Detection of Anomalous Communications with SDRs and Unsupervised Adversarial Learning

Sandamal Weerasinghe*, Sarah M. Erfani†, Tansu Alpcan*, Christopher Leckie† and Jack Riddle‡
*Dept. of Electrical and Electronic Engineering
The University of Melbourne, VIC 3010, Australia
†School of Computing and Information Systems
The University of Melbourne, VIC 3010, Australia
‡Northrop Grumman Corporation, USA
p.weerasinghe@student.unimelb.edu.au;{sarah.erfani,tansu.alpcan,caleckie}@unimelb.edu.au;jack.riddle@ngc.com

***Abstract*—Software-defined radios (SDRs) with substantial cognitive (computing) and networking capabilities provide an opportunity for observing radio communications in an area and potentially identifying malicious rogue agents. Assuming a prevalence of encryption methods, a cognitive network of such SDRs can be used as a low-cost and flexible scanner/sensor array for distributed detection of anomalous communications by focusing on their statistical characteristics. Identifying rogue agents based on their wireless communications patterns is not a trivial task, especially when they deliberately try to mask their activities. We address this problem using a novel framework that utilizes adversarial learning, non-linear data transformations to minimize the rogue agent's attempts at masking their activities, and game theory to predict the behavior of rogue agents and take the necessary countermeasures. Adopting One-Class Support Vector Machines (OCSVMs) as an unsupervised learning method, we show that under adversarial conditions, selective nonlinear random projections can be leveraged to increase the attack resistance of OCSVMs. Experiments with benchmark data sets and OMNET++ simulations of specific communications scenarios illustrate the benefits of our framework numerically.**

I. Introduction

In a given populated area (e.g., a city block in an urban area), a multitude of individuals communicate with each other using various communication methods. Individuals use a plethora of devices for communication purposes, and such devices may utilize infrastructure provided by third parties such as mobile network providers as well as peer-to-peer communications. 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. The communications between individuals is 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.

Recent advances in software-defined radios (SDRs) and cognitive networking technologies provide an opportunity for identifying rogue agents by observing radio communications in an area. A cognitive network of software-defined radios can be used as a low-cost and flexible scanner/sensor array for distributed observation of the radio spectrum, focusing on statistical characteristics of the communication patterns. The cheap and small-sized SDRs such as RTL-SDR mini receivers [12] can be deployed in UAVs, vehicles, and even on individuals, as part of an intelligent defense network, covering a broad range of the radio spectrum from MHz to GHz frequencies (Figure 1). The distributed nature of the proposed SDR network is necessary for identifying potential rogue agents for further investigation, especially taking into account the often peer-to-peer nature of such communications.

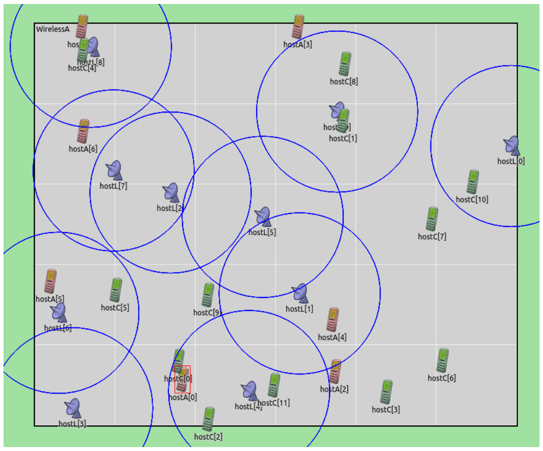


Fig. 1: A representation of the SDR listeners (blue), civilians (green) and rogue agents (red) in OMNET++.

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 rouge agents differ from those of regular background traffic to some degree, otherwise, the detection problem would be infeasible. We propose to address this problem using a unique anomaly detection framework that utilizes non-linear data transformations to minimize the rogue agent's attempts to mask their activities, unsupervised adversarial learning for classification, and game theory to predict the behavior of rogue agents and take the necessary countermeasures. Unsupervised learning discovers patterns in data and identifies data points that do not conform to the learned patterns, i.e., anomalies/outliers. 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). Many machine learning methods, such as One-Class Support Vector Machines (OCSVM) [17], have been proven to be effective in anomaly detection applications. Although they are designed to withstand the effects of random noise in data, their performance may degrade significantly when adversaries deliberately alter the input data.

By distorting the input data used by a learning algorithm, the adversaries can force a learner to learn a model that favors the adversary. In the above scenario, if the rogue agents alter their communication patterns in a targeted manner 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. A sophisticated adversary has the capacity to conduct an attack in numerous ways [10]. Therefore, it is not feasible to provide a general analysis that covers the whole range of attacks, across different machine learning algorithms. In this work, we focus on *integrity attacks*, where the adversary deliberately poisons the training data used by the learner in order to make them learn a compromised decision boundary that exaggerates the region where normal data points (civilians) lie. Subsequently, during the evaluation phase (i.e., when the sensor network is used to identify rogue agents), the rogue agents would be able to make the learner classify them as civilians by making minor alterations to their regular communication patterns.

The main components of the framework we introduce are as follows. The learner leverages the theory of low rank kernel approximation (using non-linear projections) which facilitates large-scale, data-oriented, multi-agent decisions by reducing the number of optimization parameters and variables. Recent work in the literature shows that nonlinear random projections improve the training and evaluation times of kernel machines, without significantly compromising the accuracy of the trained models [7], [15]. In this paper, we show that under adversarial conditions, selective nonlinear random projections can be leveraged to increase the attack resistance of unsupervised classifiers (e.g., OCSVMs) as well. In addition, we design a security game [1] and model the adversary-learner interaction as a non-cooperative, two-player, nonzero-sum game with the strategies and utility functions formulated around a nonlinear data projection based algorithm that reduces the effects of integrity attacks against OCSVMs. The equilibrium solutions obtained from the game can be used to predict the adversary's behavior and decide a suitable configuration for the learner [2].

The main **contributions** of this work are summarized as follows:

1) We introduce a unique framework that uses unsupervised learning, low rank kernel approximation in selective directions and game theory that can be applied in situations where adversaries try to evade learning systems by poisoning the training data used by the learners.
2) As part of this framework, we introduce a novel index to identify suitable directions for nonlinear data transformations and study the resistance added by such transformations against an adversarial opponent through numerical experiments on several benchmark datasets.
3) We pose the problem of finding an appropriate defence mechanism as a game and find the Nash equilibrium solution that gives 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 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 transformation, making it very difficult for the adversary to guess the projection mechanism used by the learner.

II. Background and Related Work

As our proposed approach on adversarial learning for anomaly detection is based on randomized kernels, in this section we briefly review these two lines of research.

A. Randomized Kernels for SVMs

To improve the efficiency of kernel machines, [15] embedded a random projection into the kernel formulation. They introduced a novel, data independent method (Random Kitchen Sinks (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 $z : \mathbb{R}^d \to \mathbb{R}^r$. Subsequently, [13] introduced a transformation method that has lower time and space complexities compared to RKS.

More recently, the method of [15] has been applied to other types of kernel machines. [7] 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. Our work differs from these as we look at random projections as a defense mechanism for OCSVMs under adversarial conditions. However, to the best of our knowledge, no existing work adopts Rahimi and Recht's method to address adversarial learning for anomaly detection with OCSVMs.

B. Learning under adversarial conditions

The problem of adversarial learning has inspired a wide range of research from the machine learning community, see [3] for a survey. For example, [19] introduced an Adversarial SVM (AD-SVM) model. AD-SVM incorporated additional constraint conditions to the binary SVM optimization problem in order to thwart an adversary's attacks. Their model leads to unsatisfactory results when the severity of real attacks differs from the expected attack severity by the model. While we gain valuable insights regarding attack strategies from this work, the defense mechanism in our work is significantly different. Furthermore, our work primarily focuses on unsupervised learning, whereas [19] uses a binary SVM. [5] introduced a poisoning attack algorithm that finds the optimal attack point by maximizing the hinge loss of a binary SVM when tested on a validation set. They assume that the adversary is aware of the learning algorithm and knows the training data used by the user.

In an online setting, [11] analyzed the effects of adversarial injections on centroid anomaly detection. The centroid anomaly detection algorithm can be considered as a hard margin OCSVM. 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, [16] 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 make an incorrect decision. While their work is in the same application domain, the methodology and the attack type differ from this work.

Deep Neural Networks (DNNs) have been shown to be robust to noise in the input [8], but are unable to withstand carefully crafted adversarial data [9]. While these works are in the same domain, they are not directly related to our work, which uses OCSVMs and kernels.

This paper presents a novel framework comprising adversarial learning, anomaly detection using OCSVMs, randomized kernels, and game-theoretic analysis. To the best of our knowledge, no existing work has explored this unique combination.

III. Problem Definition

We consider an adversarial learning problem for anomaly detection in the presence of a malicious adversary. The adversary's ultimate goal is to smuggle specially crafted adversarial data points past the decision boundary of the learner during testing, which we identify as false negatives (i.e., anomalies classified as normal). To succeed in this, the attacker would inject malicious data during training in order to alter the decision boundary of the learner in a manner favorable to him. 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.

For example, consider the illustrative example of digit '9' as the normal class and digit '7' as the anomaly class. Intuitively, an anomaly detection algorithm would attempt 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) that are in between digit '7' and 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*. For example, as Figure 2 shows, when a digit '7' is perturbed with less severity (e.g., 0.2), it 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'.

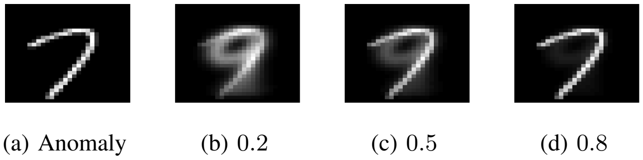


Fig. 2: A digit from anomaly class ('7') perturbed by the adversary using different s_{attack} values to appear like a digit from the normal class ('9').

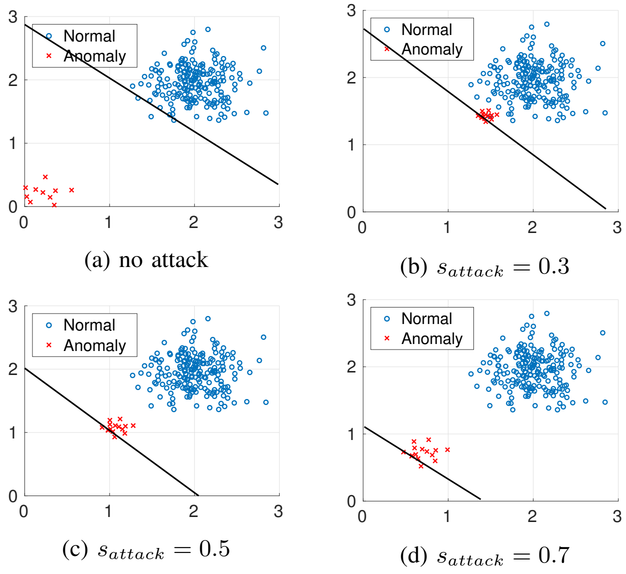
In the context of the SDR application scenario we are interested in, the rogue agents would be the adversaries who wish to make the learner classify anomalies (i.e., rogue agents) as normal data points (i.e., civilians) during the evaluation phase. To achieve this, they would first poison the training data used by the learner (i.e., change their habitual communication patterns to resemble that of civilians to some extent). Note that the rogue agents would not want to have identical communication patterns as the civilians as that would require them to use third party infrastructure as well frequent and lengthy transmissions. 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 (most anomaly detection algorithms assume that the majority of the training set contains normal data). 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.

IV. Attack Model

This section presents the method used for generating adversarial samples for benchmark datasets. In the network simulation, where we have access to the data generating system, the adversarial samples are generated by changing the parameters that define the behavior of the rogue agents.

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

The adversary is able to orchestrate different attacks by changing the percentage of distorted anomaly data points in the training dataset (i.e., p_{attack}) in addition to the severity of the distortion (i.e., s_{attack}). Figure 3 illustrates the data distributions when different levels of attack severities are applied to the anomaly data. As s_{attack} increases, the anomaly data points are moved closer to the origin, reducing the gap between the origin and the separating hyperplane.



(a) no attack

(b) $s_{attack} = 0.3$

(c) $s_{attack} = 0.5$

(d) $s_{attack} = 0.7$

Fig. 3: 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 using the entire (unlabeled) dataset as normal.

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

The attack model used is inspired by the restrained attack model described by [19] where it is assumed that the adversary has the capability to move any data point in any direction by adding a non-zero displacement vector κ_i to x_i. 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. The attacker, similar to [19], 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. For each attribute j in the original feature space, the adversary is able to add κ_{ij} to x_{ij}, where

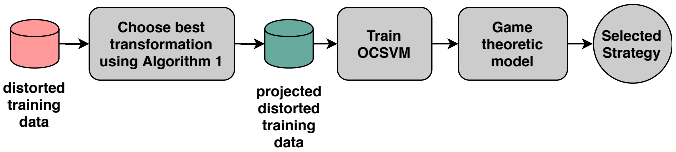

distorted training data
Choose best transformation using Algorithm 1
projected distorted training data
Train OCSVM
Game theoretic model
Selected Strategy

Fig. 4: Flowchart of the defense framework.

$$\begin{aligned} \kappa_{ij} &= (1 - s_{attack})(x_{ij}^t - x_{ij}), \\ |\kappa_{ij}| &\le |x_{ij}^t - x_{ij}|, \forall j \in d. \end{aligned} \tag{1}$$

V. Defense Framework

The novel defence framework introduced in this paper has three main components: (1) a randomized projection that increases attack resistance due to use of a novel metric, (2) an unsupervised classifier (OCSVM), and (3) a game-theoretic model that supports defensive decision-making (Figure 4).

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. The learner projects the data to a lower dimensional space using a projection matrix $A \in \mathbb{R}^{d\times r}$, comprised of elements randomly sampled from a normal distribution, i.e., $(X + D)A$. Each i^{th}sample $(X + D)_i$ is then non-linearly transformed using the function

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

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 a r-dimensional vector whose elements are drawn uniformly from $[0, 2\pi]$ [15].

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 considerate flexibility to select the dimension to which the data is projected, as well as the direction, which gives a significant advantage from a security perspective. 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 D from the training data, it is not possible to identify an ideal projection that conceals the adversarial distortions. Thus, 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 transformed space.

Therefore, motivated by a generalized version of Dunn's index [4], we propose a compactness measure to rank suitable projection directions in a one-class problem. The learner would draw multiple samples from a normal distribution for the projection matrix A and rank them using Equation 3. The projection direction A that gives the highest compactness value would be considered as the projection that gives the best attack resistance. The compactness of projection P_i, where μ_i is the centroid of the projected training set, 0 is the origin in the transformed space, and the function d is the Euclidean distance, can be calculated as

$$\text{compactness of } P_i = \frac{d(0, \mu_i)}{\big(\sum_{x \in P_i} d(x, \mu_i)\big)/n}. \tag{3}$$

The approach used by the learner to identify suitable projection directions is formalized in Algorithm 1 in terms of the random projection parameters A and b, the dimension of the projected dataset r and the adversary's data distortion strategy D.

Algorithm 1 Identifying compact projections

```
input $X + D, X_{test}$,$r$,sample count $N$
$A^{'}, b^{'} \leftarrow null$                                  ▷ transformation parameters
for $i \leftarrow 1, N$ do                                      ▷ nonlinearly transform
    $[(X + D)^{*}, A, b] \leftarrow z(X + D)$
    $compactness \leftarrow calculate\_compactness((X + D)^{*})$   ▷ calculate compactness. (Equation 3)
    if $compactness > max\_compactness$ then
        $max\_compactness \leftarrow compactness$
        $best\_transformation \leftarrow (X + D)^{*}$
        $A^{'} \leftarrow A$ and $b^{'} \leftarrow b$
    end if
end for
output $A^{'}, b^{'}$                               ▷ Return best transformation parameters
```

The next component in our framework would be the anomaly detection algorithm. The anomaly detection problem is addressed in this paper using the OCSVM algorithm in [17], which separates the training data from the origin with a maximal margin in the transformed space. Since the above transformation approximates the RBF kernel in the lower dimensional space, a linear kernel can be used on the transformed data.

A. Game Formulation

In the final component of our framework, we pose the aforementioned problem as a bimatrix game due to the innate information asymmetry present. By formulating a game based on the adversary-learner interaction, the learner can (i) predict the possible actions of the adversary, and (ii) decide what actions to take in order to thwart the adversary's attempts.

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 it is able to simulate attacks during the training process. Since the adversary can vary the severity of the attacks by changing their communication parameters during training, 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 during training, we consider s_{attack} to be 0. If the rogue agents closely mimic the civilians during training (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. Therefore we choose different s_{attack} levels (keeping p_{attack} constant) as the finite set of actions 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 available for the learner.

$$\begin{aligned} x_A &\in \{0, 0.3, 0.4, 0.5\}, \\ x_L &\in \{20\%, 40\%, 60\%, 80\%, 100\%\}. \end{aligned} \tag{4}$$

A bimatrix game is comprised of two $(m \times n)$ matrices, $A = \{a_{i,j}\}$ and $B = \{b_{i,j}\}$ where each pair of entries $(a_{i,j}, b_{i,j})$ 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' utility functions, U_A and U_L. A pair of strategies $(a_{i^*,j^*}, b_{i^*,j^*})$ 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 [14].

Due to the adversary's ability to evaluate the model during testing (i.e., calculating the false negative rate (FNR)), we design U_A to reflect his desire to achieve false negatives and to penalize large adversarial perturbations. This is because if the adversary greedily perturbs 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 his desire to achieve high classification accuracies. Note that a linear transformation of either of the utility functions would not change the outcome of the game, therefore the scalar values in the following can be modified without affecting the overall outcome. The utility functions of the two players are defined as

$$\begin{aligned} U_A(x_A, x_L) &= 1 + FNR - \frac{1}{2} s_{attack}, \\ U_L(x_A, x_L) &= f\text{-}score. \end{aligned} \tag{5}$$

VI. Network Simulation

Simulations are preformed using the INET framework for OMNeT++ (datasets available at [18]). 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 simulation is conducted for 4 hours, with the civilians and rogue agents shifting their positions every hour. Refer to Figure 1 for a simplified representation of how the different individuals are placed in the simulation environment.

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. The following features are obtained using the data captured by the listeners,

- Duration of reception
- Message length
- Inter arrival time (IAT)
- Carrier frequency
- Bandwidth
- Bitrate

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 perturbed (i.e., they deliberately changed their communication patterns to deceive the learner).

We choose 20%, 40%, 60% and 80% of the input dimension as the dimensions to which the datasets are transformed. By doing a grid search for parameters, 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. For statistical significance, we conduct the experiments multiple times using different seeds for the random number generator. The average results for the f-score and FNR are shown in Figure 5d and 7d for the different dimensions.

VII. Numerical Experiments on Benchmark Datasets

A. Benchmark datasets

We demonstrate the effectiveness of our proposed defense mechanism on three benchmark datasets: MNIST, CIFAR-10, and SVHN. MNIST contains 28x28 pixel images of handwritten digits resulting in a feature vector of length 784 for each data point. The SVHN and CIFAR-10 datasets consist of 32x32 pixel color images resulting in a feature vector of length 3072 for each data point. We compare the performance of OCSVMs and nonlinear random projections, when an active adversary is conducting a directed attack by maliciously distorting the data.

TABLE I: Datasets used for training and testing purposes.

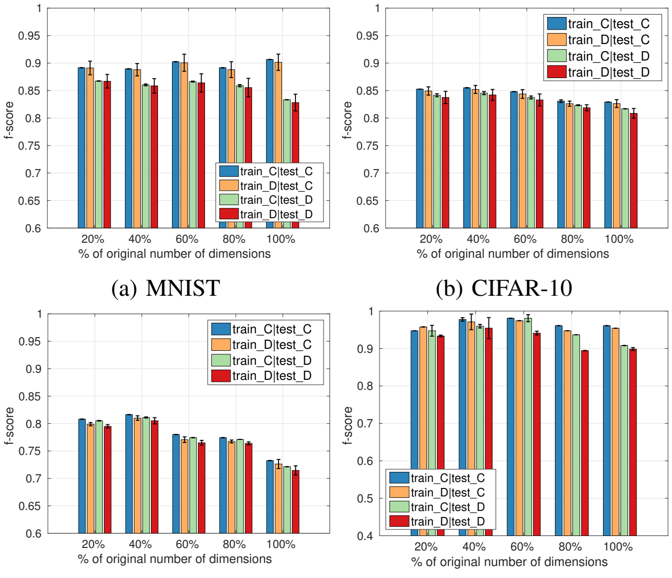
Dataset	Training size	Test size	Normal	Anomaly
MNIST	2,000	1,200	digit ‘9’	digit ‘8’
CIFAR-10	3,650	1,200	airplane	truck
SVHN	4,200	1,200	digit ‘8’	digit ‘0’

Datasets: We generate single-class (unlabeled) datasets considering one of the original classes as the normal class, and another class in the dataset 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 unperturbed anomaly data and normal data, (ii) a distorted test set (test$_D$) with its anomaly points moved closer to the normal data cloud. Table I gives the class and number of samples used in each training and test set.

Experimental setup: Different attack scenarios are simulated (creating train$_D$) by varying the attack severity s_{attack} and attack percentage p_{attack}. In anomaly detection problems we usually do not find a large percentage of attack points within the training set, therefore we choose 5% for p_{attack} (percentage of perturbed points in the training set). 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 with OCSVMs using the RBF kernel in the input feature space.

For nonlinear transformations, we choose 20%, 40%, 60% and 80% of the input dimension as the dimensions to which the datasets are transformed. The test sets are transformed using the same parameters that give the highest compactness for the corresponding distorted training set. The learner then uses the transformed training set to train a OCSVM with a linear kernel, and the resulting model is evaluated using the test sets. The popular SVM implementation, LIBSVM library [6] is used in our experiments. For these experiments the ν parameter of the OCSVM is 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 learner in the input space, we use the same *gamma* values in the low rank kernel approximation using Equation 2 in order to have identical kernel parameters in the input feature space as well as the projected spaces.

Accuracy metric: For comparison purposes, we also train a OCSVM using an undistorted training set (called train$_C$). We report the performance against test$_C$ and test$_D$ using the f-score. We observed similar patterns for each dataset across different experiments, but due to space limitations, graphs and tables of only some are shown.



(a) MNIST

(b) CIFAR-10

(c) SVHN

(d) Omnet simulation

Fig. 5: The performance of OCSVMs under attacks on integrity when the training takes place in different dimensional spaces. We compare the f-scores of OCSVMs trained on train$_C$ and train$_D$ against the two test sets: test$_c$ and test$_D$.

VIII. Results and Discussion

Figures 5a-5d present how the f-score is affected by the non-linear transformation and the adversary's distortion. For each number of dimensions, four results are presented; f-score 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$_D$.

First, the classification performance of OCSVMs trained on nonlinearly transformed data are comparable to the performance of the OCSVM trained on the input feature space, although they require far less computation time. Therefore, the range of the y axes in the graphs have been altered so that the differences can be observed. 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 in the train$_C$|test$_C$ scenario 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 [7].

We observe that the f-scores of train$_D$|test$_D$ in all four datasets, across all the dimensions, are less than the f-scores of train$_C$|test$_D$. 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 manipulate OCSVMs to learn models that are favorable to them.

A comparison of the f-score in the train$_D$|test$_D$ scenario shows that, as the dimension is reduced from the original dimension, the f-scores increase on average. The increase in 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 confirmed by Figure 7, which shows 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 (e.g., 7.25% on SVHN, 23.25% on CIFAR-10, and 19.26% on MNIST).

For completeness, we also tested the online centroid anomaly detection approach proposed by [11] in our problem scenario using the nearest-out replacement policy. The resulting anomaly detection model performed very poorly on the distorted test sets giving a FNR of 1 in all the test cases. We believe that this is due to two reasons (1) we do not make the same assumption as the authors that the initial training dataset only contains normal data, and (2) a hard margin classifier would not perform well when the data is noisy and has many dimensions (such as the benchmark datasets that we use).

The effectiveness of the compactness index for selecting projection directions can be seen by the difference in FNRs in Figures 7a-7c. Although random projection directions have resulted in higher FNRs compared to selective projection directions, it is possible for a randomly sampled direction to be one that minimizes the adversarial distortions. But the probability 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.

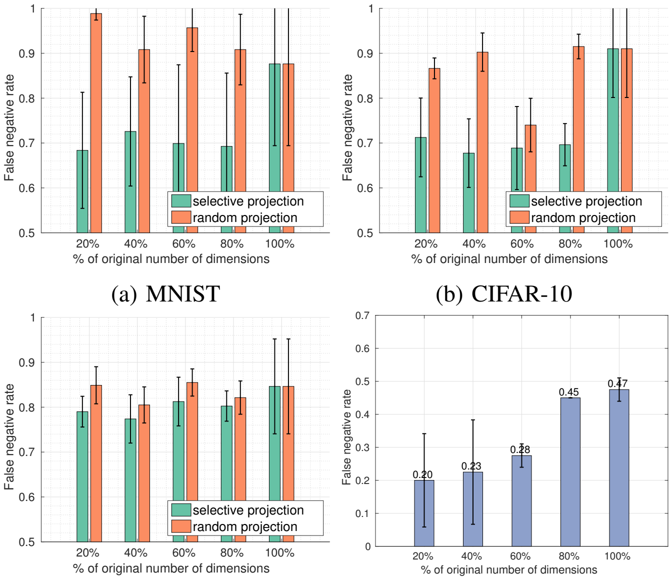
Although it is possible to achieve significantly good results by reducing the dimensionality of data to as low as 20% of the original number of dimensions, we expect the performance to decline when it is reduced below a dataset dependent threshold. We postulate that the explanation of this effect is the reduction in distance between classes (in this case perturbed anomalies and normal data points) with the dimension. As we reduce the dimension of the transformation, we are able to reduce the effects of the adversarial datapoints. But at the same time, there is a significant loss of useful information due to the dimensionality reduction.

Finally, Figure 6 shows the payoff matrix of the adversary and learner for the game introduced in Section V-A. By considering the best responses of both players, we obtain the Nash equilibrium solution to the game, which is $s_{attack} = 0.4$ and 40% of the original number of dimensions. Based on this result, we conclude that it is in the best interest of the learner to always transform data to 40% of the original number of dimensions in this particular problem.

In summary, the above experiments demonstrate that, (i)

		% 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. 6: The utility matrix of the game 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).



(a) MNIST

(b) CIFAR-10

(c) SVHN

(d) Omnet simulation

Fig. 7: The FNR of OCSVMs under an integrity attack (i.e., trained on train$_D$ and evaluated using test$_D$).

OCSVMs are vulnerable to adversarial *attacks on integrity*, (ii) by projecting a distorted dataset to a lower dimension in an appropriate direction we can increase the robustness of the learned model w.r.t. integrity attacks, and (iii) the performance in the projected spaces, when there are no attacks on integrity, is comparable to that in the original dimensional space, but with less computational burden.

IX. Conclusions and Future Work

This paper presents a framework for anomaly detection in the presence of a sophisticated adversary and analyses its effectiveness numerically. The framework combines nonlinear data transformations in selective directions using a novel ranking index that we introduce together with unsupervised anomaly detection using OCSVMs and game theory. The results suggest that 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 results in a OCSVM that is able to identify adversarial samples that would have been missed by a OCSVM in the original dimensional space. Therefore, our approach can be utilized to make a learning system secure by (i) reducing the impact of possible adversarial perturbations by contracting and moving the normal data cloud away from the origin 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 by adding a layer of randomness.

Directions for future work include: how to optimally select the number of dimensions to transform the data to, and formulating different games with randomized and more sophisticated strategies for the players.

References

[1] T. Alpcan and T. Basar, *Network Security: A Decision and Game-Theoretic Approach*, 1st ed., 2010.

[2] T. Alpcan, B. I. P. Rubinstein, and 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.

[3] M. Barreno, B. Nelson, A. D. Joseph, and J. Tygar, “The security of machine learning,” *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[4] J. C. Bezdek and N. R. Pal, “Some new indexes of cluster validity,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 301–315, Jun 1998.

[5] B. Biggio, B. Nelson, and 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.

[6] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[7] S. M. Erfani, M. Baktashmotlagh, S. Rajasegarar, S. Karunasekera, and 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.

[8] A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard, “Robustness of classifiers: from adversarial to random noise,” *CoRR*, vol. abs/1608.08967, 2016.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” *CoRR*, vol. abs/1412.6572, 2014.

[10] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, “Adversarial Machine Learning,” in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, 2011, pp. 43–58.

[11] M. Kloft and P. Laskov, “Security analysis of online centroid anomaly detection,” *Journal of Machine Learning Research*, vol. 13, pp. 3681–3724, 2012.

[12] C. Laufer, *The Hobbyist’s Guide to the RTL-SDR: Really Cheap Software Defined Radio*, 2015.

[13] Q. V. Le, T. Sarlos, and A. J. Smola, “Fastfood: Approximate Kernel Expansions in Loglinear Time,” in *Proceedings of International Conference on Machine Learning (ICDM)*, 2013.

[14] J. Nash, “Non-cooperative games,” *Annals of Mathematics*, vol. 54, pp. 286–295, 1951.

[15] A. Rahimi and B. Recht, “Random Features for Large-Scale Kernel Machines,” in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2008, pp. 1177–1184.

[16] S. Rajasegarar, C. Leckie, and 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.

[17] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support Vector Method for Novelty Detection,” in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2000, pp. 582–588.

[18] P. Weerasinghe, “Omnet simulation dataset,” May 2018. [Online]. Available: https://github.com/sandamal/omnet_simulation

[19] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and 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.