Pattern Recognition 96 (2019) 106985

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Support vector machines resilient against training data integrity attacks

Sandamal Weerasinghe[a,*], Sarah M. Erfani[b], Tansu Alpcan[a], Christopher Leckie[b]

[a] Electrical and Electronic Engineering Department, University of Melbourne, Parkville, Victoria 3010, Australia
[b] School of Computing and Information Systems, University of Melbourne, Parkville, Victoria 3010, Australia

ARTICLE INFO

Article history:
Received 2 January 2019
Revised 31 May 2019
Accepted 29 July 2019
Available online 1 August 2019

Keywords:
Support Vector Machines
Integrity attack

ABSTRACT

Support Vector Machines (SVMs) are vulnerable to integrity attacks, where malicious attackers distort the training data in order to compromise the decision boundary of the learned model. With increasing real-world applications of SVMs, malicious data that is classified as innocuous may have harmful consequences. This paper presents a novel framework that utilizes adversarial learning, nonlinear data projections, and game theory to improve the resilience of SVMs against such training-data-integrity attacks. The proposed approach introduces a layer of uncertainty through the use of random projections on top of the learners, making it challenging for the adversary to guess the specific configurations of the learners. To find appropriate projection directions, we introduce novel indices that ensure the contraction of the data and maximize the detection accuracy. Experiments with benchmark data sets show increases in detection rates up to 13.5% for OCSVMs and up to 14.1% for binary SVMs under different attack algorithms when compared with the respective base algorithms.

1. Introduction

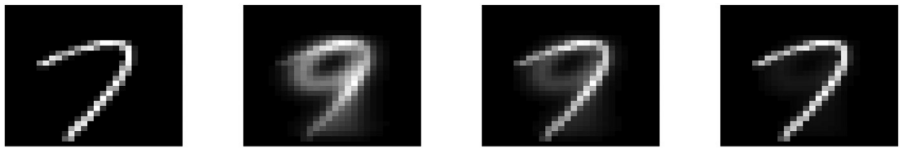
Since their introduction, Support Vector Machines (SVMs) [1] have been successfully applied to a wide range of domains such as intrusion detection, image recognition, and bioinformatics [2]. Binary SVMs are primarily used for classification problems where the learner has access to labeled training data with balanced classes. In many SVM implementations, class weights are used to compensate for imbalanced classes. In contrast, its unsupervised counterpart, One-Class Support Vector Machines (OCSVMs) [3], are used for anomaly detection problems where the learner does not have access to training labels and the classes are significantly imbalanced. Binary SVMs and OCSVMs are designed to withstand the effects of random noise in data [4]. However, their performance may degrade significantly when malicious adversaries deliberately alter the training data. It has become imperative to secure machine learning systems against such adversaries due to increased automation in many day to day applications. For example, if machine learning algorithms utilized in safety-critical environments (e.g., airports, power plants) are compromised by adversaries, it could result in loss of human lives.

Adversarial learning [5] is a broad field, which covers a whole range of attacks across different machine learning algorithms. Adversarial attacks on learning systems can be divided into two main categories, poisoning attacks during training and evasion attacks during testing [6]. In many applications, learned models are periodically updated using new batches of data in order to adapt to the natural evolution of data. This periodic updating provides an opportunity for adversaries to inject malicious data into the training process and carry out poisoning attacks. By injecting adversarial samples into the training dataset, the attackers aim to make the learner learn a decision boundary that is significantly different from the boundary it would have obtained if the dataset was not compromised. The mechanism used for generating adversarial samples depends on the objective of the adversary and the knowledge the adversary possesses about the learning system [7–10].

Poisoning attacks can be further divided into two categories based on the attacker's objective: attacks on integrity and attacks on availability. Although the attacker's objectives are different, these attack types are closely related in terms of how the attacks are carried out. In an integrity attack, the attacker purposely distorts a portion of the attack/anomaly data points that are used for training. Such integrity attacks result in a compromised decision boundary that exaggerates the region where innocuous data points lie. Subsequently, during the evaluation phase, the learner classifies specific attack data points as innocuous ones, which can cause significant harm in mission-critical applications. In an availability at-

* Corresponding author.
E-mail address: pweerasinghe@student.unimelb.edu.au (S. Weerasinghe).

https://doi.org/10.1016/j.patcog.2019.106985
0031-3203/



(a) Anomaly (b) 0.2 (c) 0.5 (d) 0.8

Fig. 1. A digit from anomaly class ('7') distorted by the adversary using different s_{attack} values to appear like a digit from the normal class ('9').

tack, the attacker's objective is to force the learned model to classify innocuous samples as attack/anomaly samples during testing, denying them access to the resource protected by the learning system. This paper focuses specifically on SVMs and OCSVMs and addresses the following key question: "Is it possible to make SVMs more resistant to poisoning attacks on integrity?".

Consider the illustrative example of digit '9' as the normal class and digit '7' as the anomaly class in an image anomaly detection setting (Fig. 1). Assume a hypothetical anomaly detection algorithm that attempts to identify the smallest hypersphere that contains the images of digit '9'. The objective of the adversary in such a situation would be to maximize the radius of the minimum enclosing hypersphere. The adversary can achieve this by injecting data points (i.e., images) of digit "7" that are distorted to appear as digit "9" into the training set. Consider a parameter $s_{attack} \in [0, 1]$ that controls the severity of the attack. An image of digit "7" that closely resembles a digit "9" (small s_{attack}) would be considered as a *moderate attack*, whereas, digit "7" that actually resembles a "7" (large s_{attack}) would be considered a *severe attack*. As Fig. 1 shows, after a less severe attack (e.g., 0.2), a digit "7" resembles a "9" visually, but as the attack severity increases, the digit tends to look like a "7" even though the learner considers it as a "9'. In practice, such attacks can be carried out in scenarios where the attacker has the opportunity to introduce distorted samples to the training process. For example, this can occur when data is collected using crowdsourcing marketplaces, where organizations build data sets with the help of a large group of people, or when malware examples are collected using honeypots which mimic likely targets of cyberattacks to lure attackers. In such scenarios, attackers can place adversarial samples among the normal data samples which would later be used by the learners for training.

Among adversarial defense techniques for SVMs, most works in the literature alter the optimization problem of SVMs in order to thwart an adversary's attacks [8,9]. Meanwhile, recent works in the literature use nonlinear random projections to improve the training and evaluation times of kernel machines, without significantly compromising the accuracy of the trained models [11,12]. In this paper, we show that under adversarial conditions, selective nonlinear projections can be leveraged as a defense technique for learners (SVMs/OCSVMs) as well. The learners gain an additional advantage due to the uncertainty that comes from the randomness of the projections. To the best of our knowledge, no existing work has explored the use of nonlinear random projections for adversarial defense.

The learner leverages the theory of low rank kernel approximation (using nonlinear projections) which facilitates large-scale, data-oriented decision making by reducing the number of optimization parameters and variables. As not all random projections result in low rank representations that mask the adversarial distortions, we introduce novel indices to identify prospective projection directions that could provide resistance against adversaries. In addition, we design security games [13] and model the adversary-learner interaction as non-cooperative, two-player, nonzero-sum games with the strategies and utility functions formulated around a nonlinear data-projection-based algorithm. The equilibrium solutions obtained from the games are used to predict the adversary's behavior and decide on advantageous configurations for the learner [14]. The main **contributions** of this work are summarized as follows:

1. We theoretically analyze the effects of adversarial distortions on the separating margins of binary SVMs and OCSVMs trained on data that has been **nonlinearly projected** to a lower dimensional space. We provide an upper-bound for the length of the weight vector when there are no adversarial distortions, using the length of the weight vector when adversarial distortions are present. We prove a similar bound for OCSVMs under **linear projections** when the data is assumed to be linearly separable.
2. We introduce a unique framework that incorporates nonlinear data projections to minimize the adversary's attempts to mask their activities, SVMs for learning, and game theory to predict the behavior of adversaries and take the necessary countermeasures. As part of this framework, we introduce novel indices based on the Dunn's index [15] to identify suitable directions for nonlinear data projections.
3. We pose the problem of finding an appropriate defense mechanism as a game, and find the Nash equilibrium solutions that give us insights into what the attacker may do and what precautionary strategy the learner should take.
4. We show through numerical experiments conducted with benchmark data sets as well as OMNET++ simulations that our proposed approach can (i) increase the attack resistance of OCSVMs (up to 13.5%) and SVMs (up to 14.1%) under adversarial conditions, and (ii) give the learner a significant advantage from a security perspective by adding a layer of unpredictability through the randomness of the data projection, making it very difficult for the adversary to guess the projected space used by the learner.

2. Background and related work

This study builds upon the extended abstract [16], where the effects of adversarial distortions on the separating margin of OCSVMs under nonlinear projections were theoretically analyzed. In this work we extend the analysis to binary SVMs as well as for linear projections. In the short paper [17], the anomaly detection approach using OCSVMs was applied to a specific networking application and numerical results were provided based on simulations. In this paper, we provide the complete numerical results for OCSVMs and binary SVMs on several benchmark datasets. As our proposed approach on adversarial learning for SVMs is based on randomized kernels, in this section we briefly review these two lines of research.

Randomized Kernels for SVMs. To improve the efficiency of kernel machines, Rahimi and Recht [11] embedded a random projection into the kernel formulation. They introduced a novel, data independent method (RKS) that approximates a kernel function by mapping the dataset to a relatively low dimensional randomized feature space. Instead of relying on the implicit transformation provided by the kernel trick, Rahimi and Recht explicitly mapped

the data to a low-dimensional Euclidean inner product space using a randomized feature map $z : \mathbb{R}^d \rightarrow \mathbb{R}^r$. The kernel value of two data points is then approximated by the dot product between their corresponding points in the transformed space z (i.e., $k(x, x') = \langle \phi(x), \phi(x') \rangle \approx z(x)z(x')$). As z is a low dimensional transformation, it is more computationally efficient to transform inputs with z and train a linear SVM as the result is comparable to that of its corresponding nonlinear SVM. Refer to [11] for more details concerning the theory of kernel approximation.

More recently, the method introduced by Rahimi and Recht [11] has been applied to other types of kernel machines. Erfani et al. [12] introduced *Randomized One-class SVMs (R1SVM)*, an unsupervised anomaly detection technique that uses randomized, nonlinear features in conjunction with a linear kernel. They reported that R1SVM reduces the training and evaluation times of OCSVMs by up to two orders of magnitude without compromising the accuracy of the predictor. Erfani et al. [18] used a deep belief network (DBN) as a nonlinear dimension reduction algorithm to transform the high-dimensional data into a low-dimensional set of features. Subsequently, a OCSVM with a linear kernel is trained on the feature vectors produced by the DBN. Our work differs from these as we look at random projections as a defense mechanism for SVMs under adversarial conditions. To the best of our knowledge, no existing work adopts Rahimi and Recht's method to address adversarial learning for SVMs.

Learning under adversarial conditions. The problem of adversarial learning has inspired a wide range of research from the machine learning community, see [19] for a security evaluation of SVMs under adversarial conditions. Poisoning attacks, which alter the training data, can be carried out either by distorting the training data or by distorting the training labels. In label noise attacks, the attackers change the labels of a subset of the training data in order to increase the SVM's classification error [20–22]. In this work, we focus on attacks that distort the training data with the purpose of harming the integrity of SVMs.

For classification using binary SVMs, Biggio et al. [7] introduced an attack algorithm that finds the optimal attack point by maximizing the hinge loss of a binary SVM when tested on a validation set. Dalvi et al. [8] modeled classification as a game between the classifier and the adversary. They extend the naive Bayes classifier to optimally detect and reclassify distorted data points, by taking into account the adversary's optimal feature-changing strategy. Zhou et al. [9] introduced an Adversarial SVM (AD-SVM) model which incorporated additional constraint conditions to the binary SVM optimization problem in order to thwart an adversary's attacks. Their model only supports data that is linearly separable, and leads to unsatisfactory results when the severity of real attacks differs from the expected attack severity by the model. Suykens and Vandewalle [23] introduced Least-Squares SVM (LS-SVM) where a quadratic loss function is used instead of the hinge loss which results in a non-sparse solution to the optimization problem (all the training samples are assigned non-zero α values). The authors claim this approach prevents the SVM from over relying on the contribution of certain samples (e.g., poisoned samples).

For anomaly detection, Kloft and Laskov [24] analyzed the effects of adversarial injections on the centroid anomaly detection (CAD) algorithm, which can be considered as a hard margin, hypersphere-based SVDD model [25] (SVDD is equivalent to OCSVM when the RBF kernel is used). As their work focuses on an online learning setting, they use different data replacement policies as the defense mechanism against integrity attacks. In our work we use a batch learning approach instead of online training and do not assume a fixed training dataset size. We also do not assume that the initial dataset consists of purely innocuous data, which is unrealistic in situations where data is collected from a real world system. Previously, Rajasegarar et al. [26] used OCSVMs in order to detect anomalous secondary users that provide misleading observations in cognitive radio networks. They utilized anomaly detection in a scenario where a central node is attempting to determine if a spectrum is being utilized by a primary user or not, with one or many malicious users providing false information in order to force the central node to make an incorrect decision. While their work is in the same application domain as the case study presented in this work, the methodology and the attack type differ.

One approach for learning in the presence of poisoned training data is to identify and remove such samples prior to training. Steinhardt et al. [27] introduced a framework that uses an outlier detector prior to training in order to filter out poisoned data. They consider two scenarios, (i) where there is a clean outlier detector (trained independently without being affected by the poisoned data), and (ii) where the outlier detector is also compromised. While the framework performs well in the first scenario, the authors claim that the attacker is able to subvert the outlier removal and obtain stronger attacks in the second scenario. Laishram and Phoha [28] introduced an algorithm that clusters the data in the input space and utilizes the distances among data points in the same cluster in the (input feature + label) space to identify the outliers. These works can be considered as a pre-processing step and could be used in conjunction with our proposed framework to further increase the attack resistance of SVMs.

Vinh et al. [29] used multiple linear projections of the training data in order to train a single neural network. The random projections are used as a regularization mechanism to boost the accuracy of NNs under adversarial conditions. While this work is in the same domain, it is not directly related to our work, which utilizes nonlinear projections and uses novel indices to identify the single best projection direction among many. Wong et al. [30] use a nonlinear random projection technique to estimate the upper-bound on the robust loss for DNNs in the worst case that scales linearly in the size of the hidden units. The authors claim that the introduced robust training procedure results in networks with minimal degradation in detection accuracy. Although we use a significantly different nonlinear projection algorithm that is designed specifically for kernel based learners, our defense framework yields similar benefits in terms of minimal accuracy degradation, and lower training times for SVMs as the authors claim their technique [30] does for DNNs.

3. Problem statement: Adversarial learning against integrity attacks

We consider an adversarial learning problem in the presence of a malicious adversary. The adversary's ultimate goal is to force the learner to accept a specially crafted adversarial point as innocuous after some learning iteration. To clarify, if we consider the innocuous class as the negative class, and the attack class as the positive class, the attacker would want to increase the false negative rate (FNR). To succeed in this, the attacker injects malicious training data in order to alter the decision boundary of the learner in a manner favorable to him/her. Subsequently, during the testing phase, it would be easier for the attacker to craft adversarial data points that still retain their harmful qualities, but are classified by the learner as innocuous. In the following section, we formalize the underlying problem using the attack strategy of the adversary.

Adversarial attack model. In the context of OCSVMs, the decision boundary (i.e., the separating hyperplane) is located closer to the normal data cloud and the undistorted anomalies lie close to the origin. The adversary would distort anomalies in order to shift them closer to the normal data cloud. Since the OCSVM algorithm

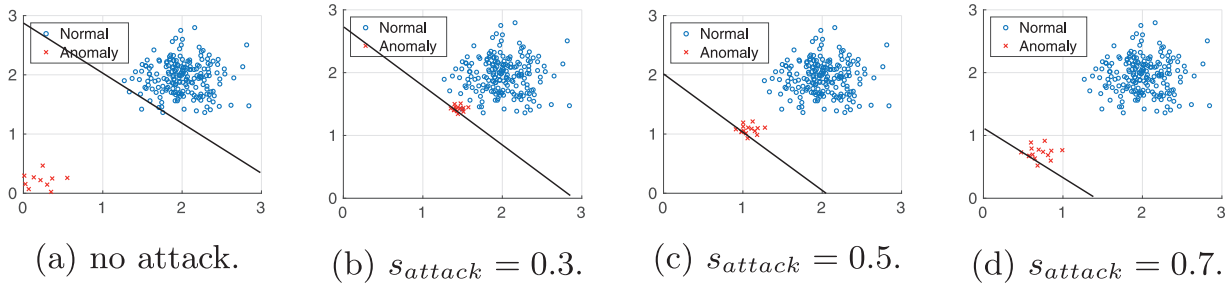


Fig. 2. Training data distribution and separating hyperplane (black line) of a toy problem under different attack severities. "o" (blue) denotes the undistorted data points and "x" (red) denotes the data points distorted by the adversary. The OCSVM is trained considering the entire (unlabeled) dataset is from one class. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

considers all the data points in the training set to be from a single class, these distorted anomalies would shift the separating hyperplane in the direction of the attack points (towards the origin). In binary SVMs, the decision boundary lies between the data clouds of the two classes. Therefore, when the adversary distorts attack data and shift them closer to the normal data (with the labels flipped), the decision boundary would shift towards the attack data class.

The adversary is able to orchestrate different attacks by changing the percentage of distorted attack data points in the training dataset (i.e., p_{attack}) in addition to the severity of the distortion (i.e., s_{attack}). Fig. 2 illustrates the data distributions when different levels of attack severities are applied to the anomaly data in a OCSVM problem. When there is no attack (i.e., 2 a), the undistorted anomalies lie closer to the origin and the OCSVM disregards their contribution to the decision boundary by considering them as outliers. In the presence of an attack (e.g., 2 c), the distorted anomalies are positioned much closer to the normal data cloud, and the OCSVM's decision boundary is influenced by their contribution. As s_{attack} increases, the anomaly data points are moved closer to the origin, reducing the gap between the origin and the separating hyperplane.

Let $X \in \mathbb{R}^{n \times d}$ be the training dataset and $T \in \mathbb{R}^{n \times d}$ be the distortions made by the adversary, making $X + T$ the training dataset that has been distorted (if the i^{th} data point is not distorted, T_i is a vector of zeros). It should be noted that the learner cannot demarcate T from $(X + T)$, otherwise the learner would be able to remove the adversarial distortions during training, making the problem trivial. The adversary has the freedom to determine T based on the knowledge it possesses regarding the learning system, although the magnitude of T is usually bounded due to its limited knowledge about the learners" configuration, the increased risk of being discovered, and computational constraints.

The attack model in this work is inspired by the restrained attack model described in [9], where it is assumed that the adversary has the capability to move the i^{th} data point in any direction by adding a non-zero displacement vector $\kappa_i \in T$ to $x_i \in X$. It is also assumed that the adversary does not have any knowledge about the projection used by the learner. Therefore, all of the adversary's actions take place in the input space. The adversary picks a target x_i^t for each x_i to be distorted and moves it towards the target by some amount. Choosing x_i^t for each x_i optimally requires a significant level of computational effort and a thorough knowledge about the distribution of the data.

We assume that the attacker knows the distribution of the normal data used by the learner. While this does exaggerate the adversary's capabilities, in real-world applications there is a possibility to approximate this distribution based on domain knowledge or prior experience. For example, an attacker posing as a data supplier in a crowdsourcing marketplace could use the data samples of legitimate suppliers to approximate the distribution of normal data. This approach also tests our proposed defense framework in a worst-case scenario. Please note that the theoretical analyses we present in Section 5 do not depend on this assumption. In the analyses, the adversarial distortions added to the data can come from any poisoning attack algorithm under which Assumption 1 holds.

The attacker, similar to [9], uses the centroid of the normal data cloud in the training set as the target point for all anomaly data points that it intends to distort. A data point sampled from the normal data cloud or an artificial data point generated from the estimated normal data distribution could be used as alternatives. For each feature j in the input feature space, the adversary is able to add κ_{ij} to x_{ij} as follows where $s_{attack} \in [0, 1]$ controls the severity of the attack,

$$\kappa_{ij} = (1 - s_{attack})(x_{ij}^t - x_{ij}) \text{ and } |\kappa_{ij}| \leq |x_{ij}^t - x_{ij}|, \forall j \in d. \quad (1)$$

4. Defense framework against integrity attacks

Our novel defense framework consists of three main components: (1) a selective randomized projection using a novel metric that increases attack resistance by masking the adversary's distortions, (2) SVMs for learning and predicting, and (3) a game-theoretic model that supports defensive decision-making by considering the best responses of the players (Fig. 3).

In order to increase the attack resistance of a learning system, the impact of adversarial inputs should be minimized. Therefore, at the heart of our framework we use a projection mechanism that projects data points to lower dimensional spaces in a manner that conceals the potential distortions of an adversary. Projecting a high dimensional dataset, using a carefully chosen projection matrix would preserve its pairwise Euclidean distances with high probability in the projected space [31]. Therefore, the properties of the original data distribution would be present in the projected dataset with only minor perturbations. Assuming the existence of a lower dimensional intrinsic subspace, the learner projects the data to a lower dimensional space using a projection matrix $A \in \mathbb{R}^{d \times r}$, i.e., $(X + T)A$. In this work, the learner trains a linear SVM in a low dimensional space where the data has been nonlinearly transformed using the algorithm introduced by Rahimi and Recht [11], instead of training a nonlinear SVM with a RBF kernel. Therefore, this direction is drawn from the Fourier transform of the shift invariant kernel being approximated. For the RBF kernel, A is sampled from $\mathcal{N}(0, 1)$.

By randomly drawing projection directions from some distribution, the learner also introduces a layer of uncertainty to the adversary-learner problem. For high dimensional datasets, this method gives the learner considerable flexibility to select the dimension to which the data is projected, as well as the direction. Therefore the learner gains a significant advantage from a secu-

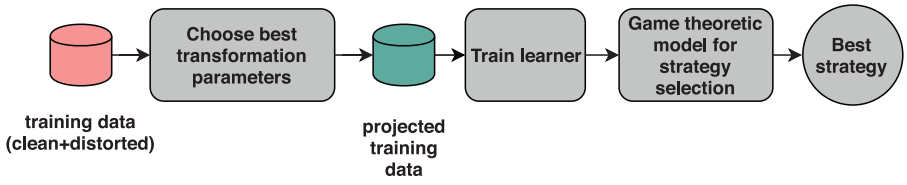


Fig. 3. Flowchart of our proposed defense framework.

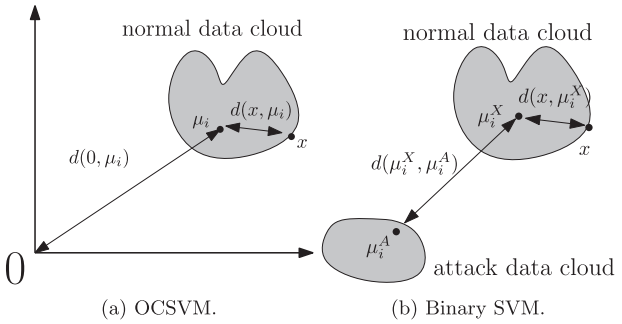


Fig. 4. Graphical illustrations of the compactness indices for OCSVMs and binary SVMs. The normal data cloud is shown in blue while the attack data cloud is shown in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

rity perspective as this expands the search space for the adversary. But this unpredictability can also be seen as the main caveat of using random projections to reduce the dimensionality of data. While some random projections result in better separated volumetric clouds than the original ones, some projections result in the data from different classes being overlapped. As the learner cannot demarcate T from the training data, it is not possible to identify an ideal projection that conceals the adversarial distortions.

Thus, in a OCSVM problem, the learner would have to select a projection that contracts the entire training set (expecting the adversarial points to be masked by normal data) and separates the training data from the origin with the largest margin in the projected space. Therefore, motivated by a generalized version of the Dunn's index [15], we propose a compactness measure to rank suitable projection directions in a one-class problem. The learner draws multiple samples from $\mathcal{N}(0, 1)$ for the projection matrix A and ranks them using Eq. 2. The projection direction A that gives the highest compactness value is considered as the projection that gives the best attack resistance as it gives a compact normal data cloud that is well separated from the origin. The compactness of projection P_i, where μ_i is the centroid of the projected training set, $\mathbf{0}$ is the origin in the projected space, and the function d is the Euclidean distance, can be calculated as

$$c_i^1 = \frac{d(0, \mu_i)}{\left(\sum_{x \in P_i} d(x, \mu_i)\right)/n}. \tag{2}$$

In a binary SVM problem, the learner would have to select the projection that contracts the innocuous data and separates it from the attack data with the largest distance. Therefore, we devise a similar compactness measure where μ_i^X is the centroid of the projected innocuous data (contains the adversarial distortions as well) and μ_i^A is the centroid of the projected attack data (Fig. 4).

$$c_i^2 = \frac{d(\mu_i^X, \mu_i^A)}{\left(\sum_{x \in P_i} d(x, \mu_i^X)\right)/n}. \tag{3}$$

Following the linear projection, each i^{th} sample $(X+T)_i A$ is then nonlinearly transformed using the function

$$z((X+T)_i) = \frac{\sqrt{2}}{r} \cos\left(\sqrt{2\gamma}(X+T)_i A + b\right), \tag{4}$$

where γ is a parameter taken from the RBF kernel being approximated, r is the dimension to which the data is projected, d is the input space dimension and b is an r-dimensional vector whose elements are drawn uniformly from $[0, 2\pi]$ [11]. This transformation projects the data onto the interval [0,1]. The approach used by the learner to identify suitable projection directions in a OCSVM problem is formalized in Algorithm 1 in terms of the random projection

Algorithm 1 Identifying compact projections.

input $(X+T)$, r, number of samples N
$A', b' \leftarrow null\sim$ ▷ projection parameters
for $i \leftarrow 1, N$ **do**~ ▷ sample N random directions
  Draw A from $\mathcal{N}(0, , 1)\sim$ ▷ sample A
  $c_i^1 \leftarrow calculate\_compactness((X+T)A)$ ▷ calculate compactness. (Eq.2)
  if $c_i^1 > max\_compactness$ **then**
    $max\_compactness \leftarrow c_i^1$
    $A' \leftarrow A$
  end if
end for
$[(X+T)^*, b'] \leftarrow z(X+T, A')\sim$ ▷ nonlinear projection (Eq.4)
output A', b' ▷ Return best projection parameters

parameters A and b, the dimension of the projected dataset r and the adversary's data distortion strategy T.

The next component in our framework is the learning algorithm. Anomaly detection problems are addressed in this paper using the OCSVM algorithm in [32], which separates the training data from the origin with a maximal margin in the feature space. For classification problems, the binary SVM algorithm introduced in [1] is used.

4.1. Game theoretic models to identify best defense strategies

In the final component of our framework, we pose the adversary-learner interaction as a bimatrix game. Using the formulated game, the learner can (i) predict the possible actions of the adversary, and (ii) decide what countermeasure to take in order to thwart the adversary's attempts. In this section, we present a game formulation that can be employed in adversarial conditions for anomaly detection. In Section 6.1, we demonstrate how the following game can be utilized in a real world application scenario.

In the following formulation we consider the adversary (M) and learner (L) to be the two players. The adversary is unaware of the learner's configuration and projections used, but it is capable of evaluating the learned model by sending adversarial samples during testing. Similarly, the learner is unaware of the details of the adversary's attack, but is able to simulate attacks during the training process. Since the adversary can vary the severity of the attacks, we choose different s_{attack} values (keeping p_{attack} constant) as the finite set of actions (a_M) available for the adversary. As the learner uses the projection based method to detect adversarial samples, the dimensions to which the data is projected will be

used as the set of actions (a_L) available to the learner.

$$\begin{aligned} a_M &\in \{0, 0.3, 0.4, 0.5, 0.6\}, \\ a_L &\in \{20\%, 40\%, 60\%, 80\%, 100\%\}. \end{aligned} \tag{5}$$

A bimatrix game comprises two matrices, $G = \{g_{i,j}\}$ and $H = \{h_{i,j}\}$ where each pair of entries $(g_{i,j}, h_{i,j})$ denotes the outcome of the game corresponding to a particular pair of decisions made by the players. These entries in the matrix are populated by the players' (adversary and learner) utility functions, $u_M : a_M, a_L \to \mathbb{R}$ and $u_L : a_M, a_L \to \mathbb{R}$. A pair of strategies $(g_{i^*,j^*}, h_{i^*,j^*})$ is said to be a non-cooperative Nash equilibrium outcome of the bimatrix game if there is no incentive for any unilateral deviation by any one of the players. While it is possible to have a scenario where there is no Nash equilibrium solution in pure strategies, there would always be a Nash equilibrium solution in mixed strategies [33].

Due to the adversary's ability to evaluate the model during testing (i.e., calculating the false negative rate (FNR)), we design u_M to reflect his/her desire to achieve false negatives and to penalize large adversarial distortions. This is because if the adversary greedily distorts the data, it would result in the distortions becoming quite evident and increase the risk of the attack being discovered. Similarly, the learner's utility function reflects a desire to achieve high classification accuracies, which is captured by the f-score. Note that an affine linear transformation of either of the utility functions would result in a strategically equivalent bimatrix game. All strategically equivalent games have the same Nash equilibria as shown by Basar and Olsder in Proposition 3.1 of [34]. The utility functions of the two players are, therefore, defined as

$$\begin{aligned} u_M(a_M, a_L) &= 1 + FNR - \tfrac{1}{2} s_{attack}, \\ u_L(a_M, a_L) &= f\text{-}score. \end{aligned} \tag{6}$$

5. Analysing the impact of adversarial distortions on the separation margin

This section analyzes the effects of the adversary's distortions on the separation margin of OCSVMs and SVMs when the data is projected to low dimensional spaces. We investigate nonlinear projections as well as linear projections. Although the projection direction is randomized, the following analyses are conditional on the direction chosen by the defense framework. The margin of OCSVMs and SVMs is largely dependent on the regularization parameter ν, therefore in the following analyses we take ν to be fixed to a value chosen prior to learning.

5.1. Nonlinear projections

5.1.1. Attack effectiveness on the OCSVM margin

Let w_p^* be the primal solution of the OCSVM optimization problem in the projected space without adversarial distortions. Similarly, define w_{pt}^* as the primal solution in the presence of a malicious adversary. Since the learner cannot demarcate the distortions from the normal training data, it cannot empirically calculate $\|w_p^*\|_2$. Therefore, based on the assumptions given below, we analytically derive an upper-bound on $\|w_p^*\|_2$ of a OCSVM that has been trained on a nonlinearly projected undistorted dataset.

As the adversary distorts data in the input feature space, we can align any given dataset in such a way that any outliers present in the data would lie closer to the origin and the normal data would lie in the positive orthant. Such a projection would compel the adversary to make adversarial distortions in the direction of the normal data cloud (positive) as distortions in the negative direction would favor the learner (Fig. 2).

Definition 1. Let $X \in \mathbb{R}^{n \times d}$ be the matrix that contains the training data (normalized between $0 \; -\!- \; 1$) and $T \in \mathbb{R}^{n \times d}$ the matrix that contains the adversarial distortions. Let $A \in \mathbb{R}^{d \times r}$ be the projection matrix where each element is an i.i.d. $\mathcal{N}(0, 1)$ random variable. Define b as a $1 \times r$ row vector where each element is drawn uniformly from $[0, 2\pi]$. Using these variables, we define $C \in \mathbb{R}^{n \times r}$ (which is linearly separable [11]), where the element at row i column j takes the following form.

$$\begin{aligned} C_{i,j} = \cos\Big(\Big[&\big(X_{i,1} + T_{i,1}\big)A_{1,j} + \big(X_{i,2} + T_{i,2}\big)A_{2,j} + \dots \\ &+ \big(X_{i,d} + T_{i,d}\big)A_{d,j}\Big] + b_{1,j}\Big). \end{aligned} \tag{7}$$

Similarly, we define the matrices C^X, C^T, S^X and S^T where the element at row i column j is defined as,

$$\begin{aligned} C^X_{i,j} &= \cos\Big(\big[X_{i,1}A_{1,j} + X_{i,2}A_{2,j} + \dots + X_{i,d}A_{d,j}\big] + b_{1,j}\Big), \\ C^T_{i,j} &= \cos\Big(\big[T_{i,1}A_{1,j} + T_{i,2}A_{2,j} + \dots + T_{i,d}A_{d,j}\big]\Big), \\ S^X_{i,j} &= \sin\Big(\big[X_{i,1}A_{1,j} + X_{i,2}A_{2,j} + \dots + X_{i,d}A_{d,j}\big] + b_{1,j}\Big), \\ S^T_{i,j} &= \sin\Big(\big[T_{i,1}A_{1,j} + T_{i,2}A_{2,j} + \dots + T_{i,d}A_{d,j}\big]\Big). \end{aligned}$$

We address the anomaly detection problem using the OCSVM algorithm introduced by [32], which separates the training data from the origin with a maximal margin in the projected space. Following the above nonlinear transformation of data, the dual form of the OCSVM algorithm can be written in matrix notation as

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\alpha^T CC^T \alpha, \;\; \text{s.t } 0 \le \alpha \le \frac{1}{\nu n} \text{ and } \mathbf{1}^T\alpha = 1. \tag{8}$$

Assumption 1. The distortions made by the adversary are small s.t. the small angle approximation $\cos(\theta) = 1 - \frac{\theta^2}{2}$ holds.

This assumption is reasonable because small distortions decrease the risk of the adversary being discovered, therefore a rational adversary would refrain from conducting attacks with significant distortions.

Theorem 1. *Let r be the dimension to which the data is projected using the method in (7). Then, if Assumption 1 holds, the length of the weight vector w_p^* of a OCSVM is bounded above by*

$$\|w_p^*\|_2 \le \|w_{pt}^*\|_2 + \frac{3\sqrt{r}}{2}. \tag{9}$$

We defer the proof of Theorem 1 to A.1.

5.1.2. Attack effectiveness on the binary SVM margin

We address the classification problem using the ν-SVC algorithm, which uses the parameter ν to adjust the proportion of outliers, similar to the OCSVM algorithm [35]. Similar to the bound on OCSVMs, we analytically derive an upper-bound on $\|w\|_2$ of a binary SVM that has been trained on a nonlinearly projected undistorted dataset.

Definition 1. The dual form of the ν-SVC classification algorithm is defined as,

$$\underset{\alpha}{\text{minimize}} \frac{1}{2}\alpha^T YCC^T Y\alpha, \;\; \text{s.t } 0 \le \alpha \le \frac{1}{n}, \;\; \mathbf{1}^T Y\alpha = 0 \text{ and } \mathbf{1}^T\alpha \ge \nu, \tag{10}$$

where Y is a $n \times n$ diagonal matrix that contains the labels.

Theorem 2. *If Assumption 1 holds, the length of the weight vector w_p^* of a binary SVM is bounded above,*

$$\|w_p^*\|_2 \le \|w_{pt}^*\|_2. \tag{11}$$

The proof follows the same steps followed in A.1 for OCSVMs. Using the constraint condition $\mathbf{1}^T Y\alpha = 0$ of the optimization problem instead of $\mathbf{1}^T\alpha = 1$ in Eq. A.7 leads to a trivial upper-bound for the binary SVM problem that is independent of the projection dimension r.

5.2. Linear projections

5.2.1. Attack effectiveness on the OCSVM margin

The analysis adopts the approaches in [14] and [36] for anomaly detection using OCSVMs without labeled data. We use the length of the weight vector of a OCSVM (with a linear kernel) trained on a linearly projected, distorted dataset and present an upper-bound for the length of the weight vector of a OCSVM (with a linear kernel) trained in the input space of the data without any adversarial distortions.

Definition 2. Let $V \in \mathbb{R}^{d\times d}$ be any matrix with orthonormal columns. Define $E := V^TV - V^TAA^TV$, and assume $\|E\|_2 < \epsilon, \epsilon \in (0, \frac{1}{2}]$ for a given A.

Theorem 3. *Let w^* be the primal solution of the OCSVM optimization problem in the input feature space without adversarial distortions. Similarly, define w^*_{pt} as the primal solution in the presence of a malicious adversary in the projected space. Then the length of the weight vector w^* is bounded above by*

$$\|w^*\|_2^2 \leq \frac{(1+\lambda)}{|(1-\delta)|^2}\|w^*_{pt}\|_2^2, \tag{12}$$

where, $\delta := \dfrac{\|\alpha^T TA\|_2}{\|\alpha^T XA\|_2}$ *and* $\lambda = \dfrac{1}{2}\dfrac{\|E\|_2}{(1-\|E\|_2)}$.

We defer the proof of Theorem 3 to A.2.

5.3. Discussion of the theorems

In Theorems 1 and 2, we derive an upper-bound for the norm of the weight vector $\|w^*_p\|_2$ of a SVM/OCSVM that has been trained on projected undistorted data. The separating margin of a OCSVM is given by $\rho/\|w\|_2$, where ρ is the offset and w is the vector of weights. Similarly, in binary SVMs, the data from two classes are separated by the margin $2\rho/\|w\|_2$. This implies that a small $\|w\|_2$ corresponds to a large margin of separation. Therefore, the difference between $\|w^*_p\|_2$ and $\|w^*_{pt}\|_2$ is therefore an indicator of the attack's effectiveness.

In both cases, the strength of the adversary's attacks will be reflected in the value of the upper-bound (i.e., $\|w^*_{pt}\|_2$ component). Therefore the upper-bounds can be used to measure the impact of different attack algorithms on SVMs/OCSVMs in the same projected space (i.e., using the same projection direction A). For OCSVMs, the learner has the advantage to make the upper-bound of $\|w^*_p\|_2$ tighter by reducing the dimensionality of the dataset (i.e., r). Therefore, by projecting the data to low dimensional spaces, the learner is able to reduce the adversary's effects on the margin of separation (i.e., $\rho/\|w\|_2$) of a OCSVM. The upper-bound derived for binary SVMs is rudimentary, and the relationship between $\|w^*_p\|_2$ and $\|w^*_{pt}\|_2$ is as anticipated.

In Theorem 3, we derive an upper-bound on w^* (the primal solution of a OCSVM in the input feature space without adversarial distortions) using w^*_{pt} (the primal solution of a OCSVM in the presence of a malicious adversary in the projected space). This analysis assumes that the data is linearly separable, therefore a linear kernel is used in the OCSVM. From Theorem 3, we see that solving the OCSVM optimization problem using distorted data in the projected space results in a margin that is comparable to the margin of a OCSVM trained on undistorted data in the input feature space. The parameter δ is the ratio between the contribution from the distorted data to the objective function and the contribution from the normal data to the objective function. In the special case of $\delta = 1$, the upper-bound of $\|w^*\|_2^2$ diverges to infinity.

Table 1
Datasets used for training and testing with OCSVMs.

Dataset	Training size	Test size	Normal	Anomaly
MNIST	2000	1200	digit '9'	digit '8'
CIFAR-10	3650	1200	airplane	truck
SVHN	4200	1200	digit '8'	digit '0'

Table 2
Datasets used for training and testing with binary SVMs.

Dataset	Training size	Test size	Innocuous	Attack
MNIST	2000	1200	digit '9'	digit '1'
CIFAR-10	3800	1200	airplane	truck
SVHN	4200	1200	digit "8"	digit "0'

6. Evaluating the detection capability of the defense framework

We demonstrate the effectiveness of our novel defense framework on several datasets. This section describes how the datasets are obtained, pre-processing and other procedures of the experimental setup. We also show the applicability of our framework to real world security applications by simulating a particular network security application.

6.1. Experimental setup

Datasets. For experiments using binary SVMs, we choose data from two classes, considering one class as the innocuous class and another as the attack class. For experiments using OCSVMs, we generate single-class (unlabeled) datasets considering one of the original classes as the normal class, and a different one as the anomaly class. For each dataset, we create two test sets (with a normal to anomaly ratio of 5: 1): (i) a clean test set (called test_C) with undistorted anomaly/attack data and normal data, (ii) a distorted test set (test_D) with its anomaly/attack points distorted. Tables 1 and 2 give the class and number of samples used in each training and test set.

Experimental setup. Attacks of different intensities are conducted (creating train_D) by varying the attack severity s_{attack} and attack percentage p_{attack}. In anomaly detection problems, it is unlikely to find a large percentage of attack points within the training set, therefore we choose 5% for p_{attack} (percentage of distorted points in the training set) when using OCSVMs. For binary SVMs we choose 10%, 20%, 30% and 40% for p_{attack} as done in prior works in this area. We specifically choose the values 0.3, 0.4, 0.5 and 0.6 for s_{attack}. For comparison, we test all the attack scenarios against learners using the RBF kernel in the input feature space.

For nonlinear projections, we choose 20%, 40%, 60% and 80% of the input dimension as the dimensions to which the datasets are projected. The test sets are projected using the same parameters that give the highest compactness for the corresponding distorted training set. The learner then uses the projected training set to train a SVM model with a linear kernel, and the resulting model is evaluated using the test sets. For these experiments the ν parameter of the learners are kept fixed across all experiments conducted for each dataset. Since ν sets a lower bound on the fraction of outliers, it is crucial to keep its value fixed across different attack scenarios in order to evaluate the interplay between the adversarial distortions and the performance. As RBF kernels are used by the learners in the input space, we use the same *gamma* values in the low rank kernel approximation using Eq. 4 in order to have identical kernel parameters. The ν and γ parameters for each dataset are found by performing grid searches on clean datasets.

Evaluation metric. For comparison purposes, we also train the learners using an undistorted training set (called train_C). We report the performance against test_C and test_D using the f-score (classification performance of the learners) and AUC (classification performance around the decision boundary). We also report the FNRs, which indicate the percentage of attack data points that are classified as normal data points by the learners.

6.2. Engineering case study: Identifying adversaries from radio communication signals

In this section we present an experimental evaluation of the developed framework for OCSVMs in the context of a real world security application: identifying rogue communication nodes using captured radio signals. We present the problem being addressed, data collection and pre-processing procedures followed by the learning model.

Application setting. In a given populated area, a multitude of individuals communicate with each other using a plethora of devices. While a majority of parties utilize such devices for innocuous, day-to-day activities (civilians), there may be a few malicious individuals (rogue agents) whose purpose is to cause harm and disrupt the lives of others. Even though these rogue agents would prefer to remain hidden, they would need to communicate electronically in order to plan and coordinate their activities. Communications are increasingly encrypted at various layers for privacy reasons. It is natural to assume that rogue agents prefer to conceal their radio communications among the civilian (background) radio traffic while enjoying the privacy protection provided by encryption systems.

Identifying rogue agents based on their wireless communication patterns is not a trivial task, especially when they deliberately try to mask their activities. An inherent assumption we make is that communication patterns of the rogue agents differ from those of regular background traffic to some degree, otherwise, the detection problem would be infeasible. Rogue agents would naturally prefer to avoid the standard networks used by civilians as there is a possibility for eavesdropping attacks, but using specialized communication equipment alone would highlight their presence if the radio spectrum was to be analyzed. Therefore, we can safely assume that rogue agents would strategically utilize specialized radio equipment as well as third party network infrastructure in order to avoid detection.

The above scenario can be posed as an anomaly detection problem where the learner creates a representation of normal data (i.e., civilians) using the data captured by the sensors and attempts to identify anomalies (i.e., rogue agents). In the aforementioned problem, if the rogue agents alter their communication patterns to resemble those of civilians to some extent during the initial stages of system deployment, they would be able to inject malicious data points into the dataset that will be used by the learner to create the anomaly detection model. As the learner cannot distinguish the radio signals of the rogue agents from those of the civilians, the learner would use the entire dataset collected by the sensors to train the anomaly detection model. This would result in a deformed representation of the normal data in the learned model. Therefore, during the evaluation (operational) phase, the rogue agents would be able to evade the classifier without having to use identical communication patterns to those of the civilians.

Network simulation and data collection. Simulations are performed using the INET framework for OMNeT++ [37](datasets available at https://goo.gl/xDYD2k). In order to conduct a realistic simulation, signal attenuation, signal interference, background noise and limited radio ranges are considered. The nodes (civilians, rogue agents and listeners) are placed randomly within the given confined area. The simulator allows control of the frequencies and bit rates of the transmitter radios, their communication ranges, message sending intervals, message lengths, sensitivity of the receivers, minimum energy detection of receivers among other parameters. It is assumed that all nodes communicate securely, therefore the listeners are unable to access the content of the captured messages. Duration of reception, message length, inter arrival time (IAT), carrier frequency, bandwidth and bitrate are obtained as features using the data captured by the listeners.

Since the objective is to classify transmission sources, we consider the data received by the three closest listeners (using the power of the received signal) of each transmission source. The duration, message length and IAT of the messages received by each listener is averaged every five minutes, which results in 108 $(12 \times 3 \times 3)$ features in total. Adding the latter three parameters (fixed for each transmission source) gives the full feature vector of 111 features. Using the collected data, we create two training datasets, train_C, train_D and two test datasets test_C and test_D. The two training datasets consist of 95% civilian data points and 5% rogue agent data points, while the two test datasets consist of 80% civilian data points and 20% rogue agent data points. In both train_D and test_D, the rogue agent data points are distorted (i.e., they deliberately changed their communication patterns to deceive the learner).

Learning model. We choose 20%, 40%, 60% and 80% of the input dimension as the dimensions to which the datasets are projected. By performing a grid search for parameters using only clean data, we set $\nu = 0.13$ and $\gamma = 0.009$. Identical parameter values are used in the OCSVM with a RBF kernel in the input space as well as in the OCSVMs that used kernel approximations in the lower dimensional spaces.

Game formulation. Finally, we model the interaction between the rogue agents and the listeners as the bimatrix game explained in Section 4.1. Since the adversary can vary the severity of attacks during the initial stages of system deployment (i.e., data collection) by changing their communication parameters, we select four such communication patterns as the finite set of actions available for the adversary. If the adversary does not carry out an attack, we consider s_{attack} to be 0. If the rogue agents closely mimic the civilians (resulting in a small shift of the margin) we consider s_{attack} to be small. Conversely if rogue agents change their patterns to ones that are significantly different than those of the civilians, we consider s_{attack} to be larger. As the learner uses the projection based method to detect adversarial samples, the dimensions to which the data is projected will be used as the set of actions available for the learner.

$$a_M \in \{0, 0.3, 0.4, 0.5\}, \;\; a_L \in \{20\%, 40\%, 60\%, 80\%, 100\%\}. \tag{13}$$

We present the outcome of the game in Section 7.3.

7. Results and discussion

We demonstrate the effectiveness of the first part of the proposed defense mechanism (i.e., selective nonlinear projection) on three benchmark datasets. As most real world data are not linearly separable, we focus on the performance of nonlinear random projections using the indices introduced in Section 4 when an active adversary is conducting a directed attack by maliciously distorting the data. We extensively investigate the effectiveness of the defense framework under different attack configurations (i.e., s_{attack} and p_{attack} values) when the adversary is using the attack model described in Section 3. Then, we present the outcomes of the game

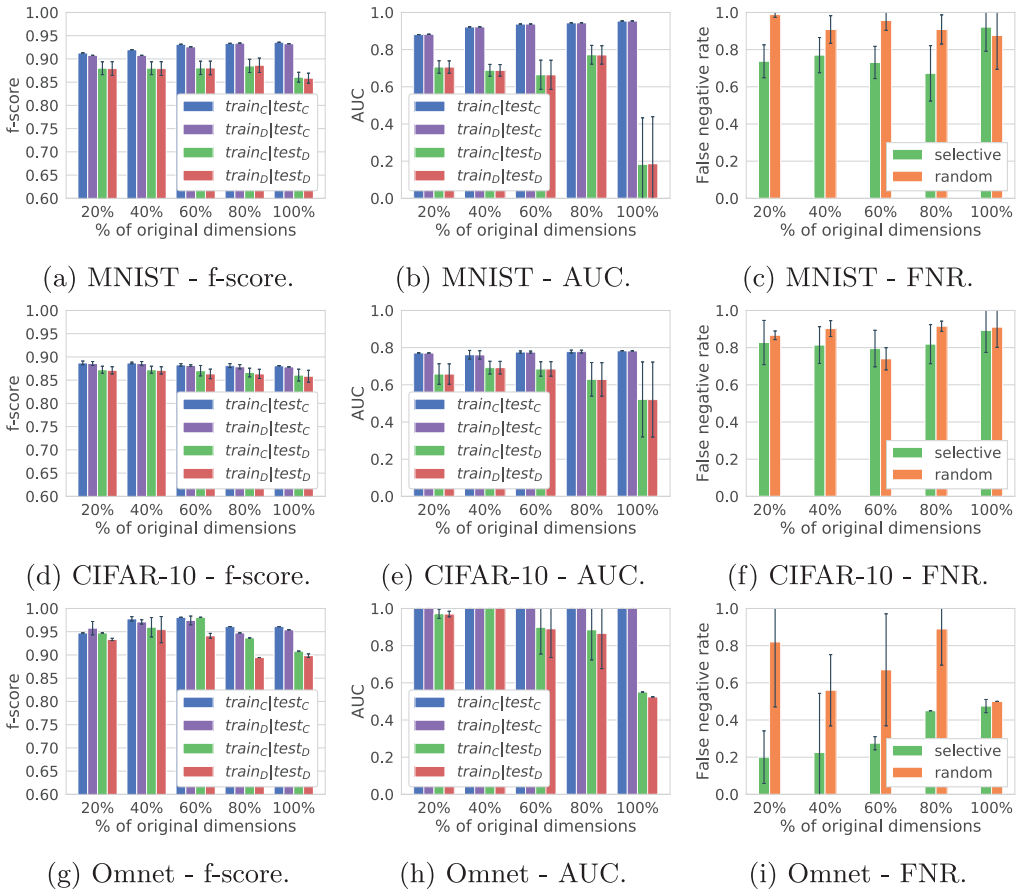


Fig. 5. The performance of OCSVMs under attacks on integrity when the training takes place in different dimensional spaces. The left column compares the f-scores of OCSVMs trained on train_C and train_D against the two test sets: test_C and test_D. The middle column shows the corresponding AUC values for each dataset. The right column compares the FNRs of OCSVMs under an integrity attack (i.e., trained on train_D and evaluated using test_D).

presented in Section 4.1 using two datasets. Finally, to further evaluate the robustness added by the framework, we compare its effectiveness against other related attack/defense strategies in the literature.

7.1. OCSVM results

Fig. 5 presents f-scores, AUC values and FNRs at different projected spaces under adversarial distortions. For f-score and AUC values, we present four results; when (i) trained using train_C, and tested with test_C; (ii) trained with train_C and tested with test_D; (iii) trained with train_D and tested with test_C; and finally (iv) trained with train_D and tested with test_D. We present the FNR values when the models are trained with train_D and tested with test_D.

Absence of an attack. First, in the absence of an attack (i.e., $\text{train}_C|\text{test}_C$), the classification performance of OCSVMs (i.e., f-score) trained on nonlinearly projected data are close to the performance of the OCSVM trained on the input feature space (2% lower on MNIST, 1% and 8% higher on CIFAR-10 and SVHN). In some cases (e.g., SVHN where the images contain parts of the adjacent digits, making the data noisy) we see that the f-score can be higher in lower dimensional spaces than the input dimension OCSVM. We speculate that this occurs because a clearer separation can occur among data points from different classes when data is projected to a lower dimensional space, as shown in [12].

Under an attack on integrity. We observe that the f-scores of $\text{train}_D|\text{test}_D$ in all three datasets, across all the dimensions, are up to 1% less than the f-scores of $\text{train}_C|\text{test}_D$. Even though the difference is not large, this indicates that a OCSVM trained on clean data can identify adversarial samples better than a OCSVM trained on distorted data. Consequently this shows that OCSVMs are not immune to integrity attacks by design, and by carefully crafting adversarial data points, adversaries can increase the classification error of OCSVMs.

A comparison of the f-score in the $\text{train}_D|\text{test}_D$ case shows an increase in f-score when the proposed defense algorithm is used compared to an OCSVM in input space. The increased f-score confirms that by projecting data to a lower dimensional space using a carefully selected direction, we can identify adversarial samples that would not have been identifiable in the input space. This is further supported by the figures on the rightmost column, which show the average false negative rates of the OCSVMs under different levels of integrity attacks. We find that there is a significant improvement in detecting adversarial samples under the proposed approach compared to a OCSVM in input space (e.g., 7.25% on SVHN, 9.75% on CIFAR-10, and 24.87% on MNIST). The AUC values shown by the figures in the middle column of Fig. 5 further supports this. As reducing data to low dimensional spaces results in a loss of information, we believe that when the data dimensionality is reduced below a certain threshold, the performance would start to degrade.

Effectiveness of the compactness index. The effectiveness of the compactness index for selecting projection directions can be seen by the difference in FNRs in the figures on rightmost column in Fig. 5. Although random projection directions have resulted in higher FNRs compared to selective projection directions in most test cases, it is possible for a randomly sampled direction to be one that minimizes the adversarial distortions. But the probability

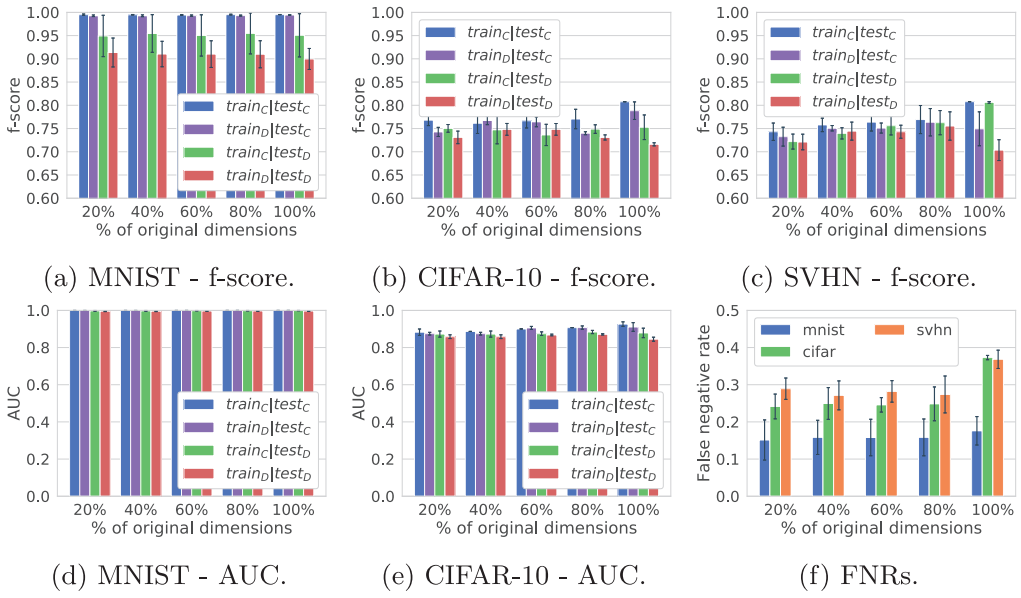


Fig. 6. The performance of binary SVMs under attacks on integrity when the training takes place in different dimensional spaces. The top row compares the f-scores of SVMs trained on train_C and train_D against the two test sets: test_C and test_D. Fig. 6d and e show the corresponding AUC values for the MNIST and CIFAR-10 datasets. Fig. 6f shows the FNR of binary SVMs under an integrity attack (i.e., trained on train_D and evaluated using test_D).

of obtaining such a direction will be low due to the large number of possible directions available for high dimensional datasets and would depend significantly on the distribution of the data clouds. An alternative approach to finding good directions would be to train an anomaly detection model on every projected dataset and test its accuracy on a validation set. But the proposed index would be able to achieve this with much less computational burden.

7.2. Binary SVM results

Absence of an attack. Fig. 6a–c present the f-score values at different dimensional spaces when the adversary distorts 20% (i.e., p_{attack}) of the training data. We show the average f-score along with the standard deviations when the attack severity varies from 0.3 to 0.6. Compared to the anomaly detection scenario using OCSVMs, the binary SVM trained on the input space with a RBF kernel outperforms the SVMs in the projected spaces when the training and testing data are clean (i.e., $\text{train}_C|\text{test}_C$) (e.g., 6.47% on SVHN, 4.64% on CIFAR-10, and less than 0.01% on MNIST).

Under an attack on integrity. Under adversarial conditions, the opposite can be observed with the defense framework giving better performance in terms of f-score (e.g., 5.19% on SVHN, 3.22% on CIFAR-10, and 1.39% on MNIST) (i.e., $\text{train}_D|\text{test}_D$). This characteristic is often found in defenses against adversarial examples where there is a trade-off between the learners' performance in the absence of attack, and their robustness under attack. Improving robustness often causes a decrease of the performance in the absence of attack. The resistance added against integrity attacks by our proposed approach is confirmed by Fig. 6f, which shows the average false negative rates of the classifiers under the different attack severities. Again, we find that there is a significant improvement in detecting adversarial samples compared to the SVM in the input space (e.g., FNR reduction of 9.70% on SVHN, 13.15% on CIFAR-10, and 2.47% on MNIST).

Fig. 6 d and e show the AUC values for MNIST and CIFAR-10 that correspond to the f-score values in Fig. 6a and b. Although the AUC values for CIFAR-10 show the effects of the adversary's attack on the performance of the SVM, the AUC values for MNIST fail to exhibit the attack's effects (where as the f-score values do). The AUC value represents the classification performance of a SVM around the separation boundary obtained from training, where as the f-score represents the performance of a SVM at the selected separation boundary. Therefore the AUC values can be considered as a supplementary result that supports the f-score and FNR results.

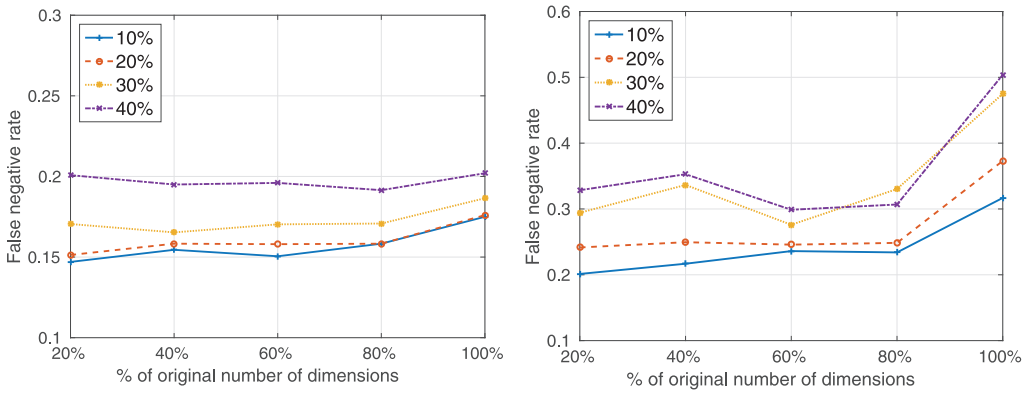
We also observe that for CIFAR and SVHN, when the dimension is reduced below 40% of the input dimension, the performance starts to degrade. We postulate that the explanation of this effect is the reduction in distance between classes with the dimension. As we reduce the dimension of the projection, we are able to reduce the effects of the adversarial distortions. But at the same time, there is a significant loss of useful information due to the dimensionality reduction. Due to the interplay between these two factors, the performance of SVMs reduces as we decrease the dimension beyond a dataset dependent threshold.

Effect of large percentages of attack data during training. Fig. 7 depicts the FNRs of binary SVMs in different dimensional spaces when the adversary increases the percentage of distorted attack points in the training set from 10% to 40%. As expected, we see that when p_{attack}) increases, the FNRs also increase. It should be noted that while an adversary theoretically has the ability to arbitrarily increase the number of attack points in the training set, in real world applications, greedily increasing the attack percentage would inevitably make the attack obvious.

7.3. Outcomes of the game

Fig. 8 shows the payoff matrix of the adversary and learner when the deterministic game described in Section 4.1 is played on the MNIST dataset. By considering the best responses of both players, we obtain the Nash equilibrium solution to the game, which is $s_{attack} = 0.3$ and 20% of the original number of dimensions. We observe that these strategies are in fact the dominant strategies for both players. In a dominant strategy, each player's best strategy is unaffected by the actions of the other player, which gives a stronger outcome than the Nash equilibrium. Based on this result, we conclude that it is in the best interest of the learner to always project data to 20% of the original number of dimensions.

Similarly, Fig. 9 shows the payoff matrix of the adversary and learner for the Omnet simulation dataset. As per Proposition 3.1 by



(a) MNIST. (b) CIFAR-10.

Fig. 7. Comparison of the FNRs of binary SVMs when the adversary changes the percentage of distorted attack points in the training set (i.e., p_{attack}) from 10% to 40%.

		% of the original # of dimensions				
		20%	40%	60%	80%	100%
Attack severity	0	(1.50,0.867)	(1.63,0.848)	(1.66,0.845)	(1.37,0.854)	(1.40,0.859)
	0.3	**(1.62,0.838)**	(1.72,0.822)	(1.75,0.821)	(1.51,0.825)	(1.85,0.802)
	0.4	(1.48,0.847)	(1.59,0.829)	(1.63,0.827)	(1.36,0.837)	(1.80,0.802)
	0.5	(1.36,0.856)	(1.48,0.838)	(1.53,0.832)	(1.23,0.847)	(1.73,0.804)
	0.6	(1.27,0.862)	(1.39,0.843)	(1.42,0.838)	(1.14,0.855)	(1.58,0.812)

Fig. 8. The utility matrix of the game played on MNIST, depicting the outcomes. The adversary is the row player and the learner is the column player and payoffs are displayed as (adversary utility, learner utility). The highlighted cell is the Nash equilibrium solution.

		% of the original # of dimensions				
		20%	40%	60%	80%	100%
Attack severity	0	(1.00,0.947)	(1.00,0.945)	(1.00,0.981)	(1.00,0.908)	(1.00,0.908)
	0.3	(0.95,0.935)	(0.85,0.974)	(1.10,0.938)	(1.30,0.894)	(1.35,0.896)
	0.4	(1.10,0.932)	**(1.25,0.935)**	(1.10,0.925)	(1.25,0.894)	(1.25,0.901)
	0.5	(1.10,0.913)	(1.15,0.933)	(1.75,0.830)	(1.75,0.844)	(1.75,0.844)

Fig. 9. The utility matrix of the game played on simulation data, depicting the outcomes. The adversary is the row player and the learner is the column player and payoffs are displayed as (adversary utility, learner utility). The highlighted cell is the Nash equilibrium solution.

Basar and Olsder [34], any deterministic bimatrix game that uses a positive affine transformation of the utility functions described in Section 4.1 used in this game would also have the same Nash equilibrium strategies for both players.

The Nash equilibrium solution to the game in this scenario is $s_{attack} = 0.4$ and 40% of the original number of dimensions. If the learner deviates from the Nash equilibrium solution unilaterally, there can be a reduction to its utility value up to 0.04. Similarly, if the attacker deviates from the Nash equilibrium solution unilaterally, the reduction to its utility would be up to 0.4. Therefore it is in the best interest of the learner to always project data to 40% of the original number of dimensions, and the attacker to use an attack severity of 0.4 in this particular problem. The average f-score values and FNRs are shown in Fig. 5g and i for the different dimensions. The results show that the proposed framework can be successfully utilized in a practical application scenario such as this.

7.4. Comparison of defense framework

We compare the effectiveness of the proposed framework against related attack strategies and defense algorithms in the literature. First, we evaluate the performance of the algorithms when there is no attack present (NA). Then, we compare the performance against the *Restrained Attack (RA)* introduced by Zhou et al. [9], *Poisoning Attack (PA)* by Biggio et al. [7] and the *Coordinate Greedy (CG)* attack proposed by Li et al.[10]. We also compare the performance against the online Centroid Anomaly Detection (CAD) approach proposed by Kloft and Laskov [24] with the nearest-out replacement policy, LS-SVM proposed by Suykens and Vandewalle [23] and a Logistic Regression (LR) learner. The training and test datasets are generated in the same way as in our previous experiments. The regularization parameter C of LR was selected by performing a grid search on a clean data set.

The optimal strategy for the learner (i.e., 20% of the original number of dimensions) is selected using the game outcome in Section 7.3 and is used for all three datasets when the defense framework is used. Table 3 gives the FNR of each defense mechanism averaged over five cross validation sets. We see that the SVMs trained with the defense framework (SVM Fw and OCSVM Fw) have an increased ability to accurately detect adversarial samples during test time. For classification, the SVM with the framework (SVM Fw) has lower FNRs compared to the other learners in all test cases except for CIFAR-10 under PA and CG. Under PA, LS-SVM has a 1.9% lower FNR and under CG, LS-SVM has a 2.3% lower FNR. For anomaly detection, the OCSVM with the framework (OCSVM Fw) outperforms the other learners with consistently lower FNRs on all test cases.

We also observe that in the absence of an attack (i.e., NA), the performance of the SVMs with the defense framework are relatively less compared to the other learners. For classification, the SVM with no defense has a 0.6% lower FNR on CIFAR-10 and LS-SVM has a 0.7% lower FNR on SVHN. For anomaly detection, CAD has lower FNRs of 7.91% and 1.2% compared to OCSVMs with the defense framework when tested on MNIST and CIFAR-10. On SVHN however, the OCSVM with the framework has a 0.4% lower FNR. Note that the decrease in performance is relatively less compared to the advantage gained in the presence of an attack. This may be due to the fact that the RKS algorithm [11] only approximates the actual kernel matrix in a low dimensional space, therefore it is likely that some useful information is lost during the transformation. From this extensive comparison, we see that the defense framework is able to consistently reduce the FNRs across different attack strategies compared to other learners.

In summary, the above experiments demonstrate that (i) OCSVMs and SVMs are vulnerable to adversarial *attacks on integrity*, (ii) the performance in the projected spaces, when there are no attacks, is comparable to that in the input space, but with

Table 3
A comparison of the discrimination power (FNR) of the defense framework against different learners and attacks. The best results are highlighted in bold and columns corresponding to the proposed defense framework are highlighted in grey.

	Dataset	SVM	SVM Fw	LS-SVM	LR	OCSVM	OCSVM Fw	CAD
NA	MNIST	0.003	**0.002**	0.005	0.003	0.097	0.079	**0.000**
	CIFAR	**0.194**	0.200	0.198	0.200	0.210	0.212	**0.200**
	SVHN	0.210	0.209	**0.202**	0.204	0.290	**0.281**	0.285
RA	MNIST	0.122	**0.077**	0.084	0.146	0.694	**0.653**	1.000
	CIFAR	0.372	**0.237**	0.346	0.369	0.970	**0.835**	1.000
	SVHN	0.352	**0.234**	0.326	0.371	0.900	**0.790**	0.980
PA	MNIST	0.294	**0.203**	0.216	0.284	0.767	**0.675**	0.730
	CIFAR	0.364	0.223	**0.204**	0.361	0.814	**0.723**	0.863
	SVHN	0.384	**0.287**	0.321	0.410	0.863	**0.750**	0.886
CG	MNIST	0.417	**0.393**	0.396	0.426	0.797	**0.767**	0.818
	CIFAR	0.397	0.364	**0.341**	0.386	0.862	**0.784**	0.842
	SVHN	0.386	**0.298**	0.321	0.391	0.912	**0.845**	0.924

less computational burden, and most importantly, (iii) by projecting a distorted dataset to a lower dimension in an appropriate direction we can increase the robustness of SVMs w.r.t. integrity attacks.

8. Conclusion

This paper presents a theoretical and experimental investigation on the effects of integrity attacks that poison the training data and affect the learners in the course of training. We introduce a unique framework that combines nonlinear data projections using novel ranking indices that we introduce, together with SVMs and game theory. Through the network simulation we show that the flexibility of the proposed framework allows it to be applied to real world security applications. Our numerical analysis focuses on the performance of the proposed defense framework under adversarial conditions. The results suggest that SVMs and OCSVMs can be significantly affected if an adversary can manipulate the data on which they are trained. For each dataset, with very high probability, there is at least one dimensionality and projection direction that can accurately identify adversarial samples that would have been missed by a SVM or a OCSVM in the input space. Therefore, our approach can be utilized to make SVM based learning systems secure by (i) reducing the impact of possible adversarial distortions by contracting and separating data points from different classes in the projected space, and (ii) making it challenging for an adversary to guess the underlying details of the learner by making its search space unbounded through a layer of randomness.

Acknowledgement

This work was supported in part by the Australian Research Council Discovery Project under Grant DP140100819, and by Northrop Grumman Mission Systems' University Research Program. The authors thank Prof. Margreta Kuijper for the helpful comments and discussions.

Appendix A. Proofs

A1. Proof of theorem 1

Proof: Let $\tilde{\alpha}$ be the vector achieving the optimal solution in the projected space when adversarial distortions are present. Then, the solution for the primal problem in the projected space with adversarial distortions, defined as $\|w_{pt}^*\|_2$, can be obtained as

$$\left\|w_{pt}^*\right\|_2 = \left\|\tilde{\alpha}^T C\right\|_2. \tag{A.1}$$

Using the cosine angle-sum identity on the matrix defined by Eq. 7 (the symbol $\odot$ denotes the Hadamard product for matrices),

$$\left\|w_{pt}^*\right\|_2 = \left\|\tilde{\alpha}^T\left(C^X \odot C^T\right) - \tilde{\alpha}^T\left(S^X \odot S^T\right)\right\|_2. \tag{A.2}$$

Using the reverse triangle inequality we obtain

$$\left\|w_{pt}^*\right\|_2 \geq \left\|\tilde{\alpha}^T\left(C^X \odot C^T\right)\right\|_2 - \left\|\tilde{\alpha}^T\left(S^X \odot S^T\right)\right\|_2. \tag{A.3}$$

From the constraint conditions of the OCSVM problem (8), we get $\mathbf{1}^T\tilde{\alpha} = 1$. Also, as $\sin(\theta) \in [-1, 1]$ the inequality can be further simplified as,

$$\left\|w_{pt}^*\right\|_2 \geq \left\|\tilde{\alpha}^T\left(C^X \odot C^T\right)\right\|_2 - \sqrt{r}. \tag{A.4}$$

Due to *Assumption 1*, using small-angle approximation on C^T, we obtain

$$\left\|w_{pt}^*\right\|_2 \geq \left\|\tilde{\alpha}^T\left(C^X \odot \left(1 - \frac{TA \odot TA}{2}\right)\right)\right\|_2 - \sqrt{r}. \tag{A.5}$$

Applying the reverse triangle inequality to the first term on the right hand side, we obtain

$$\left\|w_{pt}^*\right\|_2 \geq \left\|\tilde{\alpha}^T C^X\right\|_2 - \left\|\tilde{\alpha}^T\left(C^X \odot \left(\frac{TA \odot TA}{2}\right)\right)\right\|_2 - \sqrt{r}. \tag{A.6}$$

As the training data is normalized between $(0 \; -- \; 1)$, the maximum distortion magnitude that can be achieved is 1. Also, for any θ, $\cos(\theta) \leq 1$. Therefore, $C^X \odot (\frac{TA \odot TA}{2})$ on the right hand side can be replaced by a $n \times r$ matrix where each entry is $\frac{1}{2}$. Also, $\mathbf{1}^T\tilde{\alpha} = 1$, the inequality can be further simplified as,

$$\left\|w_{pt}^*\right\|_2 \geq \left\|\tilde{\alpha}^T C^X\right\|_2 - \frac{\sqrt{r}}{2} - \sqrt{r}. \tag{A.7}$$

Since the optimization problem is a minimization problem, as shown in (8), the optimal solution for the OCSVM without any distortion (i.e., α^*) would give a value less than or equal to the value given by $\tilde{\alpha}$. Thus,

$$\begin{aligned} \left\|\alpha^{*,T} C^X\right\|_2 &\leq \left\|w_{pt}^*\right\|_2 + \tfrac{3\sqrt{r}}{2}, \\ \left\|w_p^*\right\|_2 &\leq \left\|w_{pt}^*\right\|_2 + \tfrac{3\sqrt{r}}{2}. \end{aligned} \tag{A.8}$$

A2. Proof of theorem 3

Proof: Let $\tilde{\alpha}$ be the vector achieving the optimal solution in the projected space when adversarial distortions are present. Then, the optimization problem of the OCSVM in the projected space with adversarial distortions is given by

$$\begin{aligned} &\underset{\alpha}{\text{minimize}} \; \frac{1}{2}\|\alpha^T(X+T)A\|_2^2, \\ &\text{subject to } 0 \leq \alpha \leq \tfrac{1}{\nu n} \text{ and } \mathbf{1}^T\alpha = 1. \end{aligned} \tag{A.9}$$

As any dataset can be aligned in a manner that compels T to be positive and because $X \in [0, 1]$, using the reverse triangle inequality we obtain,

$$|(1-\delta)|\|\alpha^T XA\|_2 \leq \|\alpha^T(X+T)A\|_2. \tag{A.10}$$

Let $Z_{pt}(\tilde{\alpha})$ be the optimal value of (A.9). Define next

$$Z_p(\tilde{\alpha}) := \|\tilde{\alpha}^T XA\|_2^2 \tag{A.11}$$

which is the optimal value without any malicious distortion ($T = 0$). Then, it follows from (A.10),

$$Z_p(\tilde{\alpha}) \leq \frac{Z_{pt}(\tilde{\alpha})}{|(1-\delta)|^2}. \tag{A.12}$$

Let $Z(\alpha^*)$ denote the optimal value for the optimization problem in the input space. Although $\tilde{\alpha}$ is feasible, it is not an optimal solution. Therefore, by definition, $Z(\alpha^*) \leq Z(\tilde{\alpha})$.

Then, using the singular-value decomposition (SVD) of X, we get,

$$\begin{aligned} Z(\alpha^*) &= \frac{1}{2}\alpha^{*T} XX^T \alpha^*, \\ &= \frac{1}{2}\alpha^{*T} U\Sigma(V^T V)\Sigma U^T \alpha^*, \\ &= \frac{1}{2}\alpha^{*T} U\Sigma(E + V^T AA^T V)\Sigma U^T \alpha^*, \\ &= \frac{1}{2}\alpha^{*T} U\Sigma V^T AA^T V\Sigma U^T \alpha^* + \tfrac{1}{2}\alpha^{*T} U\Sigma E\Sigma U^T \alpha^*, \end{aligned} \tag{A.13}$$

where α^* is the vector achieving the optimal solution in the input feature space without adversarial distortions and

$$\begin{aligned} Z_p(\tilde{\alpha}) &= \frac{1}{2}\tilde{\alpha}^T XAA^T X^T \tilde{\alpha}, \\ &= \frac{1}{2}\tilde{\alpha}^T U\Sigma V^T AA^T V\Sigma U^T \tilde{\alpha}. \end{aligned} \tag{A.14}$$

By substituting (A.14) into (A.13),

$$\begin{aligned} Z(\alpha^*) &= \frac{1}{2}\tilde{\alpha}^T U\Sigma V^T AA^T V\Sigma U^T \alpha^* + \tfrac{1}{2}\tilde{\alpha}^T U\Sigma E\Sigma U^T \tilde{\alpha} \\ &\leq \frac{1}{2}\tilde{\alpha}^T U\Sigma V^T AA^T V\Sigma U^T \tilde{\alpha} + \tfrac{1}{2}\tilde{\alpha}^T U\Sigma E\Sigma U^T \tilde{\alpha} \\ &= Z_p(\tilde{\alpha}) + \tfrac{1}{2}\tilde{\alpha}^T U\Sigma E\Sigma U^T \tilde{\alpha}. \end{aligned} \tag{A.15}$$

The second term of the above equation can be further analyzed by taking $Q = \tilde{\alpha}^T U\Sigma$ (note that $V^T V = I$)

$$\begin{aligned} \frac{1}{2}\tilde{\alpha}^T U\Sigma E\Sigma U^T \tilde{\alpha} &\leq \tfrac{1}{2}\|Q\|_2 \|E\|_2 \|Q\|_2, \\ &= \frac{1}{2}\|E\|_2 \|Q\|_2^2, \\ &= \frac{1}{2}\|E\|_2 \|\tilde{\alpha}^T U\Sigma V^T\|_2^2, \\ &= \frac{1}{2}\|E\|_2 \|\tilde{\alpha}^T X\|_2^2. \end{aligned} \tag{A.16}$$

Using the above result, (A.15) can be written as

$$Z(\alpha^*) \leq Z_p(\tilde{\alpha}) + \frac{1}{2}\|E\|_2 \|\tilde{\alpha}^T X\|_2^2. \tag{A.17}$$

The following steps result in a bound for the second term of the above equation.

$$\begin{aligned} &|\tilde{\alpha}^T XX^T \tilde{\alpha} - \tilde{\alpha}^T XAA^T X^T \tilde{\alpha}| \\ =&|\tilde{\alpha}^T U\Sigma VV^T \Sigma U\tilde{\alpha} - \tilde{\alpha}^T U\Sigma VAA^T V^T \Sigma U\tilde{\alpha}|, \\ =&|(\tilde{\alpha}^T U\Sigma(VV^T - VAA^T V^T)\Sigma U\tilde{\alpha})|, \\ =&|\tilde{\alpha}^T U\Sigma E\Sigma U\tilde{\alpha}|, \\ \leq&\|E\|_2 \|\tilde{\alpha}^T U\Sigma V\|_2^2, \\ =&\|E\|_2 \|\tilde{\alpha}^T X\|_2^2. \end{aligned} \tag{A.18}$$

The above inequality can be rewritten as

$$\left| \|\tilde{\alpha}^T X\|_2^2 - \|\tilde{\alpha}^T XA\|_2^2 \right| \leq \|E\|_2 \|\tilde{\alpha}^T X\|_2^2. \tag{A.19}$$

Thus, as shown in [36], using (A.11)

$$\begin{aligned} \|\tilde{\alpha}^T X\|_2^2 &\leq \frac{1}{1-\|E\|_2}\|\tilde{\alpha}^T XA\|_2^2 \\ &\leq \frac{1}{1-\|E\|_2} Z_p(\tilde{\alpha}) \end{aligned} \tag{A.20}$$

From (A.12) and (A.17), by taking $\lambda = \frac{1}{2}\frac{\|E\|_2}{(1-\|E\|_2)}$,

$$\begin{aligned} Z(\alpha^*) &\leq Z_p(\tilde{\alpha}) + \tfrac{1}{2}\frac{\|E\|_2}{(1-\|E\|_2)} Z_p(\tilde{\alpha}) \\ &= (1+\lambda) Z_p(\tilde{\alpha}) \\ &= (1+\lambda)\tfrac{Z_{pt}(\tilde{\alpha})}{|(1-\delta)|^2} \end{aligned} \tag{A.21}$$

By definition, $w^* = \alpha^{*T} X$ and $w_{pt}^* = \tilde{\alpha}^T X$. Therefore, $Z(\alpha^*) = \frac{1}{2}\|w^*\|_2^2$, $Z_{pt} = \frac{1}{2}\|w_{pt}^*\|_2^2$.

$$\|w^*\|_2^2 \leq \frac{(1+\lambda)}{|(1-\delta)|^2}\|w_{pt}^*\|_2^2. \tag{A.22}$$

References

[1] C. Cortes, V. Vapnik, Support-Vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[2] Y. Ma, G. Guo, Support vector machines applications, Springer, 2014.
[3] M.M. Krell, H. Wöhrle, New one-class classifiers based on the origin separation approach, Pattern Recognit. Lett. 53 (2015) 93–99.
[4] H. Xu, C. Caramanis, S. Mannor, Robustness and regularization of support vector machines, J. Mach. Learn. Res. 10 (Jul) (2009) 1485–1510.
[5] L. Huang, A.D. Joseph, B. Nelson, B.I.P. Rubinstein, J.D. Tygar, Adversarial Machine Learning, in: Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, 2011, pp. 43–58.
[6] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 387–402.
[7] B. Biggio, B. Nelson, P. Laskov, Poisoning Attacks Against Support Vector Machines, in: Proceedings of the 29th International Coference on International Conference on Machine Learning, 2012, pp. 1467–1474.
[8] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, et al., Adversarial classification, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2004, pp. 99–108.
[9] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, B. Xi, Adversarial Support Vector Machine Learning, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1059–1067.
[10] B. Li, Y. Vorobeychik, X. Chen, A general retraining framework for scalable adversarial classification, arXiv preprint arXiv:1604.02606 (2016).
[11] A. Rahimi, B. Recht, Random Features for Large-Scale Kernel Machines, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2008, pp. 1177–1184.
[12] S.M. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, C. Leckie, R1SVM: A Randomised Nonlinear Approach to Large-Scale Anomaly Detection, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 432–438.
[13] T. Alpcan, T. Basar, Network security: a decision and game-theoretic approach, 1st, 2010.
[14] T. Alpcan, B.I.P. Rubinstein, C. Leckie, Large-scale strategic games and adversarial machine learning, in: Proceedings of the IEEE Conference on Decision and Control (CDC), 2016, pp. 4420–4426.
[15] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybernet.Part B (Cybernetics) 28 (3) (1998) 301–315.
[16] P.S. Weerasinghe, S.M. Erfani, T. Alpcan, C. Leckie, M. Kuijper, Unsupervised Adversarial Anomaly Detection Using One-Class Support Vector Machines, in: 23rd International Symposium on Mathematical Theory of Networks and Systems, 2018, pp. 34–37.
[17] S. Weerasinghe, S.M. Erfani, T. Alpcan, C. Leckie, J. Riddle, Detection of anomalous communications with SDRs and unsupervised adversarial learning, in: 2018 IEEE 43rd Conference on Local Computer Networks (LCN), IEEE, 2018.
[18] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognit. 58 (2016) 121–134.
[19] B. Biggio, I. Corona, B. Nelson, B.I.P. Rubinstein, D. Maiorca, G. Fumera, G. Giacinto, F. Roli, Security Evaluation of Support Vector Machines in Adversarial Environments, in: Support Vector Machines Applications, Springer, 2014, pp. 105–153.

[20] B. Biggio, B. Nelson, P. Laskov, Support vector machines under adversarial label noise, in: Asian Conference on Machine Learning, 2011, pp. 97–112.
[21] H. Xiao, H. Xiao, C. Eckert, Adversarial Label Flips Attack on Support Vector Machines, in: Proceedings of the 20th European Conference on Artificial Intelligence, 2012, pp. 870–875.
[22] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, F. Roli, Support vector machines under adversarial label contamination, Neurocomputing 160 (2015) 53–62.
[23] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neur. Process. Lett. 9 (3) (1999) 293–300.
[24] M. Kloft, P. Laskov, Security analysis of online centroid anomaly detection, J. Mach. Learn. Res. 13 (2012) 3681–3724.
[25] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[26] S. Rajasegarar, C. Leckie, M. Palaniswami, Pattern based anomalous user detection in cognitive radio networks, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19–24, 2015, 2015, pp. 5605–5609.
[27] J. Steinhardt, P.W.W. Koh, P.S. Liang, Certified defenses for data poisoning attacks, in: Advances In Neural Information Processing Systems, 2017, pp. 3517–3529.
[28] R. Laishram, V.V. Phoha, Curie: a method for protecting SVM classifier from poisoning attack, arXiv preprint arXiv:1606.01584 (2016).
[29] N.X. Vinh, S. Erfani, S. Paisitkriangkrai, J. Bailey, C. Leckie, K. Ramamohanarao, Training robust models using random projection, in: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE, 2016, pp. 531–536.
[30] E. Wong, F. Schmidt, J.H. Metzen, J.Z. Kolter, Scaling provable adversarial defenses, in: Advances in Neural Information Processing Systems, 2018, pp. 8400–8409.
[31] W.B. Johnson, J. Lindenstrauss, Extensions of Lipschitz mappings into a hilbert space, Contemp. Math. 26 (189–206) (1984) 1.
[32] P.B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, J.C. Platt, Support Vector Method for Novelty Detection, in: Proceedings of Advances in Neural Information Processing Systems (NIPS), 2000, pp. 582–588.
[33] J.F. Nash, Non-cooperative games, Ann. Math. 54 (1951) 286–295.
[34] T. Basar, G.J. Olsder, Dynamic noncooperative game theory, volume 23, Siam, 1999.
[35] B. Schölkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neur. Comput. 12 (5) (2000) 1207–1245.
[36] S. Paul, C. Boutsidis, M. Magdon-Ismail, P. Drineas, Random projections for linear support vector machines, ACM Trans. Knowl. Discover. Data (TKDD) 8 (4) (2014) 22:1–22:25.
[37] A. Varga, R. Hornig, An Overview of the OMNeT++ Simulation Environment, in: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, 2008, pp. 60:1–60:10.

Sandamal Weerasinghe is currently working toward the Ph.D. degree with the Department of Electrical and Electronic Engineering, University of Melbourne, Australia. His research interests include adversarial machine learning, anomaly detection, and game theory.

Sarah M. Erfani is a lecturer in the School of Computing and Information Systems at the University of Melbourne. Her research interests include large-scale data mining, machine learning, wireless sensor networks, and privacy-preserving data mining.

Tansu Alpcan received the Ph.D. degree in Electrical and Computer Engineering from University of Illinois at Urbana-Champaign (UIUC) in 2006. His research interests include applications of control, optimisation, and game theories, and machine learning to security and resource allocation problems in communications, smart grid, and Internet-of-Things. He chaired or was an Associate Editor, TPC chair, or TPC member of several prestigious IEEE workshops, conferences, and journals. Tansu Alpcan is the (co-)author of more than 150 journal and conference articles as well as the book "Network Security: A Decision and Game Theoretic Approach" published by Cambridge University Press (CUP) in 2011. He co-edited the book "Mechanisms and Games for Dynamic Spectrum Allocation" published by CUP in 2014. He has worked as a senior research scientist in Deutsche Telekom Laboratories, Berlin, Germany (2006–2009), and as Assistant Professor (Juniorprofessur) in Technical University Berlin (2009–2011). He is currently with the Dept. of Electrical and Electronic Engineering at The University of Melbourne as an Associate Professor and Reader.

Christopher Leckie is a Professor with the School of Computing and Information Systems at the University of Melbourne. His research interests include using artificial intelligence for network management and intrusion detection, and data mining techniques such as clustering and anomaly detection.