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.

I. INTRODUCTION

In a given populated area (e.g., city block), a multitude of individuals communicate with each other using various methods. Individuals use a plethora of devices for communication purposes that 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. 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.

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 SDRs 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 [4] 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. 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.

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. 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). By distorting the input data used by a learning algorithm, the adversaries can force a learner to learn a model that favors the adversary. 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 training dataset. Many machine learning methods, such as One-Class Support Vector Machines (OCSVM) [6], 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.

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. As part of this framework, we introduce a novel index to identify suitable directions for data transformations and study the resistance added by such transformations against an adversarial opponent through numerical experiments. We pose the problem of finding an appropriate defense 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 [1]. We show through numerical experiments 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.

II. BACKGROUND AND RELATED WORK

To improve the efficiency of kernel machines, [5] 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.

Among adversarial defense techniques for SVMs, [8] 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. Again, for binary SVMs, [2] introduced a poisoning attack algorithm that finds the optimal attack point by maximizing the hinge loss of a SVM when tested on a validation set. In an online setting, [3] analyzed the effects of adversarial injections on the centroid anomaly detection algorithm.

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. To succeed in this, the attacker would inject malicious data during training in order to make the learner learn a compromised decision boundary that exaggerates the region where normal data points lie.

In the context of the SDR application scenario, the rogue agents would be the adversaries who wish to make the learner classify rogue agents as civilians during the evaluation phase. To achieve this, they would first 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

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., $p_{\text{attack}}$) in addition to the severity of the distortion (i.e., $s_{\text{attack}}$). Figure 1 illustrates the data distributions when different levels of attack severities are applied to the anomaly data. At higher $s_{\text{attack}}$ values, the anomaly data points are moved closer to the origin, reducing the gap between the origin and the separating hyperplane. 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.

V. DEFENSE FRAMEWORK

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 (Figure 2) we use a projection mechanism that projects data points to lower dimensional spaces in a manner that conceals the potential distortions of an adversary. 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). 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

$$z((X + D)_i) = \frac{\sqrt{2}}{r} \cos \left(\sqrt{2\gamma}(X + D)_i A + b\right), \quad (1)$$

where $\gamma$ 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 \([0, 2\pi]\) \([5]\).

By randomly drawing projection directions from some distribution, the learner also introduces a layer of unpredictability to the adversary-learner problem, which gives a significant advantage from a security perspective. But this unpredictability can also be seen as the main caveat of using random projections as some projections result in better separated volumetric clouds than the original ones, while others result in the data from different classes being overlapped. Therefore, we propose a compactness measure to rank suitable projection data from different classes being overlapped. Therefore, the learner would evaluate the appropriateness of a randomly sampled projection direction using (2). The compactness of projection \( P_i \), where \( \mu_i \) 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

\[
\text{compactness of } P_i = \frac{d(0, \mu_i)}{\left( \sum_{x \in P_i} d(x, \mu_i) \right) / n}. \tag{2}
\]

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 \( s_{\text{attack}} \) to be 0. If the rogue agents closely mimic the civilians during training (resulting in a small shift of the margin) we consider \( s_{\text{attack}} \) to be small. Conversely if rogue agents change their patterns to ones that are significantly different than those of the civilians, we consider \( s_{\text{attack}} \) 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.

\( x_A \in \{0, 0.3, 0.4, 0.5\} \),
\( x_L \in \{20\%, 40\%, 60\%, 80\%, 100\%\} \).

Due to the adversary’s ability to evaluate the model during testing (i.e., calculating the false negative rate (FNR)), we design the utility function \( U_A \) 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

\[
U_A(x_A, x_L) = 1 + \text{FNR} - \frac{1}{2} s_{\text{attack}},
\]
\[
U_L(x_A, x_L) = f\text{-score}.
\]

VI. NETWORK SIMULATION AND EXPERIMENTS

Simulations are performed using the INET framework for OMNeT++ (datasets available at [7]). 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.

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 duration of reception, message length, inter arrival time (IAT), carrier frequency, bandwidth and bitrate are used as features for the learning algorithm.

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.

Datasets: Using the collected data, we create two training datasets, train\(_C\), train\(_D\) and two test datasets test\(_C\) and test\(_D\). 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). 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.

Experimental setup: We choose 20\%, 40\%, 60\% and 80\% of the input dimension as the dimensions to which the datasets are transformed. For comparison, we test all the attack scenarios with OCSVMs using the RBF kernel in the input space. The \(\nu\) parameter of the OCSVM is kept fixed at 0.13 across all experiments conducted. Since \(\nu\) 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 \(\gamma\) value (0.009) in the low rank kernel approximation using (1) in order to have identical
kernel parameters in the input space as well as the projected spaces.

VII. RESULTS AND DISCUSSION

Figure 3a presents 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 trainC, and tested with testC; (ii) trained with trainC and tested with testD; (iii) trained with trainD and tested with testC; and finally (iv) trained with trainD and tested with testD. We observe that the f-scores of trainD|testD, across all the dimensions, are less than the f-scores of trainC|testD. 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 trainD|testD 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 3b, 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.

Finally, Figure 4 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 $s_{\text{attack}} = 0.4$ 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 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.

VIII. 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 non-linear data transformations in selective directions using a novel ranking index that we introduce together with unsupervised anomaly detection using OCSVMs and game theory. 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.

ACKNOWLEDGMENT

This work was supported in part by the Australian Research Council Discovery Project under Grant DP140100819 and by a grant from the Northrop Grumman Corporation.

REFERENCES

Author/s:
Weerasinghe, S; Erfani, SM; Alpcan, T; Leckie, C; Riddle, J

Title:
Detection of Anomalous Communications with SDRs and Unsupervised Adversarial Learning

Date:
2019-02-08

Citation:

Persistent Link:
http://hdl.handle.net/11343/241555

File Description:
Accepted version