
Online Anomaly Detection under Adversarial Impact

Marius Kloft

University of California, Berkeley, USA

Pavel Laskov

University of Tübingen, Germany

Abstract

Security analysis of learning algorithms is gaining increasing importance, especially since they have become target of deliberate obstruction in certain applications. Some security-hardened algorithms have been previously proposed for supervised learning; however, very little is known about the behavior of anomaly detection methods in such scenarios. In this contribution, we analyze the performance of a particular method—online centroid anomaly detection—in the presence of adversarial noise. Our analysis addresses three key security-related issues: derivation of an optimal attack, analysis of its efficiency and constraints. Experimental evaluation carried out on real HTTP and exploit traces confirms the tightness of our theoretical bounds.

1 Introduction

One of the key assumptions of machine learning methods is that a phenomenon to be learned is “impartial”. The noise in observations may be high and little may be known about its distribution. However, it is tacitly assumed that the process from which observations are drawn does not obstruct a learning algorithm. This assumption does not necessarily hold for modern machine learning applications. Well-known examples of “misbehaving” applications are statistical spam filters (Lowd and Meek, 2005), intrusion detection systems (Fogla and Lee, 2006), and automatic signature generation (Newsome et al., 2006). Some recent work in machine learning (Dalvi et al., 2004; Dekel and Shamir, 2008; Globerson and Roweis, 2006; Teo et al., 2008) and computer security (Barreno et al., 2006) has therefore started to address adversarial learning scenarios.

Appearing in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. Copyright 2010 by the authors.

The majority of learning techniques proposed for adversarial scenarios has focused on supervised methods. In many applications, most notably in computer security, anomaly detection is of crucial importance, as it allows one to detect unusual events, e.g., previously unseen exploits. It is therefore essential to understand attacks against anomaly detection algorithms and potential countermeasures.

The only previous work pertaining to the security of anomaly detection is Nelson and Joseph (2006), who analyzed online centroid anomaly detection with an infinitely growing amount of training data. Their main result was surprisingly optimistic: it was shown that an attacker needs an exponentially large number of data to subvert a learning algorithm. The strength of this result, however, is based on an assumption that all data is memorized for the entire operation of an algorithm. Such assumption of an infinite training window is not realistic in practice, as the ability to adjust to non-stationarity of data is strongly desired for anomaly detection.

The main contribution of this paper is the analysis of online centroid anomaly detection with a *finite sliding window* of training data under a poisoning attack. We extend the main result of Nelson and Joseph (2006) into a general analytical framework addressing the following key security properties of learning algorithms:

1. What is the optimal policy for an attacker?
2. What gain does the attacker achieve using the optimal policy?
3. What constraints can be imposed on the attacker and how do they affect his gain?

By answering the first two questions one can explicitly evaluate an attack’s impact. This can be interpreted as a *quantification* (Laskov and Kloft, 2009) of security of a given learning algorithm, similar to the analysis of cryptographic algorithms.

Our analysis leads to interesting insights into the behavior of online centroid anomaly detection under a poisoning attack. We show that the base method

can be easily subverted by an attacker using a linear amount of data. On the other hand, an attack may become much more difficult if additional constraints are imposed on the attacker. We show that if only a certain percentage of training data can be controlled by an attacker, the attack’s progress can be strictly bounded from above, which implies that the attack *fails even with an arbitrarily large effort*. Such properties can be used for the design of constructive protection mechanisms for learning algorithms.

2 Learning and Attack Models

2.1 Centroid Anomaly Detection

Given a data set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the goal of anomaly detection is to determine whether an example \mathbf{x} is likely to originate from the same distribution as the set X . This can be done, e.g., by estimating a probability density function from the sample X and flag \mathbf{x} as anomalous if it lies in a region with low density. However, in anomaly detection, knowing the density in the entire space is superfluous, as we are only interested in deciding whether a specific point falls into a “sparsely populated” area. Hence, several direct methods have been proposed for anomaly detection, e.g., one-class SVM (Schölkopf et al., 2001), support vector data description (SVDD) (Tax and Duin, 1999) and density level set estimation (Tsybakov, 1997).

In the centroid anomaly detection, the Euclidean distance from the empirical mean of the data is used as a measure of anomaly:

$$f(\mathbf{x}) = \|\mathbf{x} - \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i\|.$$

If a hard decision is desired instead of a soft anomaly score, a data point is considered anomalous if its anomaly score exceeds a fixed threshold r . Although extremely simple, this method is quite popular in computer security, in particular in anomaly-based intrusion detection (Hofmeyr et al., 1998; Laskov et al., 2004; Rieck and Laskov, 2007), due to its low computational burden¹.

Centroid anomaly detection can be seen as a special case for the SVDD, with outlier fraction $\eta = 1$ and of the Parzen window density estimator with the Gaussian kernel function $k(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}\mathbf{x} \cdot \mathbf{y})$. Despite its straightforwardness, a centroid model can represent arbitrarily complex density level sets using a kernel mapping (Schölkopf and Smola, 2002).

¹With suitable parallelization for multicore architectures, the processing speed of over 3 Gbps can be attained.

2.2 Online Anomaly Detection

The majority of anomaly detection applications have to deal with non-stationary data. Hence the model of normality needs to be updated in the course of operation. Incorporation of new data points and the removal of irrelevant ones can be efficiently done by using on-line algorithms.

For the centroid anomaly detection, re-calculation of the center of mass is straightforward and requires $O(1)$ work. In the infinite training window case, the index n is incremented with the arrival of every new data point, and the update is computed as²

$$\mathbf{c}' = \left(1 - \frac{1}{n}\right) \mathbf{c} + \frac{1}{n} \mathbf{x}. \tag{1}$$

Notice that the storage of all training inputs is not necessary; however, the impact of new data points is decreasing with growing n .

In the case of a finite sliding training window, n remains constant, a previous example \mathbf{x}_i is replaced with a new one, and the update is performed as

$$\mathbf{c}' = \mathbf{c} + \frac{1}{n}(\mathbf{x} - \mathbf{x}_i). \tag{2}$$

Various strategies can be used to find the point \mathbf{x}_i to be removed from a working set, e.g.:

- (a) **random-out**: a randomly chosen point is removed.
- (b) **nearest-out**: a nearest-neighbor to the new point \mathbf{x} is removed.
- (c) **average-out**: the center of mass is removed. The new center of mass is recalculated as $\mathbf{c}' = \mathbf{c} + \frac{1}{n}(\mathbf{x} - \mathbf{c})$, which is equivalent to Eq. (1) with constant n .

The strategies (a) and (b) require the storage of all points in the working set, whereas the strategy (c) can be implemented by holding only the center of mass.

2.3 Poisoning Attack

The goal of a poisoning attack is to force an anomaly detection algorithm to accept an attack point \mathbf{A} that lies outside of the normal ball, i.e. $\|\mathbf{A} - \mathbf{c}\| > r$. It is assumed that an attacker knows the anomaly detection algorithm and all the training data. However, an attacker cannot modify any existing data except for

²The update formula can be generalized to $\mathbf{c}' = \mathbf{c} + \frac{\kappa}{n}(\mathbf{x} - \mathbf{x}_i)$, with fixed $\kappa \geq 1$. The bounds in the analysis change only by a constant factor, which is negligible.

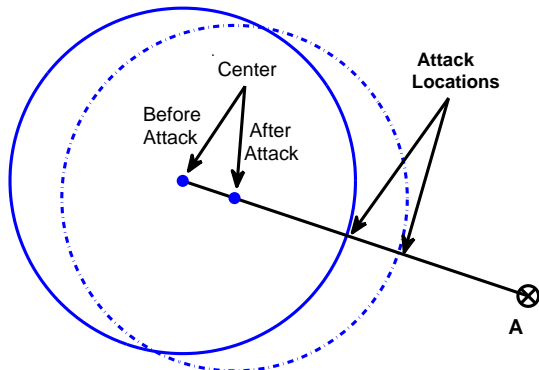


Figure 1: Illustration of a poisoning attack.

adding new points. These assumptions model a scenario in which an attacker can sniff data on the way to a particular host and can send his own data, while not having write access to that host. As illustrated in Fig. 1, the poisoning attack attempts to inject specially crafted points that are accepted as normal and push the center of mass in the direction of an attack point until the latter appears to be normal.

In order to quantify the effectiveness of a poisoning attack, we define the i -th relative displacement of the center of mass to be $D_i = \frac{(\mathbf{c}_i - \mathbf{c}_0) \cdot \mathbf{a}}{r}$, where $\mathbf{a} = \frac{\mathbf{A} - \mathbf{c}_0}{\|\mathbf{A} - \mathbf{c}_0\|}$ is the attack direction vector (w.l.o.g. we assume for the initial center $\mathbf{c}_0 = 0$). This quantity measures a relative length of the projection of \mathbf{c}_i onto \mathbf{a} in terms of the radius of a normal ball.

The effectiveness of a poisoning attack for an infinite training window has been analyzed in Nelson and Joseph (2006). We provide an alternative proof that follows the framework proposed in the introduction.

Theorem 1 *The i -th relative displacement D_i of the centroid learner with an infinite training window under the poisoning attack is bounded by*

$$D_i \leq \ln \left(1 + \frac{i}{n} \right), \quad (3)$$

where i is the number of attack points and n the number of initial training points.

Proof. We first determine an optimal attack strategy and then bound the attack progress.

(a) Let \mathbf{a} be an attack direction vector and let $\{\mathbf{a}_i | i \in \mathbb{N}\}$ be adversarial training points. The center of mass at the i -th iteration is given in the following recursion,

$$\mathbf{c}_{i+1} = \left(1 - \frac{1}{n+i} \right) \mathbf{c}_i + \frac{1}{n+i} \mathbf{a}_{i+1}, \quad (4)$$

with initial value $\mathbf{c}_0 = 0$. By the construction of the poisoning attack, $\|\mathbf{a}_i - \mathbf{c}_i\| \leq r$, which is equivalent

to $\mathbf{a}_i = \mathbf{c}_i + \mathbf{b}_i$ with $\|\mathbf{b}_i\| \leq r$. Eq. (4) can thus be transformed into

$$\mathbf{c}_{i+1} = \mathbf{c}_i + \frac{1}{n+i} \mathbf{b}_i.$$

Taking scalar product with \mathbf{a} and using the definition of a relative displacement, we obtain:

$$D_{i+1} = D_i + \frac{1}{n+i} \cdot \frac{\mathbf{b}_i \cdot \mathbf{a}}{r}, \quad (5)$$

with $D_0 = 0$. The right-hand side of the Eq. (5) is clearly maximized under $\|\mathbf{b}_i\| \leq r$ by setting $\mathbf{b}_i = r\mathbf{a}$. Thus the optimal attack is defined by

$$\mathbf{a}_i = \mathbf{c}_i + r\mathbf{a}. \quad (6)$$

(b) Plugging the optimal strategy $\mathbf{b}_i = r\mathbf{a}$ into Eq (5), we have:

$$D_{i+1} = D_i + \frac{1}{n+i}.$$

This recursion can be explicitly solved, taking into account that $d_0 = 0$, resulting in:

$$D_i = \sum_{k=1}^i \frac{1}{n+k} = \sum_{k=1}^{n+i} \frac{1}{k} - \sum_{k=1}^n \frac{1}{k}.$$

Inserting the upper bound on the harmonic series, $\sum_{k=1}^m \frac{1}{k} = \ln(m) + \epsilon_m$ into the above formula, and noting that the approximation error, ϵ_m , is monotonically decreasing, we obtain

$$D_i \leq \ln(n+i) - \ln(n) = \ln \left(\frac{n+i}{n} \right) = \ln \left(1 + \frac{i}{n} \right),$$

which completes the proof. \square

By inverting the bound in Theorem 1, it can be seen that the effort needed to subvert a centroid learner grows *exponentially* with respect to the separation of an attack point from the normal data.

To summarize the current understanding of poisoning attacks, an attacker needs to inject an exponential number of training data points in order to achieve a fixed displacement of the center in the desired direction. On the other hand, the assumption of an infinite window adversely affects the ability of anomaly detection to adjust to a change in the data distribution. Hence, it needs to be investigated whether the developed security guarantees hold for various update rule in the finite window case. This is the subject of the investigation in the following two sections.

3 Poisoning Attack with Full Control over the Training Data

We begin with the simpler case, when an attacker fully controls the training data, i.e. he can insert an arbitrary large number of points into a data stream.

3.1 Average-out and Random-out Learners

The average-out learner uses the same update rule as the infinite-window centroid learner except that the window size n remains fixed. Despite the similarity to the infinite-window case, the result presented in the following theorem is surprisingly pessimistic.

Theorem 2 *The i -th relative displacement D_i of the centroid learner with the average-out update rule under a worst-case optimal poisoning attack is*

$$D_i = \frac{i}{n}, \quad (7)$$

where i is the number of attack points and n is the training window size.

For the non-deterministic random-out rule, a stochastic analog can be shown: $E(D_i) = \frac{i}{n}$. One can see that, unlike the logarithmic bound in Theorem 1, the average-out and random-out learners are characterized by a linear bound on the relative displacement. As a result, an attacker only needs a linear amount of injected points in order to subvert an average-out learner. This cannot be considered secure.

The proof of Theorem 2 is straightforward and is omitted for brevity.

3.2 Nearest-out Learner

One might expect that the nearest-out strategy poses a stronger challenge to an attacker, as it tries to keep as much of the working set diversity as possible. It turns out, however, that even this strategy can be broken with a feasible amount of work if an attacker follows a greedy optimal strategy.

3.2.1 Greedy optimal attack

The goal of a greedy optimal attack is to find for each point \mathbf{x}_i the location of an attack point \mathbf{x}_i^* within a Voronoi cell induced by \mathbf{x}_i that maximizes the relative displacement. This can be formulated as the following optimization problem:

$$\{\mathbf{x}_i^*, f_i\} = \max_{\mathbf{x}} (\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{a} \quad (8.a)$$

$$\text{s.t. } \|\mathbf{x} - \mathbf{x}_i\| \leq \|\mathbf{x} - \mathbf{x}_j\|, \quad \forall j \quad (8.b)$$

$$\|\mathbf{x} - \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j\| \leq r. \quad (8.c)$$

The geometry of a greedy optimal attack is illustrated in Fig. 2.

The maximization of Eq. (8) over all points in a current working set yields the index of a point to be replaced by an attacker:

$$\alpha = \operatorname{argmax}_{i \in \{1, \dots, n\}} f_i \quad (9)$$

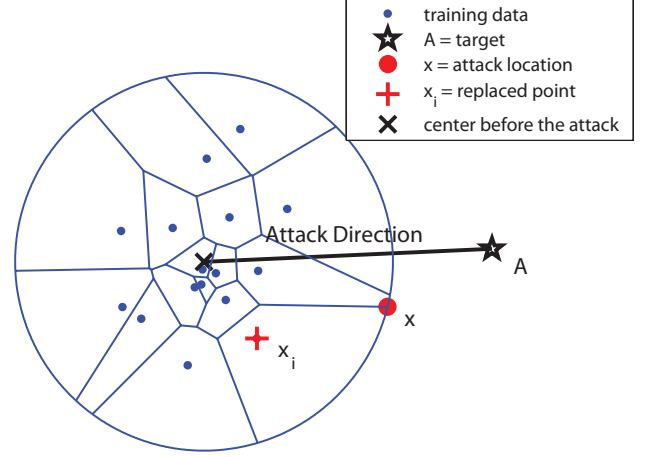


Figure 2: The geometry of a greedy optimal attack against a nearest-out learner. An adversary successfully displaces the centroid into direction of an attack point.

By plugging the definition of an Euclidean norm into the inner optimization problem (8), we obtain a simpler quadratically constrained linear problem (QCLP):

$$\max_{\mathbf{x}} (\mathbf{x} - \mathbf{x}_i) \cdot \mathbf{a} \quad (10.a)$$

$$\text{s.t. } 2(\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{x} \leq \mathbf{x}_j \cdot \mathbf{x}_j - \mathbf{x}_i \cdot \mathbf{x}_i, \quad \forall j \quad (10.b)$$

$$\|\mathbf{x} - \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j\|_2^2 \leq r^2. \quad (10.c)$$

Due to the quadratic constraint (10.c), the inner optimization task is not as simple as a linear or a quadratic program. However, several standard optimization packages, e.g., CPLEX or MOSEK, can handle such problems rather efficiently.

3.2.2 Attack effectiveness

An analytical estimate of the effectiveness of a greedy optimal attack is not possible since the optimal value of the objective function in the optimization problems (8) and (10) depends on the rather complex geometry of Voronoi cells. Hence, we experimentally investigate the behavior of the relative displacement D_i during the progress of a greedy optimal attack.

The experiment is performed according to the following protocol. An initial working set of size $n = 100$ is sampled from a d -dimensional Gaussian distribution with unit covariance. The radius r is chosen such that the expected false positive rate is bounded by $\alpha = 0.001$. An attack direction \mathbf{a} is chosen randomly with $\|\mathbf{a}\| = 1$, and 500 greedy optimal attack iterations are performed. The relative displacement of the center in the direction of the attack is measured at each iteration. Experiments are repeated for various values of d , and the results are averaged over 10 runs.

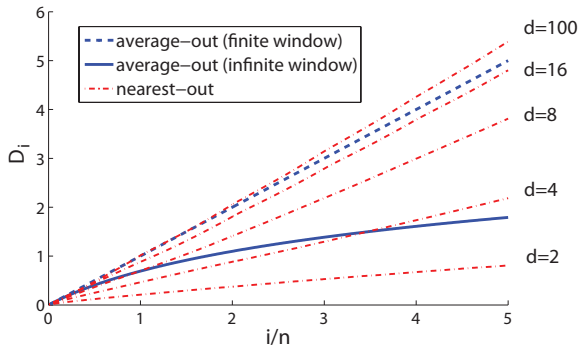


Figure 3: Empirical displacement of a poisoning attack on artificial data.

The relative displacement plots of a greedy optimal attack as a function of the number of attack iterations are shown in Fig. 3, for different values of the dimensionality d . The attack progress is clearly *linear*, despite the initial intuition that the nearest-out strategy might be more difficult to attack. For comparison, the plots for the infinite window and the average-out cases are shown. One can see that the slope of the linear progress rate of the nearest-out learner *increases with the dimensionality of the problem*. For small dimensions it is comparable in absolute terms with the infinite window case; for large dimensions it approaches the level of insecurity of the average-out learner. Such a behavior can be explained by the fast growth of a volume of Voronoi cells, which essentially contributes to the successful attack progress.

4 Poisoning Attack with Limited Control over the Training Data

We now proceed with investigation of a poisoning attack under some constraints imposed on an attacker. We assume that an attacker can only inject up to a fraction of ν of the training data. In security applications, such an assumption is natural, as it may be difficult for an attacker to surpass a certain amount of normal traffic. We restrict ourselves to the average-out learner, as it is the easiest one to analyze.

The initial learner is centered at a position \mathbf{X}_0 and has the radius r (w.l.o.g. assume $X_0 = 0$ and $r = 1$). At each iteration a new training point arrives which is either inserted by an adversary or is drawn independently from the distribution of normal points, and a new center of mass \mathbf{X}_i is calculated³. The mixing of normal and attack points is modeled by a Bernoulli random variable with the parameter ν . We assume

³To emphasize the probabilistic model used in this section, we denote the location of the center and the relative displacement by capital letters.

that the expectation of normal points ϵ_i coincides with the initial center of mass: $E(\epsilon_i) = \mathbf{X}_0$. Furthermore, we assume that all normal points are accepted by the learner, i.e. $\|\epsilon_i\| \leq r$.

The described probabilistic model is formalized by the following axiom.

Axiom 1 $\{B_i | i \in \mathbb{N}\}$ are independent Bernoulli random variables with parameter $\nu > 0$. ϵ_i are i.i.d. random variables in a reproducing kernel Hilbert space \mathcal{H} , drawn from a fixed but unknown distribution P_ϵ , satisfying $E(\epsilon_i) = \mathbf{0}$ and $\|\epsilon_i\| \leq r = 1$ for each i . B_i and ϵ_j are mutually independent for each i, j . $f : \mathcal{H} \rightarrow \mathcal{H}$ is an attack strategy satisfying $\|f(x) - x\| \leq r$. $\{X_i | i \in \mathbb{N}\}$ is a collection of random variables in \mathcal{H} such that $\mathbf{X}_0 = \mathbf{0}$ and

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \frac{1}{n} (B_i f(\mathbf{X}_i) + (1 - B_i) \epsilon_i - \mathbf{X}_i). \quad (11)$$

According to the above axiom an adversary's attack strategy is formalized by an *arbitrary* function f . This gives rise to a question which attack strategies are optimal in the sense that an attacker reaches his goal of concealing a predefined attack vector in a minimal number of iterations. Attack's progress is measured by projecting the current center of mass onto the attack direction vector:

$$D_i = \mathbf{X}_i \cdot \mathbf{a}. \quad (12)$$

The following result characterizes an optimal attack strategy for the model specified in Axiom 1.

Proposition 1 *Let \mathbf{a} be an attack vector and let C be a centroid learner. Then the optimal attack strategy f is given by*

$$f(X_i) := \mathbf{X}_i + \mathbf{a}. \quad (13)$$

Proof. Since by Axiom 1 we have $\|f(x) - x\| \leq r$, any valid attack strategy can be written as $f(x) = x + g(x)$, such that $\|g\| \leq r = 1$. It follows that

$$\begin{aligned} D_{i+1} &\leq \mathbf{X}_{i+1} \cdot \mathbf{a} \\ &= \left(\mathbf{X}_i + \frac{1}{n} (B_i f(\mathbf{X}_i) + (1 - B_i) \epsilon_i - \mathbf{X}_i) \right) \cdot \mathbf{a} \\ &= D_i + \frac{1}{n} (B_i D_i + B_i g(\mathbf{X}_i) \cdot \mathbf{a} + (1 - B_i) \epsilon_i \cdot \mathbf{a} - D_i), \end{aligned}$$

where we denote $\epsilon_i = \epsilon_i \cdot \mathbf{a}$. Since $B_i \geq 0$, the optimal attack strategy should maximize $g(\mathbf{X}_i) \cdot \mathbf{a}$ subject to $\|g(\mathbf{X}_i)\| \leq 1$. The maximum is clearly attained by setting $g(\mathbf{X}_i) = \mathbf{a}$. \square

The estimate of an optimal attack's effectiveness in the limited control case is given in the following theorem.

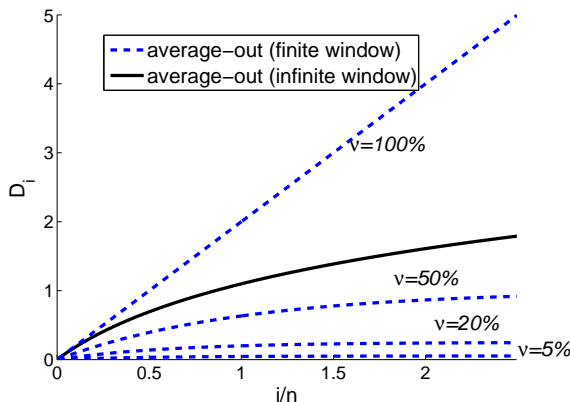


Figure 4: Theoretical displacement of a poisoning attack under limited control.

Theorem 3 *Let \mathcal{C} be a centroid learner under an optimal poisoning attack. Then, for the displacement D_i of \mathcal{C} , it holds:*

$$\begin{aligned} \text{(a)} \quad E(D_i) &= (1 - c_i) \frac{\nu}{1 - \nu} \\ \text{(b)} \quad \text{Var}(D_i) &\leq \gamma_i \left(\frac{\nu}{1 - \nu} \right)^2 + \delta_n \end{aligned}$$

where $\gamma_i = c_i - d_i$, $c_i := \left(1 - \frac{1-\nu}{n}\right)^i$, $d_i = \left(1 - \frac{1-\nu}{n} \left(2 - \frac{1}{n}\right)\right)^i$ and $\delta_n := \frac{\nu^2 + (1-d_i)}{(2n-1)(1-\nu)^2}$.

The proof of Theorem 3 is given in the supplemental material. The following corollary shows the asymptotic behavior of the above theorem.

Corollary 1 *Let \mathcal{C} be a centroid learner under an optimal poisoning attack. Then, for the displacement D_i of \mathcal{C} , it holds:*

$$\begin{aligned} \text{(a)} \quad E(D_i) &\leq \frac{\nu}{1 - \nu} \text{ for all } i \\ \text{(b)} \quad \text{Var}(D_i) &\rightarrow 0 \text{ for } i, n \rightarrow \infty. \end{aligned}$$

The growth of the above theorem’s bounds as a function of the number of attack iterations is illustrated in Fig. 4. One can see that an attack’s success strongly depends on the fraction of training data controlled by an attacker. For small ν , the attack’s progress is *bounded by a constant*, which implies that an attack fails even with an infinite effort. This result provides a *much stronger security guarantee* than the exponential bound for the infinite window case.

5 Experiments: Intrusion Detection

The experiments carried out in this section are intended to verify that the results of our analysis given

in the previous sections hold true for real data used in security applications, e.g., in intrusion detection (Hofmeyr et al., 1998; Wang and Stolfo, 2004; Wang et al., 2006; Rieck and Laskov, 2007). First, we present the data set and some preprocessing steps, followed by experiments under full or partial attacker’s control. Our experiments focus on the nearest-out method unless otherwise stated.

5.1 Data Corpus and Preprocessing

Our data is a sample of real HTTP traffic from a web server of Fraunhofer FIRST institute. We treat each inbound HTTP request as a byte string disregarding its specific, protocol dependent structure. Each request is considered to be a single data point. The benign dataset consists of 2950 byte strings of normal requests. The malicious dataset consists of 69 attack instances from 20 classes generated using the Metasploit penetration testing framework⁴.

As byte sequences are not directly suitable for the application of machine learning algorithms, we deploy a spectrum kernel (Leslie et al., 2002) for the computation of the inner products. The k -gram embedding with a fixed value $k = 3$ has been used. Kernel values have been normalized according to

$$k(\mathbf{x}, \bar{\mathbf{x}}) \mapsto \frac{k(\mathbf{x}, \bar{\mathbf{x}})}{\sqrt{k(\mathbf{x}, \mathbf{x})k(\bar{\mathbf{x}}, \bar{\mathbf{x}})}},$$

to avoid a dependence on the HTTP request length.⁵

5.2 Evaluation of a Full Control Attack

For computational reasons, several approximations must be made in solving the optimization problems for determining optimal attack points. As the dimensionality of the input space can be very large for embedded sequences, we use a reduced embedding space comprised of only 250 directions corresponding to leading principal components. Using the fact that the optimal attack lies in the span of training data, we construct the attack’s byte sequence by concatenating original sequences of basis points with rational coefficients that approximately match the coefficients of the linear combination.

To assess the validity of the bound *for various exploits*, we randomly choose an exploit and a normal training set and run a poisoning attack until success, i.e. until its vector is accepted by the anomaly detector. We record the resulting “displacement-at-success” and the

⁴<http://www.metasploit.com/>

⁵More details about the design of the intrusion detection experiments can be found in the technical report Kloft and Laskov (2010).

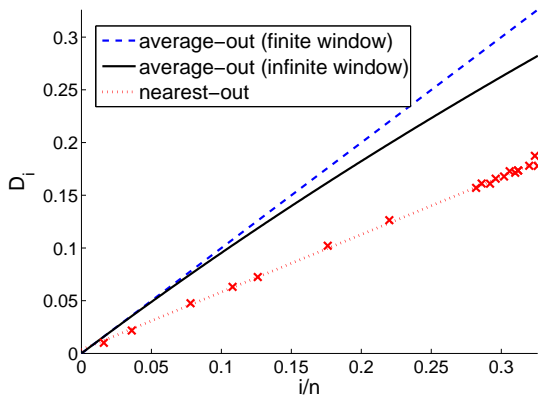


Figure 5: Empirical displacement of a poisoning attack on network intrusion detection data.

number of attack iterations required. The results are plotted for all 20 exploits. The resulting curve, c.f. Fig. 5, shows an interpolated growth rate across various exploit instances. It can be clearly seen that in practice, the growth level remains linear, although its slope is decreased roughly by 50% from numerous approximations needed for a practical attack realization. The practicality of a poisoning attack is emphasized by a small number of iterations needed for it to succeed: only up to 35% of the initial data points have to be overwritten.

5.3 Evaluation of a Limited Control Attack

The analysis of Section 4 reveals that limiting an attacker’s access to data imposes a hard upper bound on the attainable relative displacement. This implies that an attack will fail if the rate of poisoned data is less than a value ν_{crit} computed from Corollary 1 by solving bound (a) for ν .

To illustrate the accuracy of the critical values, we simulate the poisoning attack for a specific exploit (IIS WebDAV 5.0). Displacement values are recorded for a poisoning attack against the average-out learner for various values of ν . One can see from Fig. 6 that the attack succeeds for $\nu = 0.16$ but fails to reach the required relative displacement of $D_{\text{crit}} = 0.18$ for $\nu = 0.14$. The theoretically computed critical traffic ratio for this attack is 0.152. The experiment verifies the validity of ν_{crit} , implying that the derived bounds are surprisingly tight in practice.

For an exploit specific security analysis, we compute critical traffic rates for the particular real exploits as follows. A 1000-element training set is randomly drawn from the normal pool and a center of mass is calculated. We fix the radius such a false positive rate $\alpha = 0.001$ on the normal data is attained. For each

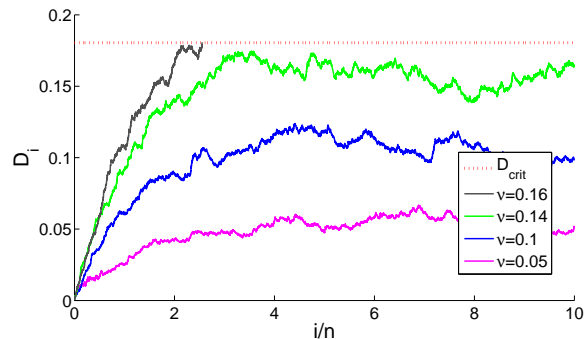


Figure 6: A simulation of a poisoning attack under limited control.

Attacks	Rel. dist.	ν_{crit}
ALT-N WebAdmin Overflow	0.058 ± 0.002	0.055
ApacheChunkedEncoding	0.176 ± 0.002	0.150
AWStats ConfigDir Execution	0.067 ± 0.002	0.063
Badblue Ext Overflow	0.168 ± 0.002	0.144
Barracuda Image Execution	0.073 ± 0.002	0.068
Edirectory Host	0.153 ± 0.002	0.132
IWebmail	0.178 ± 0.002	0.151
IIS 5.0 IDQ exploit	0.162 ± 0.002	0.140
Pajax Execute	0.107 ± 0.002	0.097
PEERCAST URL	0.163 ± 0.002	0.140
PHP Include	0.097 ± 0.002	0.088
PHP vBulletin	0.176 ± 0.002	0.150
PHP XML RPC	0.172 ± 0.002	0.147
HTTP tunnel	0.160 ± 0.002	0.138
IIS 4.0 HTR exploit	0.176 ± 0.002	0.149
IIS 5.0 printer exploit	0.161 ± 0.002	0.138
IIS unicode attack	0.153 ± 0.002	0.133
IIS w3who exploit	0.168 ± 0.002	0.144
IIS 5.0 WebDAV exploit	0.179 ± 0.002	0.152
rproxy exploit	0.155 ± 0.002	0.134

Table 1: Relative distances (in radii) of exploits to the boundary of a centroid enclosing all training points and critical values of parameter ν .

of the 20 attack classes, a class-wise median distance to the centroid’s boundary is computed. Using these distance values we calculate the “critical value” ν_{crit} . The results averaged over 10 repetitions are shown in Table 1. It can be seen that, depending on the exploit, an attacker needs to control 5–15% of traffic to successfully stage a poisoning attack. This could be a major limitation on sites with high traffic volumes.

6 Conclusions

The theoretical and experimental investigation of centroid anomaly detection in the presence of a poisoning attack carried out in this contribution points out the importance of security analysis for machine learning algorithms. Our analysis focuses on a specific—and admittedly quite simple—learning model, which is commonly employed in a particular computer security

application: network intrusion detection. We identified three key issues that are to be addressed by such a security analysis: understanding of an optimal attack, analysis of its efficiency and constraints. These criteria enable one to carry out a *quantitative security analysis* of learning algorithms. Furthermore, they provide a motivation for the design of anomaly detection algorithms with robustness to adversarial noise, similar to game-theoretic approaches recently proposed for supervised learning in adversarial environments, e.g., Dalvi et al. (2004); Dekel and Shamir (2008).

Our results suggest that online centroid anomaly detection with finite window cannot be considered secure without constraining the attacker’s access to the data. Also, we showed that a poisoning attack cannot succeed unless an attacker controls more than the critical traffic ratio, which can be a major hurdle. A potential *constructive* protection mechanism, omitted due to space limit,⁶ can be realized by controlling the false alarm rate of the anomaly detection algorithm.

Acknowledgments

This work was supported in part by the German BMBF grant REMIND (FKZ 01-IS07007A), by the European Community under the PASCAL2 Network of Excellence (ICT-216886), and scholarships by the German DAAD and DFG.

References

- M. Barreno, B. Nelson, R. Sears, A. Joseph, and J. Tygar. Can machine learning be secure? In *ACM Symposium on Information, Computer and Communication Security*, pages 16–25, 2006.
- N. Dalvi, P. Domingos, M. Sumit, and S. D. Verma. Adversarial classification. In *In KDD*, pages 99–108. ACM Press, 2004.
- O. Dekel and O. Shamir. Learning to classify with missing and corrupted features. In *International Conference on Machine Learning (ICML)*, pages 216–223, 2008.
- P. Fogla and W. Lee. Evading network anomaly detection systems: formal reasoning and practical techniques. In *ACM Conference on Computer and Communications Security*, pages 59–68, 2006.
- A. Globerson and S. Roweis. Nightmare at test time: Robust learning by feature deletion. In *International Conference on Machine Learning (ICML)*, pages 353–360, 2006.
- S. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.
- M. Kloft and P. Laskov. Security analysis of online centroid anomaly detection. Technical Report UCB/EECS-2010-22, EECS Department, University of California, Berkeley, Feb 2010. CoRR abs/1003.0079.
- P. Laskov and M. Kloft. A framework for quantitative security analysis of machine learning. In D. Balfanz and J. Staddon, editors, *AISec*, pages 1–4. ACM, 2009. ISBN 978-1-60558-781-3.
- P. Laskov, C. Schäfer, and I. Kotenko. Intrusion detection in unlabeled data with quarter-sphere support vector machines. In *Detection of Intrusions and Malware, and Vulnerability Assessment, Proc. of DIMVA Conference*, pages 71–82, 2004.
- C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: A string kernel for SVM protein classification. In *Proc. Pacific Symp. Biocomputing*, pages 564–575, 2002.
- D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 641–647, 2005.
- B. Nelson and A. D. Joseph. Bounding an attack’s complexity for a simple learning model. In *Proc. of the First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, Saint-Malo, France, 2006.
- J. Newsome, B. Karp, and D. Song. Paragraph: Thwarting signature learning by training maliciously. In *Recent Advances in Intrusion Detection (RAID)*, pages 81–105, 2006.
- K. Rieck and P. Laskov. Language models for detection of unknown attacks in network traffic. *Journal in Computer Virology*, 2(4):243–256, 2007.
- B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11–13):1191–1199, 1999.
- C. Teo, A. Globerson, S. Roweis, and A. Smola. Convex learning with invariances. In *Advances in Neural Information Processing Systems (NIPS)*, 2008. to appear.
- A. Tsybakov. On nonparametric estimation of density level sets. *Annals of Statistics*, 25:948–969, 1997.
- K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection (RAID)*, pages 203–222, 2004.
- K. Wang, J. Parekh, and S. Stolfo. Anagram: A content anomaly detector resistant to mimicry attack. In *Recent Advances in Intrusion Detection (RAID)*, pages 226–248, 2006.

⁶See technical report Kloft and Laskov (2010) for details.

Supplementary material

The following proposition is needed for the proof of Theorem 3.

Proposition 2 (Formula of the Geometric Series) *Let $(s)_{i \in \mathbb{N}_0}$ be a sequence of real numbers satisfying $s_0 = 0$ and $s_{i+1} = qs_i + p$ [or $s_{i+1} \leq qs_i + p$] for some $p, q > 0$. Then it holds:*

$$s_i = p \frac{1 - q^i}{1 - q}, \quad [\text{or } s_i \leq p \frac{1 - q^i}{1 - q}], \quad (14)$$

respectively.

Proof.

(a) We prove part (a) of the theorem by induction over $i \in \mathbb{N}_0$, the case of $i = 0$ being obvious.

In the inductive step we show that if Eq. (14) holds for an arbitrary fixed i it also holds for $i + 1$:

$$\begin{aligned} s_{i+1} &= qs_i + p = q \left(p \frac{1 - q^i}{1 - q} \right) + p = p \left(q \frac{1 - q^i}{1 - q} + 1 \right) \\ &= p \left(\frac{q - q^{i+1} + 1 - q}{1 - q} \right) = p \left(\frac{1 - q^{i+1}}{1 - q} \right). \end{aligned}$$

(b) The proof of part (b) is analogous. □

Proof of Theorem 3.

Proof.

(a) Inserting the optimal attack strategy of Prop. 1 into Eq. (11) of Ax. 1, we have:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \frac{1}{n} (B_i (\mathbf{X}_i + \mathbf{a}) + (1 - B_i) \boldsymbol{\epsilon}_i - \mathbf{X}_i),$$

which can be rewritten as:

$$\mathbf{X}_{i+1} = \left(1 - \frac{1 - B_i}{n} \right) \mathbf{X}_i + \frac{B_i}{n} \mathbf{a} + \frac{(1 - B_i)}{n} \boldsymbol{\epsilon}_i, \quad (15)$$

Taking the expectation on the latter equation, and noting that by Axiom 1 $E(\boldsymbol{\epsilon}) = 0$ and $E(B_i) = \nu$ holds, we have

$$\mathbf{E}(\mathbf{X}_{i+1}) = \left(1 - \frac{1 - \nu}{n} \right) \mathbf{E}(\mathbf{X}_i) + \frac{\nu}{n} \mathbf{a}.$$

Since by Eq. (12) we have $E(D_i) = E(\mathbf{X}_i) \cdot \mathbf{a}$ and $\|\mathbf{a}\| = R = 1$, we conclude

$$E(D_{i+1}) = \left(1 - \frac{1 - \nu}{n} \right) E(D_i) + \frac{\nu}{n}.$$

Now statement (a) follows by the formula of the geometric series, i.e. by Prop. 2, from the latter recursive Equation.

(b) Multiplying both sides of Eq.(15) with \mathbf{a} and substituting $D_i = \mathbf{X}_i \cdot \mathbf{a}$ results in

$$D_{i+1} = \left(1 - \frac{1 - B_i}{n} \right) D_i + \frac{B_i}{n} + \frac{(1 - B_i)}{n} \boldsymbol{\epsilon}_i \cdot \mathbf{a}.$$

Inserting $B_i^2 = B_i$ and $B_i(1 - B_i) = 0$, which holds because B_i is Bernoulli, into the latter equation, we have:

$$\begin{aligned} D_{i+1}^2 &= \left(1 - 2 \frac{1 - B_i}{n} + \frac{1 - B_i}{n^2} \right) D_i^2 + \frac{B_i}{n^2} + \frac{(1 - B_i)}{n^2} \|\boldsymbol{\epsilon}_i \cdot \mathbf{a}\|^2 \\ &\quad + 2 \frac{B_i}{n} D_i + 2(1 - B_i) \left(1 - \frac{1}{n} \right) D_i \boldsymbol{\epsilon}_i \cdot \mathbf{a}. \end{aligned}$$

Taking the expectation on the latter equation, and noting that by Axiom 1 $\boldsymbol{\epsilon}_i$ and \mathbf{D}_i are independent, we have:

$$\begin{aligned} E(D_{i+1}^2) &= \left(1 - \frac{1 - \nu}{n} \left(2 - \frac{1}{n} \right) \right) E(D_i^2) + 2 \frac{\nu}{n} E(D_i) + \frac{\nu}{n^2} \\ &\quad + \frac{1 - \nu}{n^2} E(\|\boldsymbol{\epsilon}_i \cdot \mathbf{a}\|^2) \\ &\stackrel{(*)}{\leq} \left(1 - \frac{1 - \nu}{n} \left(2 - \frac{1}{n} \right) \right) E(D_i^2) + 2 \frac{\nu}{n} E(D_i) + \frac{1}{n^2} \end{aligned}$$

where (*) holds because by Axiom 1 we have $\|\epsilon_i\|^2 \leq R$ and by definition $\|\mathbf{a}\| = R$, $R = 1$. Inserting the result of (a) in the latter equation results in the following recursive formula:

$$E(D_{i+1}^2) \leq \left(1 - \frac{1-\nu}{n} \left(2 - \frac{1}{n}\right)\right) E(D_i^2) + 2(1-c_i) \frac{\nu}{n} \frac{\nu}{1-\nu} + \frac{1}{n^2}.$$

By the formula of the geometric series, i.e. by Prop. 2, we have:

$$E(D_i^2) \leq \left(2(1-c_i) \frac{\nu}{n} \frac{\nu}{1-\nu} + \frac{1}{n^2}\right) \frac{1-d_i}{\frac{1-\nu}{n} \left(2 - \frac{1}{n}\right)},$$

denoting $d_i := \left(1 - \frac{1-\nu}{n} \left(2 - \frac{1}{n}\right)\right)^i$. Furthermore by some algebra

$$E(D_i^2) \leq \frac{(1-c_i)(1-d_i)}{1 - \frac{1}{2n}} \frac{\nu^2}{(1-\nu)^2} + \frac{1-d_i}{(2n-1)(1-\nu)}. \quad (16)$$

We will need the auxiliary formula

$$\frac{(1-c_i)(1-d_i)}{1 - \frac{1}{2n}} - (1-c_i)^2 \leq \frac{1}{2n-1} + c_i - d_i, \quad (17)$$

which can be verified by some more algebra and employing $d_i < c_i$. We finally conclude

$$\begin{aligned} \text{Var}(D_i) &= E(D_i^2) - (E(D_i))^2 \\ &\stackrel{\text{Th.3(a); Eq.(16)}}{\leq} \left(\frac{(1-c_i)(1-d_i)}{1 - \frac{1}{2n}} - (1-c_i)^2\right) \left(\frac{\nu}{1-\nu}\right)^2 \\ &\quad + \frac{1-d_i}{(2n-1)(1-\nu)^2} \\ &\stackrel{\text{Eq.(17)}}{\leq} \gamma_i \left(\frac{\nu}{1-\nu}\right)^2 + \delta_n \end{aligned}$$

where $\gamma_i := c_i - d_i$ and $\delta_n := \frac{\nu^2 + (1-d_i)}{(2n-1)(1-\nu)^2}$. This completes the proof the theorem. \square