \subseteq \mathbf{f}^+$; n : the amount of features adopted in the targeted classifier; m : the amount of features adopted in the surrogate classifier; l : the amount of modified features.

Output: UGC

```

1:  $[\mathbf{v}, \mathbf{f}] \leftarrow$  sort  $|\mathbf{v}^+|$  in descending order;
2:  $[\mathbf{v}^*, \mathbf{f}^*] \leftarrow$  sort  $|\mathbf{v}^-|$  in descending order;
3:  $j \leftarrow 1$ ;
4: while  $j \leq m$  do
5:    $p \leftarrow$  find the position of  $f_j^*$  in  $\mathbf{f}$  if exist, otherwise  $p \leftarrow 0$ ;
6:   if  $p > 0$  then
7:      $v'_j \leftarrow v_p$ 
8:   else
9:      $v'_j \leftarrow 0$ 
10:  end if
11:   $j \leftarrow j + 1$ 
12: end while
13: if  $m < n$  then
14:    $v'_j \leftarrow 0, j = m, m + 1, \dots, n$ 
15: end if
16:  $UGC = \frac{\sum_{i=1}^l v'_i}{\sum_{i=1}^l v_i}$ 
17: return: UGC

```

To illustrate how the updated gradient correlation works, consider the following case with five binary features (see Figure 4). The top left of Figure 4 shows the original gradient vector \mathbf{v}^+ and the feature vector \mathbf{f}^+ from the targeted classifier. \mathbf{v} denotes the vector sorted by $|\mathbf{v}^+|$ in descending order and \mathbf{f} is the feature vector sorted with \mathbf{v} . The top right of Figure 4 shows the original gradient vector \mathbf{v}^- and the feature vector \mathbf{f}^- from the surrogate classifier. \mathbf{v}^* denotes the vector sorted by $|\mathbf{v}^-|$ in descending order and \mathbf{f}^* is the feature vector sorted with \mathbf{v}^* . \mathbf{v}' is the gradient vector of the surrogate classifier relative to the targeted classifier.

Why use \mathbf{v}' ? From Algorithm 3, one can see that, in a gradient descent attack, an attacker modifies features according to the value of their gradients. In this example, according to the sequence of modifying features obtained by an attacker, the first feature should be modified is f_3 , and the gradient change is 3 for the targeted system by modifying f_3 . According to the gradient of the targeted system, the feature with the greatest impact is f_5 . By modifying f_5 , the gradient change is 4. Therefore, in the case of modifying only one feature ($l = 1$), the gradient ratio between the surrogate system and the targeted system is $UGC = 3/4$. If two features need to be modified ($l = 2$), an attacker will modify f_3 and f_2 , the sum of the gradients corresponding to these two features is 4, while for the targeted system, modifying two features can make the gradient change 7, so $UGC = 4/7$. In the case, we assume an attacker needs to modify three features to let $c(\mathbf{x}') \neq c(\mathbf{x})$. Thus, $l = 3$ and $UGC = 6/9$. It should be noted that we use GC to represent the updated version of gradient correlation.

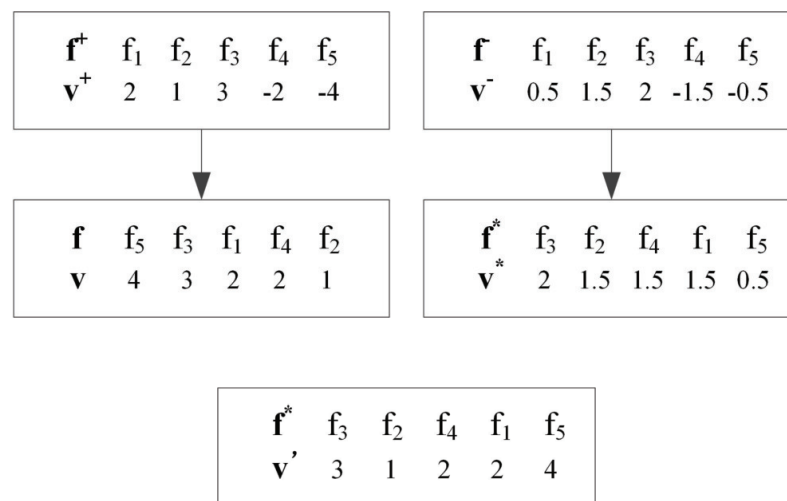


Figure 4. An example of the updated gradient correlation.

6. Experimental Evaluation

In this section, we evaluate the robustness of ARE SVMs, RSE SVMs (Random Subspace Ensemble SVMs [13,37]), a conventional SVM and a SVM trained by adversarial feature selection [18]; we call this approach AFS SVM. However, the authors did not show how to determine the optimal number of selected features. Through the analysis of the results in the paper [18], we discovered that both classification accuracy and robustness perform satisfactorily when approximately half of the total number of features are selected to train the SVM. Consequently, half the number of features were chosen to train the AFS SVM. Three tasks, namely spam email filtering, malware detection and digit recognition, are considered in the evaluation. In both the subset scenario and the surrogate data scenario, we vary an attacker’s knowledge by portions of data, 10%, 20%, ..., 100%.

For RSE SVMs, each ensemble classifier contained 100 independent base classifiers and the feature bagging ratio was set to 50%. For ARE SVMs, we applied five-fold cross-validation to train models. For each Linear-SVM-based classifier, the SVM regularization parameter was set to $C = 1$. For all RBF-SVM-based classifiers, we set the regularization parameter $C = 100$ and the kernel parameter $\gamma = 0.01$.

Both the updated gradient correlation and the hardness of evasion [18] measures were adopted for security evaluation. For a single SVM, AFS SVM and RSE SVMs, we ran each experiment 30 times and the results were averaged to produce the figures. For ARE SVMs, 30 independent models were built and we ran 30 times on each model. Thus, the results showed in the figures were averaged by 900. All experiments in the paper were implemented using MATLAB. The following shows the experimental results on three real application datasets.

6.1. Feasibility Analysis on Spam Email Filtering

We consider the PU3 dataset in the spam email filtering task [45,46]. We apply the experimental setup described in [16]. There are three subsets split by 4130 emails—one for training, one is used for the surrogate data, and the last one is used for testing.

Firstly, we give the results based on Linear-SVM and use the spam email filtering case study to show how to select parameters for ARE SVMs. Then we compare the results with single SVM, AFS SVM and RSE SVMs. The left side of Figure 5 shows the classification accuracy achieved by ARE Linear-SVMs trained using single feature vectors. We can see that the more features are selected in the vector, the the higher accuracy, when less than 40 features are selected. The accuracy is not much different when more than 40 features are selected in the vector. In order to obtain high and stable classification accuracy, the feature vectors whose feature number is more than half of the original feature space are selected to train the classifiers in the rest of the experiments. The right side of Figure 5 shows the classification accuracy achieved by ARE Linear-SVMs in which the base classifiers are trained using feature vectors with more than 100 features. It is clear that higher accuracy is achieved with a larger m . We believe $m = 3$ is a good choice for our evaluation tasks.

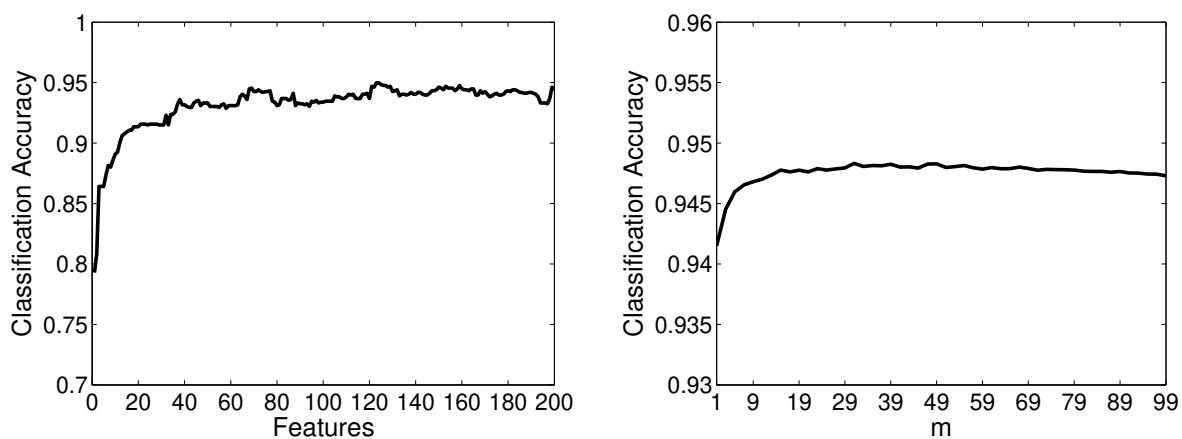


Figure 5. Classification accuracy achieved by ARE Linear-SVMs trained using single feature vector (left) and by ARE Linear-SVMs with m (right) on the PU3 dataset.

Figure 6 shows the mean ROC curves of the four methods based on Linear-SVM and RBF-SVM, respectively. From the figure, it is evident that ARE SVMs exhibit the second-highest classification performance, with RSE SVMs demonstrating the top performance among them.

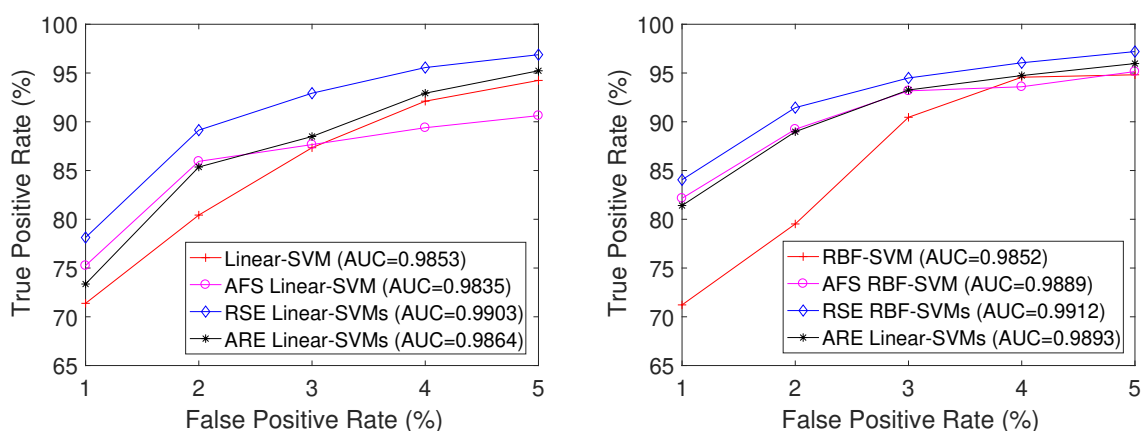


Figure 6. Mean ROC curves based on Linear-SVM (left) and RBF-SVM (right) on the PU3 dataset.

As illustrated on the left-hand side of Figure 7, ARE SVMs consistently demonstrate higher robustness compared to RSE SVMs and single SVMs, regardless of the amount of

data accessible to the attacker. Furthermore, this figure highlights that among the three types of gradients in ARE SVMs, the averaging gradient proves to be more effective than the rest. There is not a significant distinction between the gradient of minimal features and the gradient of maximal features.

On the right side of Figure 7, the gradient correlation measures for the four methodologies are displayed. It can be observed that ARE SVMs consistently exhibit lower GC scores than RSE SVMs and single SVMs, corroborating the findings in the left-hand portion of the figure. A higher GC score indicates a closer approximation of the gradient estimate between the surrogate and targeted classifiers, rendering the system more susceptible to attacks.

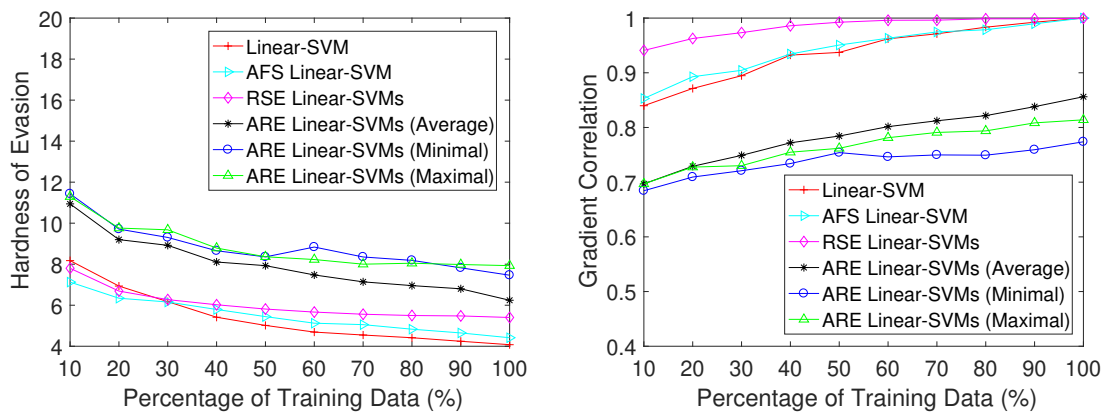


Figure 7. Hardness of evasion (i.e., the average minimum number of modified words required to classify all spam emails as legitimate) (left) and gradient correlation GC (right) based on Linear-SVM in the subset scenario.

Figure 7 also reveals that, for the ARE approach, the averaging gradient attack is more effective than the other two methods. Thus, we only give results of the averaging gradient attack in the rest of the paper.

Figure 8 shows that, under the surrogate data attack scenario, ARE SVMs are still harder to compromise than RSE SVMs and single SVMs. In this scenario, the amount of surrogate data is not as critical as that in the subset scenario and RSE SVMs are always easier to compromise by modifying fewer words on average. The gradient correlation results shown in the right side of Figure 8 also support this observation, which is ARE SVMs always have lower gradient correlation scores than RSE SVMs and single SVMs and the scores of RSE SVMs much higher than the others.

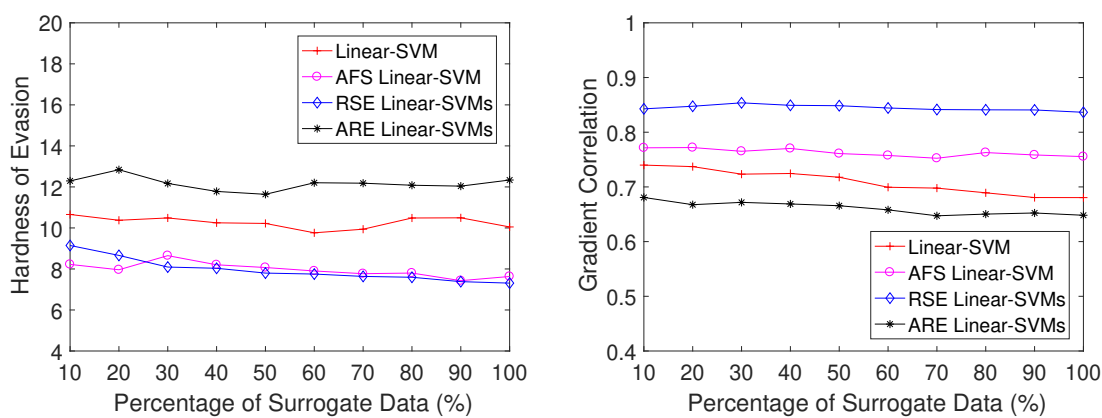


Figure 8. Hardness of evasion (left) and gradient correlation GC (right) based on Linear-SVM in the surrogate data scenario on PU3 dataset.

The results of RBF-SVM-based classifiers are shown in Figure 9. The results show the same trend as Linear-SVM-based classifiers. The ARE classifiers always have the highest

hardness of evasion scores and lowest gradient correlation scores, which indicates the ARE approach is more robust than RSE SVMs and single SVMs.

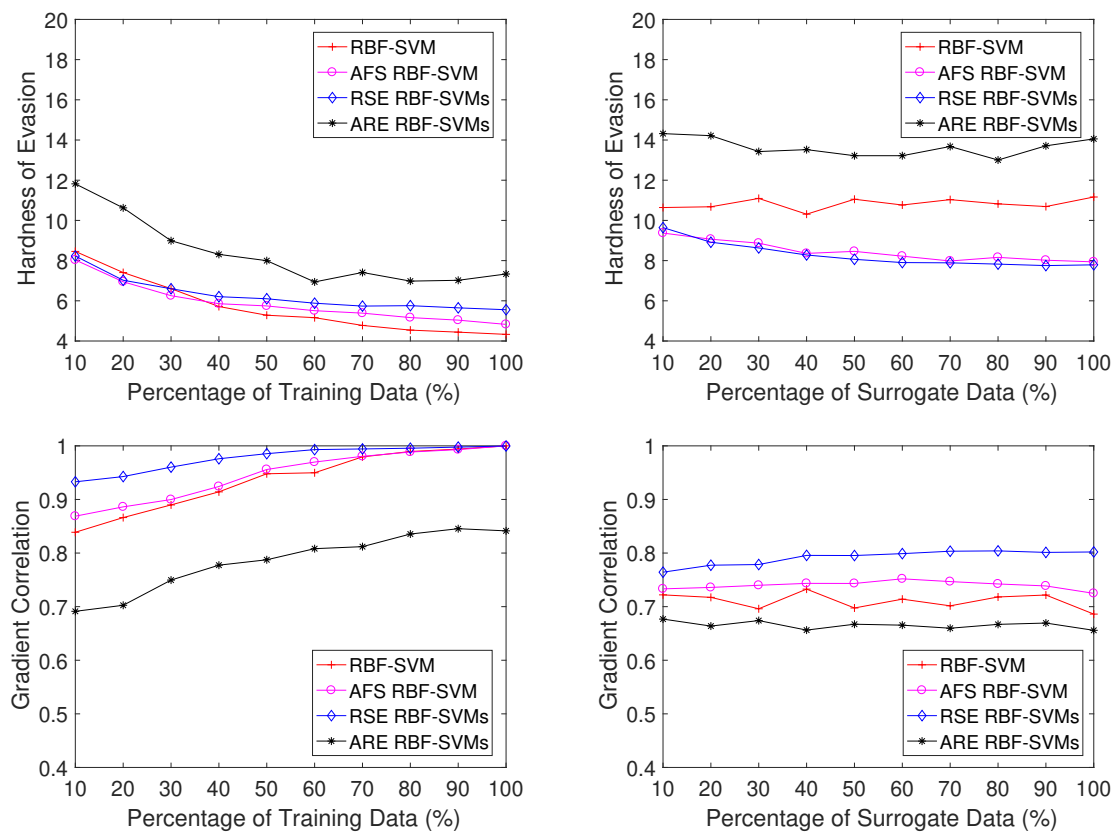


Figure 9. Hardness of evasion (top) and gradient correlation GC (bottom) based on RBF-SVM on the PU3 dataset.

6.2. Case Study on Malware Detection in PDF

Malware detection is another real-world task we considered in this paper. Also, the PDF dataset and the experimental setup described in [16] are applied in these experiments.

In this task, the gap in evasion difficulty between single SVMs and RSE SVMs is quite narrow for both Linear-based and RBF-based SVMs, and AFS SVMs significantly outperform these two methods according to Figure 10. Notably, ARE SVMs prove to be the most resistant to evasion, requiring almost twice the number of features to be manipulated for successful evasion compared to Linear-SVMs, RBF-SVMs, and RSE SVMs when the attacker possesses identical knowledge about the dataset \mathcal{D} . Figure 11 presents the gradient correlation scores, which further substantiate this observation, showing that ARE SVMs consistently display the lowest scores. This suggests that ARE SVMs exhibit enhanced robustness against gradient descent attacks.

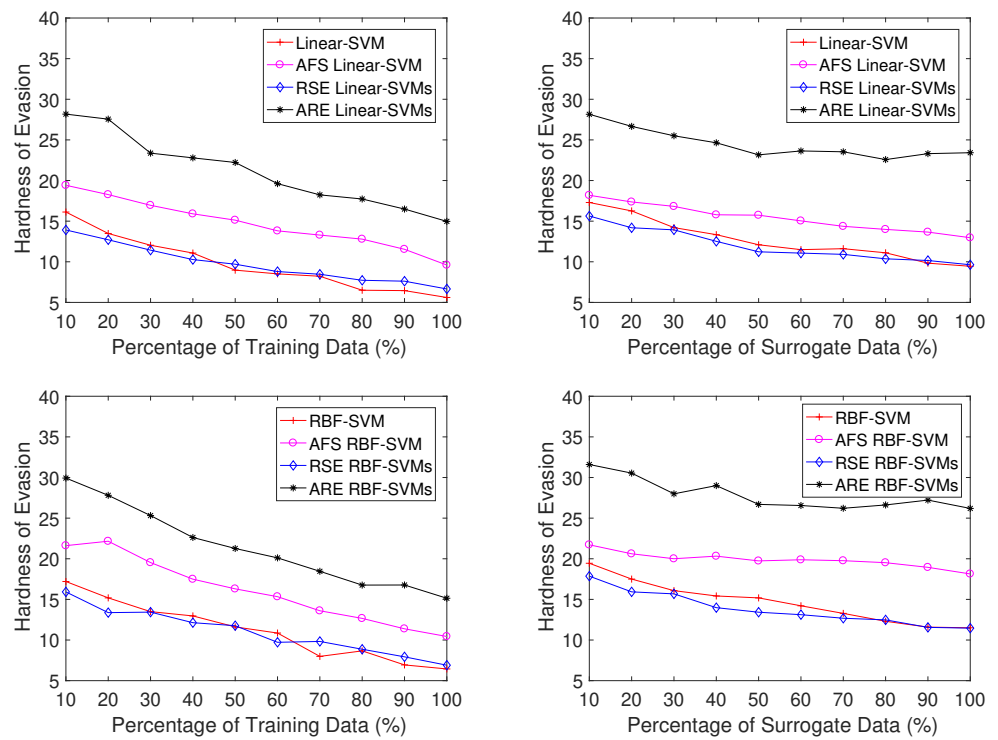


Figure 10. Hardness of evasion (i.e., average minimum number of keywords that need to be added to make each malicious PDF file be misclassified as benign) on the PDF dataset in the subset scenario (left) and the surrogate scenario (right).

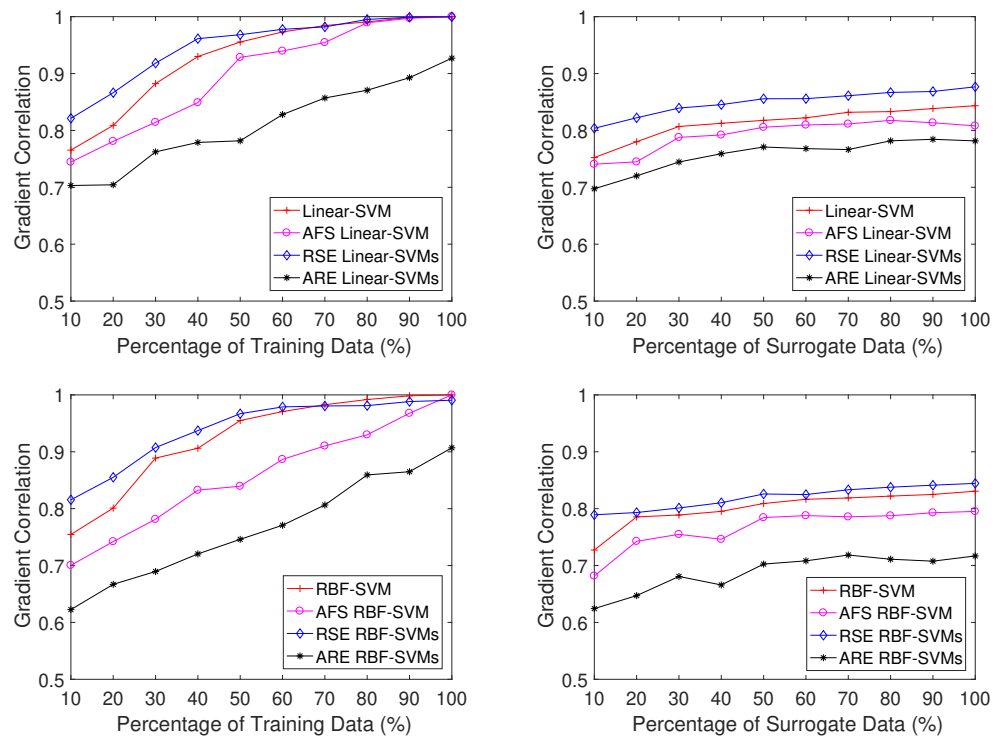


Figure 11. Gradient correlation GC on the PDF dataset in the subset scenario (left) and the surrogate data scenario (right).

6.3. Case Study on Handwritten Digit Recognition

The third task involves handwritten digit recognition, utilizing the MNIST dataset. In accordance with [14], we specifically discriminate between the digits “2” and “6”. Thus, we have 11,876 images for training and 1990 images for testing purposes. From the pool of 11,876 images, we randomly select 1000 samples and partition them into two subsets, each containing 500 images, which serve as the training data and the surrogate data, respectively. To assess robustness, we choose 100 instances of the digit “6” from the test data, ensuring that all the considered models accurately recognize these 100 instances. For evaluating the overall classification performance, all 1990 images are employed.

From Figure 12, it can be discerned that for both Linear-based and RBF-based SVMs, ARE SVMs consistently maintain the second-highest classification performance. However, in the case of Linear-based SVMs, the optimal performance is achieved by RSE Linear-SVMs, while for RBF-based SVMs, the highest degree of performance is attributed to RBF-SVM.

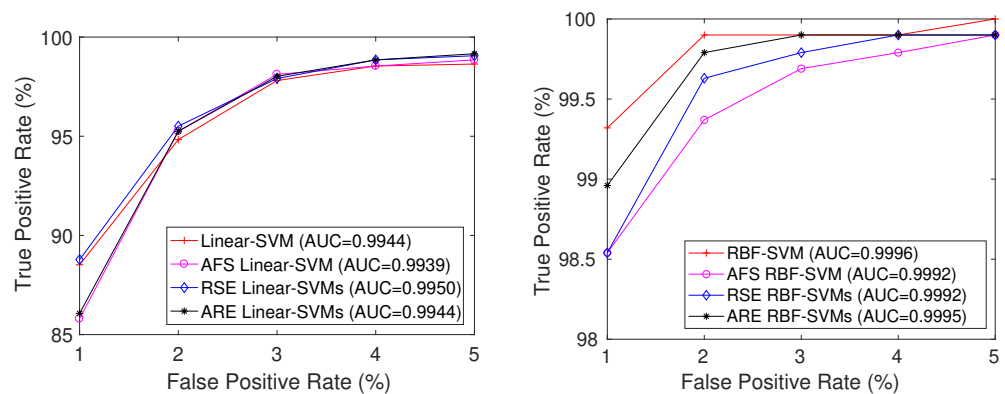


Figure 12. Mean ROC curves on MNIST dataset based on Linear-SVM (left) and RBF-SVM (right).

In the digit-recognition task, Figure 13 confirms that ARE SVMs remain the most robust classifiers. The performances of the other methods are relatively similar. The gradient correlation scores displayed in Figure 14 further validate this observation, as ARE SVMs consistently exhibit the lowest scores, while the scores of the other three approaches are closely clustered.

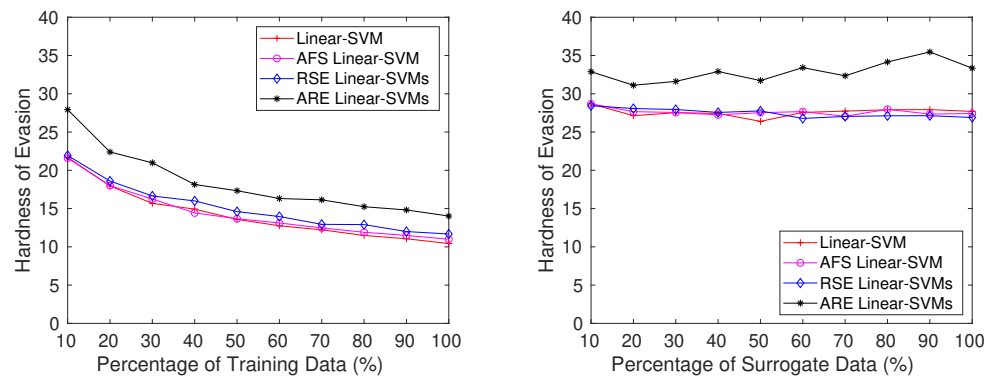


Figure 13. Cont.

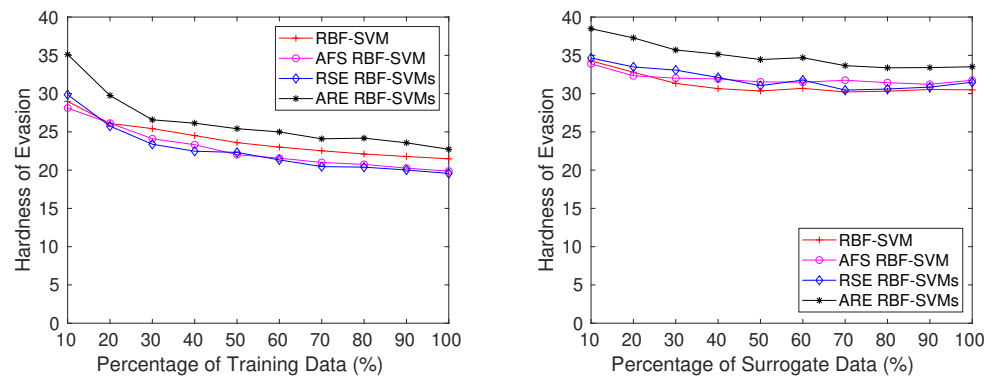


Figure 13. Hardness of evasion (i.e., average minimum number of modified pixels required to misclassify each digit “6” as “2”) on MNIST dataset in the subset scenario (left) and the surrogate data scenario (right).

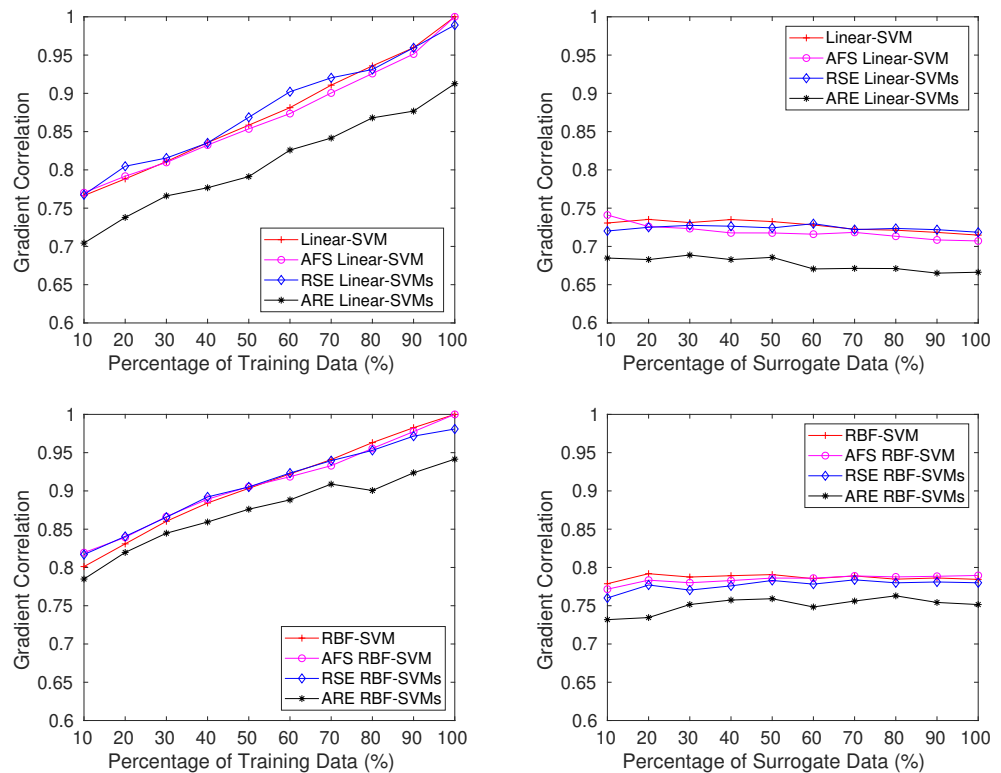


Figure 14. Gradient correlation GC on MNIST dataset in the subset scenario (left) and the surrogate data scenario (right).

7. Discussion and Limitations

7.1. Robustness under White-Box Attacks

As discussed above, even if an attacker knows all of the training data, they still cannot estimate the targeted model parameters \mathbf{p} accurately. Usually, an attacker does not have the ability to directly attack the targeted system to obtain the parameters \mathbf{p} . However, in order to evaluate the robustness of AREC in the worst-case, we show the experimental results under white-box attacks. In this case, an attacker is assumed to know the same knowledge as the targeted system, $\varphi = (\mathcal{D}, \mathcal{X}, F, \mathbf{p})$. Nevertheless, the final classifiers used for making the decision are still selected randomly. The optimal attack for an attacker is applying the average gradient. Experimental results are shown in Table 2. All results are obtained by averaging 30 independent runs. It should be noted that the experimental results of other

algorithms in Table 2 are also obtained under the assumption that an attacker knows the same knowledge as the targeted system.

Table 2. Hardness of evasion under white-box attacks.

Model	PU3	PDF	MNIST
Linear-SVM	4.08	5.61	10.44
AFS Linear-SVM	4.41	9.60	8.00
RSE Linear-SVMs	5.21	6.48	11.17
ARE Linear-SVMs	5.43	12.01	11.28
RBF-SVM	4.33	6.44	21.48
AFS RBF-SVM	4.83	10.44	14.87
RSE RBF-SVMs	5.31	6.88	18.76
ARE RBF-SVMs	5.64	13.95	20.96

For the PU3 dataset, the results obtained with ARE SVMs and RSE SVMs are comparable, both outperforming single SVMs. When it comes to the PDF dataset, ARE SVMs notably surpass RSE SVMs and single SVMs in terms of performance. Regarding the MNIST dataset, ARE SVMs continue to deliver strong performance. Based on these observations, we can conclude that even under the most adverse conditions, the proposed method exhibits greater robustness against evasion attacks compared to RSE SVMs and single SVMs.

7.2. Generalization Capability

Each base classifier in AREC is selected by evaluating $G(\theta)$ and $S(\theta)$. According to [18], these classifiers will not reduce the classification accuracy significantly compared with a single classifier. Moreover, decision-making involving m classifiers improves classification accuracy. Our experimental results show that the classification performance of ARE SVMs outperforms single SVMs in most cases. It is noteworthy that almost every malicious PDF is correctly recognized by all eight models in the PDF dataset, hence we do not include these results in Table 3.

Table 3. Classification accuracy.

Model	PU3	MNIST
Linear-SVM	0.9440	0.9729
AFS Linear-SVM	0.9395	0.9698
RSE Linear-SVMs	0.9565	0.9746
ARE Linear-SVMs	0.9447	0.9733
RBF-SVM	0.9448	0.9910
AFS RBF-SVM	0.9410	0.9763
RSE RBF-SVMs	0.9599	0.9874
ARE RBF-SVMs	0.9499	0.9800

7.3. Limitations

While the proposed approach yields encouraging outcomes, there are several limitations to consider. Firstly, the wrapper-based adversarial feature selection technique introduced in [18] is incorporated during the training process. This feature selection method effectively balances accuracy and robustness. However, it entails increased computational complexity relative to single classifiers and RSE classifiers, particularly when dealing with a large initial number of features. Although classifiers with very few features are generally not preferred, employing feature selection via backward elimination can potentially reduce training time. Nonetheless, this does not address the issue at its core. Future work calls for the development of more efficient strategies to overcome this challenge.

Secondly, due to the hierarchical nature of feature selection in the proposed model, each classifier in the ensemble adds only one additional feature based on the previous

classifier. In theory, it is possible that an attacker might infer some critical features from the feedback provided by the targeted system. However, practically speaking, this is quite challenging. The reason being that the outcome returned by the targeted system each time is derived from a unique combination of classifiers, and the system only returns the label of a sample, which does not reveal the value of the discriminant function $g(x)$. Hence, it becomes difficult to estimate the features of classifiers from the received results. An extreme scenario that cannot be entirely dismissed is if an attacker gains access to the parameters \mathbf{p} of the targeted system through social engineering tactics or other means.

8. Conclusions

Machine learning technology was initially designed with the aim of enhancing generalization capabilities. Under this objective, machine learning has thrived and been widely deployed across numerous domains such as image recognition, intrusion detection, among others. However, conventional learning-based algorithms are inherently susceptible to adversarial attacks because their original design did not account for the presence of intelligent adversaries capable of manipulating their behavior to deceive classification algorithms.

In this paper, we propose a method to improve the adversarial robustness of ensemble classifiers through diversified feature selection and stochastic aggregation. Unlike conventional ensemble classifiers that aggregate multiple *weak* classifiers, our ensemble is composed of multiple *strong* classifiers. The *strong* classifiers within AREC are trained by optimizing both their generalization ability and robustness against evasion attacks. For the ensemble integration strategy, the generalization capacity is further bolstered by employing multiple classifier voting. The application of randomly selecting decision classifiers serves to obfuscate the decision boundary of AREC. Additionally, we have updated the gradient correlation measure to ensure it is applicable to any real-number feature. Experimental results across various tasks such as spam email filtering, PDF malware detection, and handwritten digit recognition demonstrate that our proposed approach offers superior robustness compared to conventional single and ensemble classifiers. Furthermore, in contrast to the state-of-the-art algorithm AFS [18], which trades off generalization capability for security, our method significantly boosts robustness while marginally improving generalization capability.

Our future works involve designing an efficient adversarial feature selection algorithm to mitigate the training costs associated with AREC. Additionally, we aim to extend the proposed gradient correlation metric to explore the security performance of diverse learning-based classifiers beyond the current scope.

Author Contributions: Conceptualization, F.Z. and K.L.; methodology, F.Z. and K.L.; software, F.Z. and K.L.; validation, F.Z., K.L. and Z.R.; formal analysis, F.Z., K.L. and Z.R.; investigation, F.Z. and K.L.; resources, F.Z. and K.L.; data curation, F.Z. and K.L.; writing—original draft preparation, F.Z.; writing—review and editing, F.Z., K.L. and Z.R.; visualization, F.Z., K.L. and Z.R.; supervision, F.Z. and K.L.; project administration, F.Z., K.L. and Z.R.; funding acquisition, F.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Dongguan Science and Technology of Social Development Program grant number 20221800905182 and 20231800940522.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Zhang, W.; Li, X. Federated Transfer Learning for Intelligent Fault Diagnostics Using Deep Adversarial Networks With Data Privacy. *IEEE ASME Trans. Mechatronics* **2022**, *27*, 430–439. [[CrossRef](#)]

2. Wang, Z.; Cui, J.; Cai, W.; Li, Y. Partial Transfer Learning of Multidiscriminator Deep Weighted Adversarial Network in Cross-Machine Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5010010. [[CrossRef](#)]
3. Shi, Y.; Han, Y.; Hu, Q.; Yang, Y.; Tian, Q. Query-Efficient Black-Box Adversarial Attack With Customized Iteration and Sampling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 2226–2245. [[CrossRef](#)] [[PubMed](#)]
4. Kravchik, M.; Shabtai, A. Efficient Cyber Attack Detection in Industrial Control Systems Using Lightweight Neural Networks and PCA. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 2179–2197. [[CrossRef](#)]
5. Chen, L.; Ye, Y.; Bourlai, T. Adversarial machine learning in malware detection: Arms race between evasion attack and defense. In Proceedings of the 2017 European Intelligence and Security Informatics Conference (EISIC), Athens, Greece, 11–13 September 2017; pp. 99–106.
6. Liang, B.; Su, M.; You, W.; Shi, W.; Yang, G. Cracking classifiers for evasion: A case study on the google’s phishing pages filter. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 345–356.
7. Laskov, P.; Srndic, N. Practical evasion of a learning-based classifier: A case study. In Proceedings of the Security and Privacy (SP), San Jose, CA, USA, 18–21 May 2014; pp. 197–211.
8. Bhagoji, A.N.; Cullina, D.; Mittal, P. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv* **2017**, arXiv:1704.02654.
9. Huang, R.; Xu, B.; Schuurmans, D.; Szepesvári, C. Learning with a Strong Adversary. *arXiv* **2015**, arXiv:1511.03034.
10. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 23–25 May 2016; pp. 582–597.
11. Strauss, T.; Hanselmann, M.; Junginger, A.; Ulmer, H. Ensemble methods as a defense to adversarial perturbations against deep neural networks. *arXiv* **2017**, arXiv:1709.03423.
12. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
13. Smutz, C.; Stavrou, A. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In Proceedings of the Network and Distributed System Security (NDSS), San Jose, CA, USA, 21–24 February 2016.
14. Zhang, F.; Wang, Y.; Liu, S.; Wang, H. Decision-based evasion attacks on tree ensemble classifiers. *World Wide Web Internet Web Inf. Syst.* **2020**, *23*, 2957–2977. [[CrossRef](#)]
15. Kantchelian, A.; Tygar, J.; Joseph, A. Evasion and hardening of tree ensemble classifiers. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2387–2396.
16. Zhang, F.; Wang, Y.; Wang, H. Gradient Correlation: Are Ensemble Classifiers More Robust Against Evasion Attacks in Practical Settings? In *WISE 2018, Proceedings of the International Conference on Web Information Systems Engineering, Dubai, United Arab Emirates, 12–15 November 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 96–110.
17. Pang, T.; Xu, K.; Du, C.; Chen, N.; Zhu, J. Improving Adversarial Robustness via Promoting Ensemble Diversity. *arXiv* **2019**, arXiv:1901.08846.
18. Zhang, F.; Chan, P.P.; Biggio, B.; Yeung, D.S.; Roli, F. Adversarial feature selection against evasion attacks. *IEEE Trans. Cybern.* **2016**, *46*, 766–777. [[CrossRef](#)] [[PubMed](#)]
19. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognit.* **2018**, *84*, 317–331. [[CrossRef](#)]
20. Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrđić, N.; Laskov, P.; Giacinto, G.; Roli, F. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Prague, Czech Republic, 23–27 September 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 387–402.
21. Biggio, B.; Corona, I.; Nelson, B.; Rubinstein, B.I.; Maiorca, D.; Fumera, G.; Giacinto, G.; Roli, F. Security evaluation of support vector machines in adversarial environments. In *Support Vector Machines Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 105–153.
22. Xu, L.; Zhan, Z.; Xu, S.; Ye, K. An evasion and counter-evasion study in malicious websites detection. *arXiv* **2014**, arXiv:1408.1993.
23. Alzaqebah, A.; Aljarah, I.; Al-Kadi, O. A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization. *Comput. Secur.* **2023**, *124*, 102957. [[CrossRef](#)]
24. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. *arXiv* **2016**, arXiv:1611.02770.
25. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 3–18.
26. Alzantot, M.; Sharma, Y.; Chakraborty, S.; Srivastava, M. GenAttack: Practical Black-box Attacks with Gradient-Free Optimization. *arXiv* **2018**, arXiv:1805.11090.
27. Zhang, N.; Zhang, Y.; Song, S.; Chen, C.L.P. A Review of Robust Machine Scheduling. *IEEE Trans. Autom. Sci. Eng.* **2023**. [[CrossRef](#)]
28. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
29. Carlini, N.; Wagner, D. Defensive distillation is not robust to adversarial examples. *arXiv* **2016**, arXiv:1607.04311.

30. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Diego, CA, USA, 22–24 May 2017; pp. 39–57.
31. Meng, D.; Chen, H. Magnet: A two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; ACM: New York, NY, USA, 2017; pp. 135–147.
32. Metzen, J.H.; Genewein, T.; Fischer, V.; Bischoff, B. On detecting adversarial perturbations. *arXiv* **2017**, arXiv:1702.04267.
33. Huang, B.; Kei, Z.; Wang, Y.; Wang, W.; Shen, L.; Liu, F. Adversarial Defence by Diversified Simultaneous Training of Deep Ensembles. In Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, Thirty-Third Conference on Innovative Applications of Artificial Intelligence and the Eleventh Symposium on Educational Advances in Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 7823–7831.
34. Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; McDaniel, P. On the (statistical) detection of adversarial examples. *arXiv* **2017**, arXiv:1702.06280.
35. Lee, K.; Lee, K.; Lee, H.; Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 2–8 December 2018; pp. 7167–7177.
36. Huang, B.; Wang, Y.; Wang, W. Model-Agnostic Adversarial Detection by Random Perturbations. In Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4689–4696.
37. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
38. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
39. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 785–794.
40. Kim, H.C.; Pang, S.; Je, H.M.; Kim, D.; Bang, S.Y. Constructing support vector machine ensemble. *Pattern Recognit.* **2003**, *36*, 2757–2767. [[CrossRef](#)]
41. Dong, Y.S.; Han, K.S. Boosting SVM Classifiers by Ensemble. In Proceedings of the 14th International Conference on World Wide Web (WWW '05), Chiba, Japan, 10–14 May 2005; pp. 1072–1073.
42. Katakis, I.; Tsoumakas, G.; Vlahavas, I. Tracking recurring contexts using ensemble classifiers: An application to email filtering. *Knowl. Inf. Syst.* **2010**, *22*, 371–391. [[CrossRef](#)]
43. Vapnik, V. *The Nature of Statistical Learning*, 1st ed.; Springer: New York, NY, USA, 1999.
44. Demontis, A.; Melis, M.; Biggio, B.; Maiorca, D.; Arp, D.; Rieck, K.; Corona, I.; Giacinto, G.; Roli, F. Yes, machine learning can be more secure! a case study on Android malware detection. *IEEE Trans. Dependable Secur. Comput.* **2017**, *16*, 711–724. [[CrossRef](#)]
45. Mujtaba, G.; Shuib, L.; Raj, R.G.; Majeed, N.; Al-Garadi, M.A. Email classification research trends: Review and open issues. *IEEE Access* **2017**, *5*, 9044–9064. [[CrossRef](#)]
46. Androustopoulos, I.; Paliouras, G.; Michelakis, E. *Learning to Filter Unsolicited Commercial E-Mail*; Technical Report No. 2004/2; National Center for Scientific Research “Demokritos”: Athens, Greek, 2004.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.