





Kernel Optimization in SVM for Defense Against Adversarial Attacks

Wanman Li^(✉)  and Xiaozhang Liu 

Hainan University, Haikou 570228, Hainan, China

Abstract. While malicious samples were widely found in many application fields of machine learning, suitable countermeasures have been researched in the research field of adversarial machine learning. Support vector machines (SVMs), as a kind of successful approach, were widely used to solve security problems, such as image classification, malware detection, spam filtering, and intrusion detection. However, many adversarial attack methods have emerged recently, considering deep neural networks as machine learning models. Therefore, we consider applying them to SVMs and put forward an effective defense strategy against the attacks. In this paper, we aim to develop secure kernel machines against a prevalent attack method that was previously proposed in deep neural networks. This defense approach is based on the kernel optimization of SVMs with radial basis function kernels. To test this hypothesis, we evaluate our approach on MNIST and CIFAR-10 image classification datasets, and the experimental results show that our method is beneficial and makes our classifier more robust.

Keywords: Support vector machines · Kernel optimization · Adversarial machine learning

1 Introduction

During the past several decades, we have seen advances in machine learning. However, with the expansion of machine learning applications, many new challenges have also emerged. In particular, adversarial machine learning, as a machine learning technique, mainly learns the potential vulnerabilities of machine learning in adversarial scenarios and have attracted a lot of attention [1–3]. Adversarial samples have been widely found in the application fields of machine learning, notably image classification, speech recognition, and malware detection [4–6]. Meanwhile, various defensive techniques for the adversarial samples have been proposed recently, including adversarial training, defensive Distillation, pixel deflection, and local flatness regularization [7–10].

As a popular machine learning method, support vector machines (SVMs) were widely used to solve security problems, such as image classification, malware detection, spam filtering, and intrusion detection [11–13]. As described in [14], adversarial attacks against machine learning can be categorized as poisoning attacks and evasion attacks in general. A poisoning attack happens at test time, where the adversary injects a small number of specifically modified samples into the training data, which makes a

change in the boundary of the model and results in misclassification. With the rise of various poisoning attack measures against SVMs [15–19], the countermeasures for protecting SVM classifier from poisoning attacks have been developed, one is data cleaning technology [20], and the other is to improve the robustness of learning algorithms against malicious training data [21].

In this paper, we focus mainly on evasion attacks on the SVM classifier. An evasion attack is an attack that evades the trained model by constructing a well-crafted input sample during the test phase. In 2013, Biggio et al. [22] simulated various evasion attack scenarios with different risk levels to enable classifier designers to select models more wisely. However, as time went by, more and more evasion attack methods began to emerge. There are two main directions of evasion attacks to generate adversarial examples. One attack is based on the gradient, which is the most common and most successful attack method. The core idea is to use the input image as the starting and modify the image in the direction of the gradient of the loss function, such as the Fast gradient Sign Method [23], Basic Iterative Method [24], and Iterative gradient Sign Method [25]. Another is to generate adversarial samples based on hyperplane classification, such as the DeepFool algorithm [26]. Although the above methods of generating adversarial examples all consider deep neural networks as machine learning models, in this work, we focus on SVMs. Therefore, we first attempted to apply the above methods of generating adversarial samples to the SVM classifier and proposed corresponding defense strategies.

In this work, our main contribution is to propose an effective defense strategy based on kernel optimization in SVM to protect the classifier against an attack method similar to the method proposed in [26]. The experimental results (in Sect. 4) show that our approach has a very significant defensive effect on the iterative attack based on gradient. Moreover, after using kernel optimization for defense, our classifier becomes more robust. Besides, to our best knowledge, this is the first attempt to apply this adversarial attack, which is proposed in [26] to the SVM model, to generate adversarial examples and achieved good experimental results.

The remaining of this paper is arranged as follows: In Sect. 2, we introduce the relevant knowledge of SVM and the attack approach that we use throughout our work. In Sect. 3, we illustrate our defend method based on kernel optimization in SVM against adversarial examples. Experimental results are presented in Sect. 4, followed by discussion and conclusions in Sect. 5.

2 Preliminary

To better illustrate the proposed procedures, we briefly review the main concepts of the model and the adversarial attack used throughout this paper. We first introduce our notation and summarize the model we utilized in the SVM in Sect. 2.1. Then we describe the major method which was used to generate adversarial samples in Sect. 2.2.

2.1 Support Vector Machine

The SVM model is a prevailing approach of classification between two sets. For illustration, we first describe the main idea of binary SVM, which is to find a hyperplane that well-separated the two classes. In SVM, a hyperplane is a solution that can correctly divide positive and negative class samples based on the principle of structural risk minimization. Thus, the hyperplane equation is univocally represented as $\mathbf{w}^T \cdot \mathbf{x} + b = 0$, where normal vector \mathbf{w} gives its orientation, and b is its intercept displacement.

Assuming that the problem is one of binary classification, we symbol a training dataset as $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Here $\mathbf{x}_i \in \mathbb{R}^d$ is the input feature vector, $y \in \{-1, +1\}$ the output label, respectively, where N is the number of samples, and d is the dimensionality of the input space. The solution of the optimal hyperplane of the SVM model can be expressed as a convex quadratic programming problem with inequality constraints. The Lagrangian multiplier method can be used to obtain its dual problem and then α can be solved by the SMO algorithm. Finally, we can get the discriminant function.

In addition, \mathbf{w} can be calculated as $\sum_{i=1}^N \alpha_i y_i \mathbf{x}_i$, and the intercept b can be computed as

$$b = \frac{1}{|S|} \sum_{i \in S} (y_i - \sum_{j \in S} \alpha_j y_j (\mathbf{x}_i, \mathbf{x}_j)).$$

Although SVM was initially designed to solve linear classification problems, SVM was extended to nonlinear classification cases by choosing from among different kernel functions [27]. Through the kernel matrix, the training data can be projected to more complex feature space. The process of solving SVM is to solve the following quadratic optimization problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^N \alpha_i, \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0, \text{ s.t. } \sum_{i=1}^N \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, N, \end{aligned} \quad (1)$$

in which α_i is the Lagrange multiplier corresponding to the training data \mathbf{x}_i , $\mathbf{K}(\cdot)$ is the kernel function. If we define a mapping function $\Phi : X \rightarrow \chi$, that is to say, the function maps the training sets into a higher-dimensional feature space, then $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)$ can be generalized to $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$, so \mathbf{w} , and b can be written as

$$\mathbf{w} = \sum_{i=1}^N \alpha_i y_i \Phi(\mathbf{x}_i), \quad (2)$$

$$b = \frac{1}{|S|} \sum_{i \in S} (y_i - \sum_{j \in S} \alpha_j y_j \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j)), \quad (3)$$

where $S = \{i | \alpha_i > 0, i = 1, 2, \dots, m\}$ the subscript set of all the support vectors. Though it may be too complicated to compute in the feature space, one need not explicitly know, and it only corresponds to the kernel function.

2.2 Attack Strategy

In [26], they proposed the DeepFool algorithm, which is simple as well as an accurate method and based on hyperplane classification to generate adversarial samples. The primary attack method used in our study is similar to this method. In the case where the classifier f is linear, from [26], we know that the minimal perturbation to change the classifier's decision is equal to the distance from the point to the hyperplane classification times the negative gradient of the unit vector of \mathbf{w} , where \mathbf{w} is the weight vector of the hyperplane classification. For the nonlinear case, we consider the iterative procedure to find the minimum perturbation vector, as shown in Fig. 1. In some situations, we may not be able to reach the classification hyperplane in one step, like in the case of linearity, and multi-step superposition may be required. Consequently, in a high dimensional space, the minimum perturbation vector of the adversarial sample can be expressed as

$$\varepsilon_\Phi = -\frac{\mathbf{w}_\Phi^T \Phi(\mathbf{x}) + b}{\|\mathbf{w}_\Phi\|_2} \mathbf{w}_\Phi, \quad (4)$$

where \mathbf{w} and b is represented in Eq. (2) and Eq. (3).

In fact, \mathbf{w}_Φ can also be formally represented by all the support vectors in high dimension space, such as

$$\mathbf{w}_\Phi = \sum_{i \in S} \alpha_i y_i \Phi(\mathbf{x}_i). \quad (5)$$

Of course, $\Phi(\mathbf{x}_i)$ showing no explicit expression, so Eq. (5) is only part of the \mathbf{w}_Φ formalized representation, cannot be obtained.

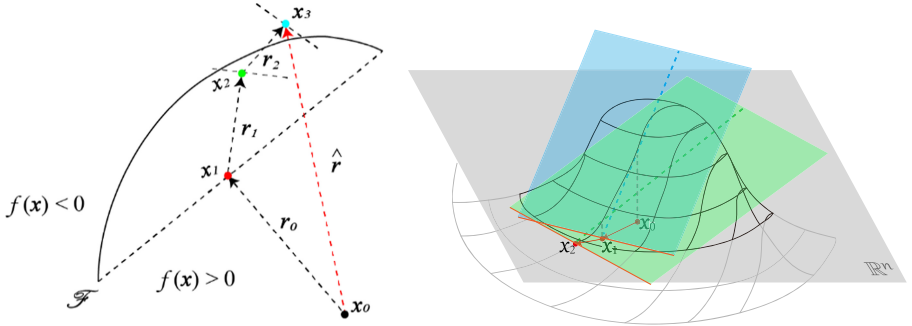


Fig. 1. The minimum perturbation that to classify the positive sample to the negative sample for a nonlinear binary classifier. On the left is the plane figure, on the right is a geometric illustration of the method.

Next, we proposed the adversarial generation method, which is based on kernel. For the nonlinear function $f(\mathbf{x})$, combined with Eq. (3) and Eq. (5), is then defined as follows

$$f(\mathbf{x}) = \mathbf{w}_\Phi^T \Phi(\mathbf{x}) + b$$

$$= \sum_{i \in S} \alpha_i y_i \mathbf{K}(\mathbf{x}_i, \mathbf{x}) + \frac{1}{\|S\|} \sum_{i \in S} (y_i - \sum_{j \in S} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x})). \quad (6)$$

For an unclassified testing sample, if the value of $f(\mathbf{x})$ is positive, the sample would be classified as a normal example. Otherwise, it would be classified as a malicious sample. The gradient of $f(\mathbf{x})$ with respect to \mathbf{x} is thus given by

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i \nabla_{\mathbf{x}} \mathbf{K}(\mathbf{x}_i, \mathbf{x}). \quad (7)$$

Here, if we use the Radial Basis Function (RBF) as the kernel function, for this kernel $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}}$, the gradient is $\nabla_{\mathbf{x}} \mathbf{K}(\mathbf{x}_i, \mathbf{x}) = -\frac{2}{\sigma^2} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}\|}{\sigma^2}} (\mathbf{x} - \mathbf{x}_i)$. Therefore, the gradient of $f(\mathbf{x})$ can be rewritten as

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = -\frac{2}{\sigma^2} \sum_{i \in S} \alpha_i y_i e^{-\frac{\|\mathbf{x}_i - \mathbf{x}\|}{\sigma^2}} (\mathbf{x} - \mathbf{x}_i). \quad (8)$$

According to Algorithm 1, we can thus find the adversarial sample.

Algorithm 1 Attack algorithm for RBF-SVM

1: input: Benign example \mathbf{x} , classifier f , kernel parameter σ .

2: output: adversarial example \mathbf{x}_{i+1}

3: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}$

4: $i \leftarrow 0$ /* Iteration count */

5: **repeat**

6: $\mathbf{r}^{(i)} \leftarrow -\frac{f(\mathbf{x}^{(i)})}{\|\nabla f(\mathbf{x}^{(i)})\|} \nabla f(\mathbf{x}^{(i)})$

7: $\mathbf{x}^{(i+1)} \leftarrow \mathbf{x}^{(i)} + \mathbf{r}^{(i)}$

8: $i \leftarrow i + 1$

9: **until** $\text{sign}(f(\mathbf{x}^{(i+1)})) \neq \text{sign}(f(\mathbf{x}^{(0)}))$

10: **return** $\mathbf{x}^{(i+1)}$

3 The Defense Based on Kernel Optimization

If we choose RBF as the kernel function, according to Eq. (1), the dual problem of SVM can be described as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}} - \sum_{i=1}^N \alpha_i \\ \text{s.t.} \quad & \sum_{i=1}^N \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, N. \end{aligned} \quad (9)$$

After solving Eq. (9) to obtain the value of α , considering optimize the kernel parameter to improve the ability of defense against adversarial attack. We noted the support vector as \mathbf{x}_s , then the discriminant function of support vectors is $f(\mathbf{x}_s) = \mathbf{w}_{\Phi}^T \Phi(\mathbf{x}_s) + b = \pm 1$. Combining with Eq. (4), correspondingly, we get the minimum perturbation radius of the support vector against the adversarial samples, which is as below

$$\varepsilon = \frac{1}{\|\mathbf{w}_{\Phi}\|_2}. \quad (10)$$

To make our model more difficult to be attacked, we urgently maximize the minimum perturbation semidiameter. Therefore, the task of defense is to maximize the value of Eq. (10), which can be achieved by minimizing $\|\mathbf{w}_{\Phi}\|_2^2$. When given the value of α , combined with Eq. (5), the optimization of the kernel parameters to defend the attacks as follows

$$\min_{\alpha} A(\sigma) = \sum_{i \in S} \sum_{j \in S} \alpha_i \alpha_j y_i y_j e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}}. \quad (11)$$

This is an unconstrained optimization problem, which can be solved by the gradient descent method

$$\sigma_k = \sigma_{k-1} - \eta A'(\sigma_{k-1}), \quad (12)$$

where $A'(\sigma) = \frac{2}{\sigma^3} \sum_{i \in S} \sum_{j \in S} \alpha_i \alpha_j y_i y_j \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2}}$. The Gaussian kernel parameter optimization algorithm for defending against adversarial attack, as shown in Algorithm 2. The initial value of the kernel parameter can be defined as $\sigma^{(0)} = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^N \sum_{j=1}^N \|\mathbf{x}_i - \mathbf{x}_j\|_2^2}$, where N is the number of training samples.

In [15], they proposed a simple yet accurate method for computing and comparing the robustness of different classifiers to adversarial perturbations; they defined the average robustness $\hat{\rho}_{adv}(f)$ as follows

$$\hat{\rho}_{adv}(f) = \frac{1}{D} \sum_{\mathbf{x} \in D} \frac{\|\mathbf{0}(\mathbf{x})\|_2}{\|\mathbf{x}\|_2}. \quad (13)$$

To verify the effectiveness of our defense method, we also use this method to compare the robustness of the classifier under different kernel parameters.

Algorithm 2 Kernel parameter optimization for defense against adversarial attacks

Set initial value $\sigma^{(0)}$. Set iteration number $m = 1$.

repeat

Solve the SVM optimization problem in Eq.(9) to obtain $\alpha^{(m)} = (\alpha_1^{(m)}, \dots, \alpha_N^{(m)})$.

Set $\alpha_j = \alpha_j^{(m)}$, $j = 1, \dots, N$, and solve the unconstrained optimization

the problem in Eq. (11) using Gradient Descent in Eq. (12) to obtain $\sigma^{(m)}$.

until No significant update for σ .

4 Experimental Results

Datasets. For the sake of demonstrating the effectiveness of the kernel optimization defense method, we validated it on MNIST [28] and CIFAR-10[29] image classification datasets, respectively. In these experiments, we only consider a standard SVM with the RBF kernel and choose data from two classes, considering one class as the benign class and a different one as the attack class. The class and number of samples employed in each training and test set are given in Table 1. In order to limit the range of the adversarial example, each pixel of the example in both datasets is normalized to $\mathbf{x} \in [0, 1]^d [0, 1]^d$ by dividing by 255, in which d represents the number of feature vectors. For the MNIST dataset, each digital image represents a grayscale image of $28 * 28$ pixels, which means that feature vectors have $d = 28 * 28 = 784$ values, while for the CIFAR-10 dataset, each image is a color image with three channels and each channel have $32 * 32$ pixels, which means that feature vectors have $d = 32 * 32 * 3 = 3072$ features. In these experiments, only the kernel parameter σ is considered, and the regularization parameter c of the SVM is fixed to default.

Table 1. Datasets used for training and testing with RBF-SVMs

Dataset	Train size	Test size	Positive	Negative
MNIST	8000	2000	Digit ‘1’	Digit ‘7’
CIFAR-10	10000	2000	Cat	Dog

After the process of training, α can be obtained, and we began to the kernel optimization training. According to Sect. 3, we know that the defense method’s task is to

maximize Eq. (10), that is, to minimize function A in Eq. (11). The gradient descent method is used to evaluate the function A , as described in Algorithm 2. The graph of the value of function A varying with the value of σ is shown in Fig. 2. We found that the value of the function A grows with the increase of σ on the two datasets. Therefore, the minimum value of the function A is obtained at the initial value of σ on both datasets.

Then we verify the effectiveness of the defense method of σ at different values. We use a method that we proposed in Sect. 2.2 to generate adversarial samples. In order to prevent the gradient from disappearing, we add a small value $\eta = 0.02$ to the disturbance each time we generate adversarial samples. The method used to generate the adversarial sample is shown in Algorithm 2. On the MNIST dataset, we selected the value of σ as 8.6 (initial value of the σ), 20, 40, and 100, respectively, and then compared the generated adversarial samples (as shown in Fig. 3 on the top). On the CIFAR-10 dataset, we selected the value of σ 19.6 (the initial value of σ), 30, 40, and 50, and then compared the resulting adversarial samples (as shown in Fig. 3 on the bottom).

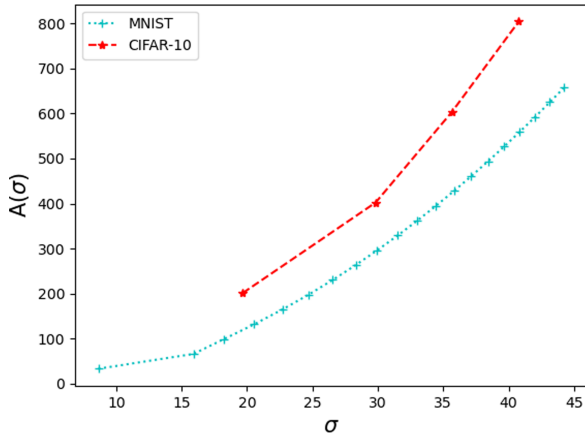


Fig. 2. How the function A changes with different values of σ on MNIST and CIFAR-10 datasets. The picture shows that function A and σ are positively correlated.

Finally, we verified the robustness of the classifier under different values of the kernel parameters. As shown in Fig. 4, after kernel optimization, it significantly increased the robustness of the classifier.

5 Discussion and Conclusion

In this work, we are the first to propose a strategy for protecting SVMs against the adversarial generation method which is based on kernel. In [26], they put forward a technique based on hyperplane classification for generating adversarial examples of deep neural networks. We think a similar approach could also work for SVMs, namely applying it to SVM classifiers. Through experiments, it is confirmed that this method was beneficial on SVM, especially on MNIST dataset, which have been caused by

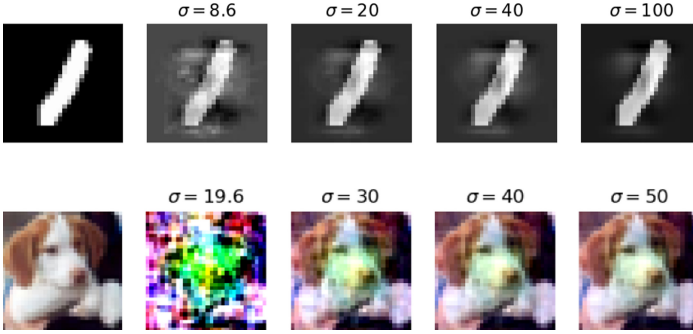


Fig. 3. Different defense effects. The figure on the top was the result obtained on the MNIST dataset. On the top row, the first picture is the original example, representing the digit ‘1’, and the other four pictures are the adversarial samples generated by the initial sample under different kernel parameters, representing the digit ‘7’. The picture below shows the results of the CIFAR-10 dataset. On the bottom row, the first one is the original example, which represents ‘dog’. The other four are the adversarial samples generated by the first one under different σ , which is meant ‘cat’.

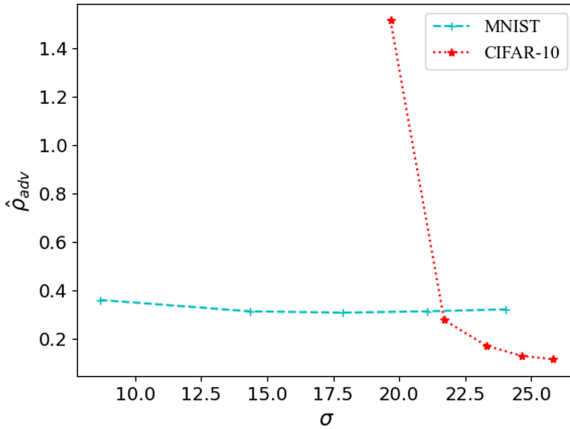


Fig. 4. Relation diagram between the robustness of the classifier and the kernel parameter on MNIST and CIFAR-10 datasets. As the value of σ increases, the robustness of the classifier will decrease. The performance is more obvious on the CIFAR-10 dataset.

nearly 100% misclassification. According to this phenomenon, we proposed a strategy for protecting SVMs against the adversarial attack. This defense approach is based on the kernel optimization of SVM. We extensively evaluate our proposed attack and defense algorithms on MNIST and CIFAR-10 datasets.

According to Fig. 3, we found that when the initial value of the σ , that is, the minimum value of its corresponding function A (see Fig. 2), was taken, there was the largest perturbation required to generate the adversarial sample, which means that the defenses are at their best. This finding holds for both datasets. The experimental results also show that our proposed defense method can effectively increase the price of attackers

and achieve a robust performance (see Fig. 4). This gives the classifier's designer a better picture of the classifier performance under adversarial attacks.

In this paper, we first described a practical attack method which has already confirmed to be effective. Then we proposed a defense method which is based on kernel. The experimental results demonstrated that the defense method is useful and effective to the security of SVM. Finally, we believe that our work will inspire future research towards developing more secure learning algorithms against adversarial attacks.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant No. 61966011.

References

1. Vorobeychik, Y.: Adversarial machine learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 12, no. (3) pp. 1–169 (2018)
2. Kumar, R.S.S.: Adversarial machine learning-industry perspectives. In: 2020 IEEE Security and Privacy Workshops (SPW), pp. 69–75. IEEE (2020)
3. Kianpour, M., Wen, S.-F.: Timing attacks on machine learning: state of the art. In: Bi, Y., Bhatia, R., Kapoor, S. (eds.) IntelliSys 2019. AISC, vol. 1037, pp. 111–125. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-29516-5_10
4. Goodfellow, I.: Making machine learning robust against adversarial inputs. *Commun. ACM* **61**(7), 56–66 (2018)
5. Jati, A.: Adversarial attack and defense strategies for deep speaker recognition systems. *Comput. Speech Lang.* **68**, 101199 (2021)
6. Islam, M.S.: Efficient hardware malware detectors that are resilient to adversarial evasion. *IEEE Trans. Comput.* (2021)
7. Papernot, N.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), Washington, pp. 582–597. IEEE (2016)
8. Prakash, A.: Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8571–8580 (2018)
9. Zheng, H.: Efficient adversarial training with transferable adversarial examples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1181–1190 (2020)
10. Xu, J.: Adversarial defense via local flatness regularization. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 2196–2200. IEEE (2020)
11. Ma, Y., Guo, G. (eds.): Support Vector Machines Applications. Springer, Cham (2014). <https://doi.org/10.1007/978-3-319-02300-7>
12. Gu, J.: A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Comput. Secur.* **86**, 53–62 (2019)
13. Zamil, Y.: Spam image email filtering using K-NN and SVM. *Int. J. Electr. Comput. Eng.* **9**(1), 2088–8708 (2019)
14. Biggio, B.: Wild patterns: ten years after the rise of adversarial machine learning. *Pattern Recogn.* **84**, 317–331 (2018)
15. Biggio, B.: Poisoning attacks against support vector machines. In: 29th International Conference on Machine Learning, pp.1807–1814. [arXiv:1206.6389](https://arxiv.org/abs/1206.6389) (2012)
16. Koh, P.W.: Stronger data poisoning attacks break data sanitization defenses. [arXiv:1811.00741](https://arxiv.org/abs/1811.00741) (2018)

17. Mei, S.: Using machine teaching to identify optimal training-set attacks on machine learners. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2871–2877 (2015)
18. Xiao, H.: Is feature selection secure against training data poisoning? In: 32th International Conference on Machine Learning, pp. 1689–1698 (2015)
19. Xiao, X.: Adversarial label flips attack on support vector machines. In: ECAI, pp. 870–875 (2012)
20. Laishram, R.: Curie: A method for protecting SVM Classifier from Poisoning Attack. [arXiv:1606.01584](https://arxiv.org/abs/1606.01584) (2016)
21. Weerasinghe, S.: Support vector machines resilient against training data integrity attacks. *Pattern Recogn.* **96**, 106985 (2019)
22. 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
23. Goodfellow, I.: Explaining and harnessing adversarial examples. [arXiv:1412.6572](https://arxiv.org/abs/1412.6572) (2014)
24. Kurakin, A.: Adversarial machine learning at scale. [arXiv:1611.01236](https://arxiv.org/abs/1611.01236) (2016)
25. Kurakin, A.: Adversarial examples in the physical world. [arXiv:1607.02533](https://arxiv.org/abs/1607.02533) (2016)
26. Moosavi-Dezfooli, S.M.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2574–2582 (2016)
27. Boser, B.E.: A training algorithm for optimal margin classifier. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp.144–152 (1992)
28. LeCun, Y.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
29. Krizhevsky, A.: Learning multiple layers of features from tiny images. Citeseer (2009)