2018 IEEE International Conference on Information Reuse and Integration for Data Science

Adversarial Anomaly Detection Using Centroid-based Clustering

Imrul Chowdhury Anindya, Murat Kantarcioglu
Department of Computer Science, The University of Texas at Dallas
Richardson, TX, USA
Email: {ixa140430, muratk}@utdallas.edu

***Abstract*—As cyber attacks are growing with an unprecedented rate in the recent years, organizations are seeking an efficient and scalable solution towards a holistic protection system. As the adversaries are becoming more skilled and organized, traditional rule based detection systems have been proved to be quite ineffective against the continuously evolving cyber attacks. Consequently, security researchers are focusing on applying machine learning techniques and big data analytics to defend against cyber attacks. Over the recent years, several anomaly detection systems have been claimed to be quite successful against the sophisticated cyber attacks including the previously unseen zero-day attacks. But often, these systems do not consider the adversary's adaptive attacking behavior for bypassing the detection procedure. As a result, deploying these systems in active real-world scenarios fails to provide significant benefits in the presence of intelligent adversaries that are carefully manipulating the attack vectors. In this work, we analyze the adversarial impact on anomaly detection models that are built upon centroid-based clustering from game-theoretic aspect and propose adversarial anomaly detection technique for these models. The experimental results show that our game-theoretic anomaly detection models can withstand attacks more effectively compared to the traditional models.**

***Keywords*-Anomaly Detection; Adversarial Machine Learning; Clustering; Mimicry Attack;**

I. Introduction

As more and more developing nations are getting digitalized without adopting enough security precautions and penetration testing drills during building and maintaining their cyber-infrastructures, the attack surface of cyber attacks is expanding at a tremendous speed [1]. A recent incidence can be exemplified by the heist of foreign reserve from the central bank of Bangladesh, in which unidentified hackers tried to steal $951 million by compromising the SWIFT Alliance Access software that the bank was using for its foreign transactions [2]. On the other hand, the *dark web* is being swarmed by an increasing number of cyber criminals for purchasing stolen cyber espionage tools and zero-day vulnerabilities of state-sponsored agencies through the use of cryptocurrencies. In 2016, an infamous hacking group named The Shadow Brokers auctioned several exploit tools used by the Equation group which was believed by many as a secret cohort working under the direct influence of the National Security Agency (NSA) [3]. Subsequently, one of the released exploits called the EternalBlue was used by the WannaCry ransomware in 2017 to propagate reportedly into 2,00,000 computers in at least 150 countries causing an estimated loss of $4 billion [4]. While attacks against traditional networks are dominating the threat landscape, emerging technologies such as Internet of Things (IoT) are creating new frontiers of cyber attacks. An alarming occurrence can be attributed to the Mirai botnet which infected 4,93,000 IoT devices to perform DDoS attacks in 2016 [4].

Due to the growing attack surface and the adoption of different obfuscation techniques to avoid detection, signature-based detection systems are no longer considered a sufficient defense maneuver against the evolving cyber attacks. While supervised machine learning techniques provide considerable supplementary defense, their effectiveness is impacted when the training data set has skewed distribution of benign and malicious samples or a novel class of attacks appear in the testing data. As a result, anomaly detection models have garnered widespread acceptance to be a part of holistic protection system. However, previous works have shown that clustering-based anomaly detection models are not robust against adversarial attempts [5], [6]. Therefore, making these models robust against different kinds of adversarial attacks seeks special research endeavor for ensuring the security of machine learning. Hence in this work, we particularly addressed the influence of mimicry attack on centroid-based clustering models and proposed a game-theoretic way of choosing parameters to make these models resilient against the attack.

A. Our Contributions

In this work, we choose traditional centroid-based clustering models as the basis of the anomaly detection model. We then analyze how an intelligent adversary can try to evade detection by means of modifying the features while incurring minimum cost. Based on this threat model, we propose the adversarial robust anomaly detection models as replacement to the traditional ones. To the best of our knowledge, this is the first work addressing mimicry attacks against centroid-based clustering models using game theoretical framework. Our major contributions can be summarized as follows.

- We develop the theory of an adversary's optimal strategy to perform mimicry attack against a clustering-based anomaly detection model.

978-1-5386-2659-7/18/$31.00 ©2018 IEEE
DOI 10.1109/IRI.2018.00009

- We formulate the defender's strategy in figuring out the optimal parameters to defend the attack from a game-theoretic viewpoint.
- We empirically analyze the adversarial impact on our proposed adversarial models as well as the traditional ones and find that our models can withstand mimicry attacks more effectively.

The remainder of this paper is organized as follows. In Section II, we introduce the reader to the concept of adversarial anomaly detection. In Section III, we briefly describe some noteworthy works related to our research. In Section IV, we describe the traditional anomaly detection model for which we propose the adversarial model. Then, we formally discuss the attacker's optimal adaptive strategy and the defender's optimal response. The data set, experimental setup and empirical results are presented in Section V. Section VI draws the conclusion and highlights some future directions.

II. Background

In this Section, we describe the necessary background of adversarial anomaly detection.

A. Anomaly Detection

Anomaly detection refers to the problem of finding the samples that deviate from some well-defined region of normalcy, possibly through the use of some thresholding mechanism. The samples are called anomalies or outliers and show significant divergence in their properties or behaviors. Anomaly detection differs from supervised-learning methods in the sense that they do not necessarily need supervised training through labeled dataset [7]. Consequently, anomaly detection models provide real benefits when there is a significant imbalance in the distribution of two classes in the training data set or there appears novel bad classes in the testing data set. Anomaly detection models are used in a broad spectrum of application domains such as network intrusion detection, fraud detection, medical diagnosis, military surveillance, monitoring of critical infrastructures and so on [8].

B. Adversarial Anomaly Detection

Adversarial Machine Learning is relatively a new research area at the intersection of machine learning and security informatics. A wide range of machine learning based security applications are designed by considering a stationary environment where the training and testing data are assumed to be generated from the same distribution [9]. But in the presence of adaptive adversaries having consummate skills to modify the features or control the training data, this hypothesis may not hold true. As a result, the real-world effectiveness of a machine learning model can slump down significantly from a higher level that was found from a static, defender controlled set of experiments having no external influence. Hence, evaluating the robustness of these models against adversarial manipulations requires substantial research effort.

1) Taxonomy of Adversarial Influence: Huang et al. introduced a taxonomy of adversarial effects on machine learning models [10]. Later, Corona et al. proposed a high-level categorization of adversarial tactics against intrusion detection systems [11]. Based on these works, we can classify the adversarial impacts into the following major types.

- *Poisoning Attack:* In some scenarios, the attacker has a significant level of control over the training data and thus has the ability to inject craftily designed samples to poison the training process and compromise the detection model. Many detection models that are designed to periodically re-train themselves in an online or streaming fashion to comply with the changing trends in the underlying data distribution may be vulnerable to this kind of attack [12], [13].
- *Mimicry/Evasion Attack:* Mimicry attack [14], [9], [15] is the most prevailing type of attack that is launched against a machine learning model during its course of operation. This attack can be materialized by modifying the features of an attack sample so that it looks like a legitimate one. Common examples include injecting good words in a spam email or obfuscating malware binaries to hide their malware-centric features.
- *Availability/Overstimulation Attack:* In the availability attack [16], [17], the attacker generates a large number of spurious samples which do not have real malicious properties but overwhelm the detection system with lots of false positive alerts, consequently compelling the security administrator to repeal the system.
- *Denial-of-Service Attack:* In this attack, the attacker tries to disable the detection system or create stagnation in the detection procedure by generating algorithmic complexity through crafted samples, eventually creating a way for the actual malicious sample to bypass the screening process [18], [19], [20].
- *Exploratory Attack/Reverse Engineering:* In the reverse engineering attack [21], the attacker tries to determine the inner working mechanism of the detection model by repeatedly probing it with carefully constructed samples. This is one of the reasons behind why *security-by-obscurity* is not endorsed.

2) Arms Race Between Attacker and Defender: In the field of computer security, the conflict between the attacker and the defender can be modeled as a game between two intelligent rational agents where each one tries to maximize its score by taking the best move against the other one's optimal strategy at each level of the game. This can be thought of as a *reactive arms race* between the two as explained by Biggio et al. [14]. At each level, the attacker tries to explore different aspects of the detection model

and figure out the vulnerabilities. The attacker then devises strategies to exploit those vulnerabilities for bypassing the detector. The defender reacts by analyzing the new attack samples and amending or adjusting the model to prevent those attacks. This kind of arms race has reached extensive sophistication in the field of spam filtering and malware detection. In this work, we apply basic leader follower structure to predict the end state (i.e., equilibrium) of such an arms race.

In summary, the goal of adversarial anomaly detection is to set up the defender's strategy in making the anomaly detection model robust against adversarial attack.

III. Related Literature

Several research endeavors have been made for understanding the adversarial impact on machine learning models. But to the best of our knowledge, no work has been done on understanding the effect of mimicry attack on clustering models. Below we summarize the works that closely match with our work.

Xu et al. developed a system to automatically evade two PDF malware classifiers with 100% success rate [9]. The team utilized Genetic Programming to perform object mutations in the malicious PDF samples until those were able to circumvent the classifiers while retaining their malicious properties. The research found that the weakness in those classifiers could be attributed to the use of superficial features that are not inherently associated with benign (or malicious) behavior but stochastically prevalent in benign (or malicious) samples.

Zhou et al. developed optimal Support Vector Machine learning strategies against active adversaries having free-range and restrained data-corruption capabilities [22]. Later they addressed the problem of having multiple types of adversaries against learning models and devised a nested Stackelberg game framework that offered more reliable defense [23].

Wang et al. performed an empirical study of adversarial attacks against popular classification algorithms in the context of detecting crowd-sourcing systems in which paid human workers actively perform certain tasks to circumvent security mechanisms (e.g., CAPTCHAs) and found that the algorithms could be highly vulnerable to simple evasion attacks and powerful poisoning attacks [24].

Kloft et al. analyzed how an online centroid anomaly detector with finite sliding window of training data performs under poisoning attack [25]. They showed that if the attacker cannot control a certain percentage of the training data, this attack fails even with an arbitrarily lengthy effort. By experimenting on a real HTTP traffic dataset, they found that an attacker needs to control 5-10% of the traffic to successfully launch a poisoning attack, which may not be possible on sites with high volume traffic.

Dutrisac et al. showed how the adversary could inject a few carefully chosen *not too unusual* samples to the training procedure of multiple clustering models so that two different clusters of good and bad class respectively merge into a single cluster of good class [5].

Biggio et al. demonstrated poisoning attack against single-linkage clustering algorithm to subvert Malheur, an open-source tool for behavioral malware clustering and found that the attacker needs to inject very small percentage of attacks into the input data [6].

IV. The Adversarial Anomaly Detection Model

Our adversarial anomaly detection model is robustly designed for centroid-based clustering systems which optionally use preprocessing steps such as data normalization and dimensionality reduction. However, the adversarial model is applicable invariably of whether these preprocessing steps are used or not, as we will see at the end of this section. Since the models are unsupervised, they only use benign data to train themselves.

A. Data Preprocessing

Our adversarial model is built in a robust way by considering possible data preprocessing steps as described below.

1) Data Normalization: Typically different features in a data set are represented using different scales of reference and thus may not be comparable to one another. For example, in the context of network traffic, the features *duration of network flow* and *number of packets sent* have different scales. If the scales are not normalized then the detection model would be biased towards the feature of larger magnitude. Moreover, there can be categorical features whose values are not comparable on numeric scale. So at first, the categorical features are replaced by multiple boolean features, each one corresponding to a particular value of the original categorical feature. Only the feature corresponding to the value present in the current instance is considered *True* while others are deemed *False*. Finally, the boolean features are represented with 1 and 0 to indicate *True* and *False* respectively. Then the standard *data scaling* technique is adopted. If min_j and max_j represent the minimum and maximum values of the j-th feature respectively and y_j corresponds to a particular value of that feature, then y_j is scaled as-

$$x_j = \frac{y_j - min_j}{max_j - min_j} \tag{1}$$

2) Dimensionality Reduction: In real data sets, more than one feature may come from the same underlying property and thus intrinsically represent the same thing. These correlated features are often overlooked by the analyst during the initial phase of data set creation. As a result, the learning algorithm trained on the data set suffers from implicit redundancies and gets biased. Moreover, as identified by

Zimek et al., the concentration of distances, interpretability of scores and exponential search space for high-dimensional data affects the performance of anomaly detection models [26]. These phenomenons are jointly known as the *curse of dimensionality*. To get rid of these problems, the model reduces the dimensions using principal component analysis (PCA) [27]. PCA is a statistical procedure to transform a set of values of possibly correlated features into a set of values of linearly uncorrelated features referred to as principal components. The first principal component is defined to have a direction for which the data set has the highest possible variance. Each subsequent principal component, in turn, has a direction that is orthogonal to that of the preceding components but also has the next highest variance possible. The number of principal components can be at most the number of original features, though in practice it is chosen to be much lower depending on the necessity. PCA generates an n-by-m coefficient matrix ($m \leq n$) which when multiplied to an n-dimensional feature vector, transforms it to an m-dimensional vector in the principal component space.

B. Clustering

After applying the preprocessing steps on the benign training samples, clusters of benign samples are formed using some centroid-based clustering algorithm (e.g., *k-Means, bisecting k-Means, k-Medians, k-Medoids*) that uses numerical distance metric to measure similarity between the samples. The model assumes that there can be several categories of benign samples in the data set and the samples of the same category form their own cluster. For example, in the context of network traffic, different types of benign traffic (e.g., *http*, *ssh* etc.) might form their own clusters. Hence, if a testing sample appears as anomalous to all of the generated clusters, then it is considered as malicious.

Let us assume that $X = \langle x_1, x_2, \cdots, x_n \rangle$ is the normalized testing point in the n-dimensional feature space that is passed to the clustering model. The model has k clusters in the m-dimensional ($m \leq n$) principal component space and the center of the i-th cluster is represented as $C^{(i)} = \langle c_1^{(i)}, c_2^{(i)}, \cdots, c_m^{(i)} \rangle$. The testing point is projected to the principal component space on which the clustering model is built and represented as $\overline{X} = XM$, where M is the PCA coefficient matrix. If the Euclidean distances of $\overline{X}$ from all the k cluster centers are greater than some predefined threshold t, then it is considered an anomalous (malicious) point. Now suppose, $F(X)$ be the output of the anomaly detection model for the point X and $\mathcal{D}$ be the function returning the Euclidean distance between two points. Also assume that if X is an anomalous point, then $F(X) = +1$ and $F(X) = -1$ otherwise.

$$F(X) = \begin{cases} +1 & \text{if } \forall i \in \{1, \cdots, k\} : \mathcal{D}(\overline{X}, C^{(i)}) > t \\ -1 & \text{otherwise} \end{cases} \tag{2}$$

The defender's goal in any detection model is to minimize the weighted sum of the number of false positives and the number of false negatives, where weights are set according to the type of the problem and the requirement in context. For example, in the case of SPAM filtering restricting a legitimate email is less desirable than allowing a junk email, i.e., false positive weight needs to be higher than the false negative weight. On the contrary, for network intrusion detection, restricting any suspicious traffic is of high priority i.e., false negative weight needs to be higher than false positive weight. Now let, $\mathbf{X_A}$ be the set of malicious testing samples, $\mathbf{X_B}$ be the set of benign testing samples, $I_{\{e\}}$ be the indicator random variable for some event e and additionally, $w_1 \in [0, 1]$ and $(1-w_1)$ be the weights for false positives and false negatives respectively. Then assuming no adversarial attack, the objective of the traditional model is to choose the parameters $\langle k, t \rangle$ to minimize the following expression-

$$\begin{aligned} \underset{\langle k,t \rangle}{\operatorname{argmin}} \quad & w_1 \sum_{X \in \mathbf{X_B}} I_{\{F(X)=+1\}} + \\ & (1 - w_1) \sum_{X \in \mathbf{X_A}} I_{\{F(X)=-1\}} \end{aligned} \tag{3}$$

The near-optimal value of the parameters $\langle k, t \rangle$ can be found by performing a grid-search using a small cross-validation data set that we can reasonably assume as being available to the defender.

C. Optimizing Parameters Against Mimicry Attacks

In the threat model of mimicry attack, the attacker is assumed to have the ability to modify the features of the malicious sample to make it look legitimate. But, intuitively the attacker would be unwilling to let the malicious point move far away from its original position in the feature space since greater displacement often entails loss of malicious utility [22]. As a result, we make the assumption that each feature's value can be modified by up to some certain amount with the restriction that the new value must lie within the original domain, D_j of the j-th feature. This certain amount which we call the *feature modification threshold* can be estimated by the defender's domain knowledge and represented by δ_j for the j-th feature. If modifying the j-th feature by the slightest amount nullify the malicious properties of the sample, then $\delta_j = 0$.

Considering the described anomaly detection model, the attacker's optimal strategy for mimicry attack is realized by modifying the features of the malicious sample within the respective feature modification thresholds while minimizing the overall modification cost. Now let, $X' =$

$\langle x'_1, x'_2, \cdots, x'_n \rangle$ be the new position of the original malicious point $X = \langle x_1, x_2, \cdots, x_n \rangle$ after modifying its features according to the strategy and $X'M$ be the projection on the principal component space. For simplicity, we consider that each feature has the same modification cost of 1. So, the attacker's objective turns out to be minimizing the overall cost, $\sum_{j=1}^{n} |x'_j - x_j|$. X' is treated as benign by the detection model only if the projected point $X'M$ has a distance of less than t from at least one of the cluster centers $C^{(i)}; i \in \{1, ..., k\}$ as described in section IV-B. Therefore, using matrix notation, the attacker's optimal strategy can be represented by the following optimization problem-

$$\begin{aligned} \underset{X'}{\text{argmin}} \quad & (X' - X)(X' - X)^T \\ \text{s.t.} \quad & \exists i \in \{1, ..., k\} : (X'M - C^{(i)})(X'M - C^{(i)})^T \leq t^2 \\ \text{and} \quad & \forall j \in \{1, ..., n\} : (x'_j - x_j)^2 \leq \delta_j^2,\ x'_j \in D_j \end{aligned} \tag{4}$$

Notice that the objective in Expression 4 has been changed to $\sum_{j=1}^{n} (x'_j - x_j)^2$ for the purpose of avoiding the absolute value operator during optimization. The first constraint indicates the condition for the new attack point X' to be treated as benign. The second constraint indicates that each feature's value must be modified by an amount not higher than the respective feature modification threshold and also the new value must be chosen from the valid domain of the feature. This kind of optimization problems are known as *Quadratically Constrained Quadratic Program (QCQP)* and can be solved by powerful optimization tools.

From game-theoretic aspect, a prudent defender responds by undertaking an adversarial model that is expected to withstand the above-mentioned strategy of the attacker. This can be achieved by replacing the objective denoted by Expression 3 with the following-

$$\begin{aligned} \underset{\langle k,t \rangle}{\text{argmin}} \quad & w_1 \sum_{X \in \mathbf{X_B}} I_{\{F(X)=+1\}} + \\ & (1 - w_1) \sum_{X \in \mathbf{X_A}} I_{\{F(X)=-1 \text{ or } \exists X' : F(X')=-1\}} \end{aligned} \tag{5}$$

where X' corresponds to a valid solution for the optimization problem in Expression 4. Thus Expression 5 searches for the optimum values of the parameters $\langle k, t \rangle$ after incorporating the attacker's optimal strategy for mimicry attack into consideration.

Notice that, even if the the data preprocessing steps are not used, the model would work seamlessly by considering $C^{(i)}$ to be the i-th cluster center in the original feature space and replacing the PCA coefficient matrix, M with n-dimensional identity matrix.

V. Experiments

We implement the machine learning algorithms using Scala functional programming language and Spark MLlib machine learning library on an Intel Core i7 3.40GHz machine with 16GB of RAM. To solve the QCQP in Expression 4, we use IBM ILOG CPLEX [28] optimizer which provides Java interface to be integrated to our Scala programs.

Table I: Number of samples in the data sets

	Benign	**Malicious**	**Total**
Training	349445	0	349445
Validation	10000	10000	20000
Testing	153106	152452	305558

A. Data Set

A labeled dataset is necessary to evaluate the performance of an anomaly detection model. KDD Cup 1999 data set [29], available from the *UCI Machine Learning Repository*, is one of the very few processed and labeled intrusion detection data sets which is widely used to compare anomaly detection methods. This data set was produced by Stolfo et al. [30] by extracting 41 features of network traffic captured from DARPA 1998 IDS evaluation program simulating a typical U.S. Airforce LAN for 7 weeks. The feature set includes *protocol type*, *number of source bytes*, *number of shell obtained*, *% of connections to the same service* and so on. The data set contains a total number of 38 types of attacks. These include *syn flood, guessing password, buffer overflows, port scanning* and others. We divide the whole data set into training, validation and testing sets after removing the duplicates. The training set is comprised of benign samples only. The validation set is used for computing the values of the parameters $\langle k, t \rangle$ of the traditional and adversarial models. Both the validation and testing sets have almost equal number of benign and malicious samples for preventing data skewness to create bias during the experiments. Naturally, a few samples are discarded for that purpose. Also, the number of samples in the validation data set is kept small to comply with our assumption that the defender has a limited number of malicious samples. Table I provides the summary statistics of the data sets. After normalization, the number of features in the data sets expands to 123.

B. Experimental Setup

At first, we set the number of principal components to 30, preserving 99.5% of the variance in the normalized data set. We build k-Means (KM) and bisecting k-Means (BKM) models as representative to the traditional centroid-based clustering models. Before constructing the corresponding adversarial models (AD-KM and AD-BKM respectively), we consider that the attacker has the capability of modifying the numeric features by some reference percentages notably 5%, 15% and 25%. Since the features are normalized and have the domain of $[0, 1]$, the feature modification thresholds (δ_j) for the above cases are set to 0.05, 0.15 and 0.25

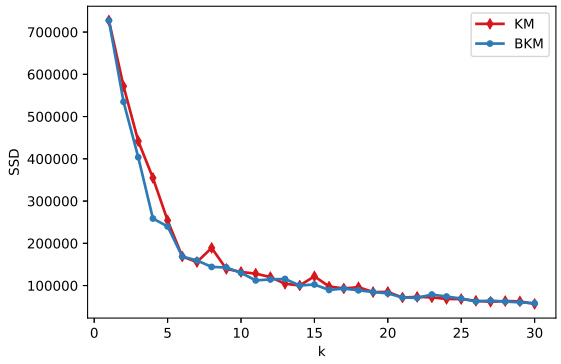

Figure 1: Sum of squared distances of the training points from their respective nearest cluster centers for different values of k

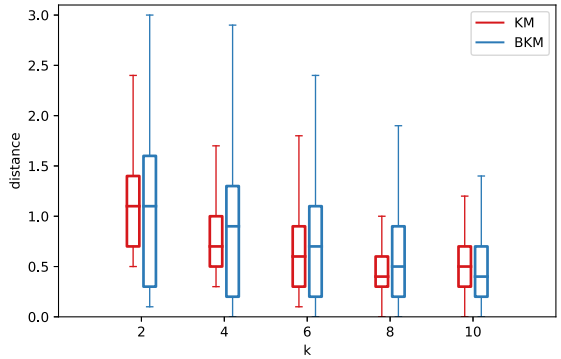

Figure 2: Distribution of distances between training points and their respective nearest cluster centers for different values of k

respectively. However, we consider the features derived from the categorical ones as unchangeable (i.e., $\delta_j = 0$). This assumption is substantiated by the fact that while numeric features such as *number of source bytes* can be changed easily, changing categorical features such as *protocol type* (UDP/TCP) may result in the nullification of malicious properties. For comparing the models using accuracy metric, we set equal weights to false positives and false negatives (i.e., $w_1 = 0.5$). Based on this setting, we find the optimal $\langle k, t \rangle$ values that minimize expression 3 and expression 5 for traditional and adversarial models respectively through grid-search using the validation data set. For doing the grid-search, a set of possible k values is selected by using the *elbow method* [31], intuitively choosing several k values near the elbows. Figure 1 shows the elbows by plotting the sum of squared distances (SSD) of the training points from their respective nearest cluster centers for different values of k. Similarly, a set of possible t values is selected by looking at the distribution of the distances between the benign training points and their respective nearest cluster centers as depicted in Figure 2. Based on these figures, the grid-search for optimum $\langle k, t \rangle$ values is performed on the set $\{1, 2, ..., 10\} \times \{0.5, 0.6, ..., 3.0\}$ for all the models.

C. Computational Overhead

The clustering models need little amount of time to be constructed. The main time consumption happens in the grid-search for finding the optimal $\langle k, t \rangle$ values of the adversarial models due to the reason we describe now. Finding if a solution exists for Expression 4 takes about 70 milliseconds on average. There can be at most 10,000 true positive samples in the validation data set for which the solutions are sought. So for different values of the parameters $\langle k, t \rangle$, it can take up to 116 minutes to know how many samples can be modified successfully to bypass the detection model and thus to find the value of Expression 5. We can reduce this time requirement drastically by resorting to the following technique. For each cluster, a binary search is performed to find the possibly farthest true positive sample to have a successful mimicry attack targeting that cluster and the distance to that sample is remembered. Then during finding the solution of Expression 4 for a sample, all those clusters are discarded from consideration whose distance to that sample exceeds the corresponding remembered distance. Using this technique, we are able to finish the grid-search within 48 hours.

D. Results

We investigate the robustness of our adversarial anomaly detection models as we increase the severity of the mimicry attacks. We assume that the attacker knows of a few samples that are considered benign by the model. When performing mimicry attack for the malicious sample $X = \langle x_1, x_2, \cdots, x_n \rangle$, the attacker targets a point $X^* = \langle x_1^*, x_2^*, \cdots, x_n^* \rangle$ from the pool of known benign points that has the lowest distance from X. Then the attacker nudges the modifiable feature values of X towards the corresponding feature values of X^* by some factor f_{attack}, which represents the aggressiveness of the attack. Based on this, the j-th feature of the new malicious point gets the value of-

$$x'_j = \begin{cases} x_j & \text{if } \delta_j = 0 \\ x_j + f_{attack}(x_j^* - x_j) & \text{otherwise} \end{cases} \quad (6)$$

Notice that, this setting results in a free-range attack [22] in which the attacker's feature modification capability is curbed indirectly because of moving the attack point towards the nearest benign target point instead of the cluster centers.

Table II and III shows the accuracies of the models for different attack intensities when the attacker has 100 and 50 benign points respectively as targets for the mimicry attacks. We observe that, under no attack ($f_{attack} = 0.0$),

Table II: Accuracies of the models for different attack intensities considering the attacker has 100 target points

		$f_{attack} = 0.0$	$f_{attack} = 0.2$	$f_{attack} = 0.4$	$f_{attack} = 0.6$	$f_{attack} = 0.8$	$f_{attack} = 1.0$
KM		0.98	0.98	0.96	0.82	0.81	0.80
	$\delta_j = 0.05$	0.98	0.98	**0.98**	**0.95**	**0.88**	**0.81**
	$\delta_j = 0.15$	0.98	0.98	**0.98**	**0.96**	**0.96**	**0.83**
AD-KM	$\delta_j = 0.25$	0.98	0.98	**0.98**	**0.96**	**0.96**	**0.83**
BKM		0.98	0.98	0.96	0.90	0.89	0.82
	$\delta_j = 0.05$	0.98	0.98	0.96	0.90	0.89	0.82
	$\delta_j = 0.15$	0.98	0.98	**0.97**	**0.96**	0.89	0.82
AD-BKM	$\delta_j = 0.25$	0.98	0.97	**0.97**	**0.96**	0.89	0.82

Table III: Accuracies of the models for different attack intensities considering the attacker has 50 target points

		$f_{attack} = 0.0$	$f_{attack} = 0.2$	$f_{attack} = 0.4$	$f_{attack} = 0.6$	$f_{attack} = 0.8$	$f_{attack} = 1.0$
KM		0.98	0.98	0.96	0.83	0.52	0.52
	$\delta_j = 0.05$	0.98	0.98	**0.98**	**0.97**	**0.88**	**0.88**
	$\delta_j = 0.15$	0.98	0.98	**0.98**	**0.96**	**0.96**	**0.82**
AD-KM	$\delta_j = 0.25$	0.98	0.98	**0.98**	**0.96**	**0.96**	**0.82**
BKM		0.98	0.98	0.95	0.89	0.82	0.52
	$\delta_j = 0.05$	0.98	0.98	0.95	0.89	0.82	0.52
	$\delta_j = 0.15$	0.98	0.98	**0.96**	**0.96**	**0.89**	**0.89**
AD-BKM	$\delta_j = 0.25$	0.98	0.97	**0.97**	**0.96**	**0.89**	**0.82**

the adversarial models (AD-KM and AD-BKM) developed for different values of δ_j achieve the same accuracies as of their traditional counterparts (KM and BKM). As the intensity of attack (f_{attack}) increases, the adversarial models tend to achieve higher accuracies than those achieved by the traditional ones as evident from the highlighted entries in the two tables. We see that, for aggressive attacks the adversarial models achieve up to 46% better accuracies than the traditional models. Moreover, we observe that, the adversarial models developed using higher values of δ_j (specifically 0.15 and 0.25), provide better resistance against the attacks. This happens because of using the free-range attack of Equation 6 which does not limit the feature modification capability rigorously. However, in real scenarios the defender should set the value of δ_j reasonably based on domain knowledge before constructing the adversarial models, to achieve the best protection possible against the attack.

VI. Conclusion

Machine learning models show phenomenal success in detecting and preventing cyber attacks. Yet, their benefits may come to a grinding halt in the presence of shrewd adversaries carefully exploiting the inherent weaknesses in the models. In this work, we proposed strategies to make centroid-based clustering models robust against mimicry attacks. We showed that choosing parameters by modeling adversarial capabilities allow the centroid-based clustering models to be more resilient especially under powerful mimicry attacks.

As a future work, we intend to apply the adversarial modeling-based parameter selection techniques to other type of anomaly detection methods. In addition, we plan to expand our game theoretical modeling to multi-interaction and multi-step attacks that could be launched by sophisticated adversaries.

Acknowledgements

The research reported herein was supported in part by NIH award 1R01HG006844, NSF awards CNS-1111529, CICI-1547324, and IIS-1633331 and ARO award W911NF-17-1-0356.

References

[1] The New York Times, "Hackers Find 'Ideal Testing Ground' for Attacks: Developing Countries," https://goo.gl/tZzA89, (Accessed on 04/11/2018).

[2] Forbes Magazine, "What The Bangladesh SWIFT Hack Teaches About The Future Of Cybersecurity and Cyberwar," https://goo.gl/N7uWeR, (Accessed on 04/11/2018).

[3] CyberScoop, "Leaked NSA tools, now infecting over 200,000 machines, will be weaponized for years," https://goo.gl/BFFbq4, (Accessed on 04/11/2018).

[4] Symantec, "Internet Security Threat Report 2017," https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf, (Accessed on 04/11/2018).

[5] J. G. Dutrisac and D. B. Skillicorn, "Hiding Clusters in Adversarial Settings," in *Proceedings of IEEE International Conference on Intelligence and Security Informatics, ISI, Taipei, Taiwan*, 2008.

[6] B. Biggio, K. Rieck, D. Ariu, C. Wressnegger, I. Corona, G. Giacinto, and F. Roli, "Poisoning Behavioral Malware Clustering," in *Proceedings of the Workshop on Artificial Intelligent and Security, AISec, Scottsdale, USA*, 2014.

[7] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion Detection with Unlabeled Data Using Clustering," in *Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA)*, 2001.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, 2009.

[9] W. Xu, Y. Qi, and D. Evans, "Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers," in *Proceedings of 23rd Annual Network and Distributed System Security Symposium (NDSS), San Diego, USA*, 2016.

[10] J. D. Tygar, "Adversarial Machine Learning," *IEEE Internet Computing*, 2011.

[11] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, 2013.

[12] J. Newsome, B. Karp, and D. X. Song, "Paragraph: Thwarting Signature Learning by Training Maliciously," in *Proceedings of Recent Advances in Intrusion Detection RAID, Hamburg, Germany*, 2006.

[13] B. Biggio, B. Nelson, and P. Laskov, "Poisoning Attacks against Support Vector Machines," in *Proceedings of the 29th International Conference on Machine Learning (ICML), Edinburgh, Scotland*, 2012.

[14] B. Biggio, G. Fumera, and F. Roli, "Security Evaluation of Pattern Classifiers under Attack," *CoRR*, vol. abs/1709.00609, 2017.

[15] C. Smutz and A. Stavrou, "Malicious PDF Detection using Metadata and Structural Features," in *Proceedings of 28th Annual Computer Security Applications Conference (ACSAC), Orlando, USA*, 2012.

[16] S. Patton, W. Yurcik, and D. Doss, "An Achilles' Heel in Signature-Based IDS: Squealing False Positives in SNORT," in *Proceedings of Recent Advances in Intrusion Detection RAID*, 2001.

[17] W. Yurcik, "Controlling Intrusion Detection Systems by Generating False Positives: Squealing proof-of-concept," in *Proceedings of 27th Annual IEEE Conference on Local Computer Networks (LCN), Tampa, USA*, 2002.

[18] S. A. Crosby and D. S. Wallach, "Denial of Service via Algorithmic Complexity Attacks." in *USENIX Security Symposium*, 2003.

[19] E. Tsyrklevich, "Attacking Host Intrusion Prevention Systems," *Black Hat USA*, 2004.

[20] B. Hernacki, J. Bennett, and J. A. Hoagland, "An overview of network evasion methods," *Information Security Technical Report*, 2005.

[21] D. Mutz, C. Kruegel, W. Robertson, G. Vigna, and R. A. Kemmerer, "Reverse Engineering of Network Signatures," in *Proceedings of the AusCERT Asia Pacific Information Technology Security Conference, Gold Coast, Australia*, 2005.

[22] Y. Zhou, M. Kantarcioglu, B. M. Thuraisingham, and B. Xi, "Adversarial Support Vector Machine Learning," in *Proceedings of The 18th ACM International Conference on Knowledge Discovery and Data Mining (KDD), Beijing, China*, 2012.

[23] Y. Zhou and M. Kantarcioglu, "Modeling Adversarial Learning as Nested Stackelberg Games," in *Proceedings of Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD), Auckland, New Zealand*, 2016.

[24] G. Wang, T. Wang, H. Zheng, and B. Y. Zhao, "Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers," in *Proceedings of the 23rd USENIX Security Symposium, San Diego, USA*, 2014.

[25] M. Kloft and P. Laskov, "Online Anomaly Detection under Adversarial Impact," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Sardinia, Italy*, 2010.

[26] A. Zimek, E. Schubert, and H. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, 2012.

[27] S. Wold, K. Esbensen, and P. Geladi, "Principal Component Analysis," *Chemometrics and intelligent laboratory systems*, 1987.

[28] IBM Analytics, "Cplex Optimizer," https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer, (Accessed on 04/11/2018).

[29] The UCI KDD Archive, "KDD Cup 1999 Data," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, (Accessed on 04/11/2018).

[30] W. Lee and S. J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," *ACM Transactions on Information and System Security*, 2000.

[31] D. J. K. Jr and C. L. Shook, "The Application of Cluster Analysis in Strategic Management Research: an Analysis and Critique," *Strategic management journal*, pp. 441–458, 1996.