Evaluation of Classifiers: Practical Considerations for Security Applications

Alvaro A. Cárdenas and **John S. Baras**
Department of Electrical and Computer Engineering
and The Institute for Systems Research
University of Maryland, College Park
{acardena,baras}@isr.umd.edu

Abstract

In recent years several tools based on statistical methods and machine learning have been incorporated in security related tasks involving classification, such as intrusion detection systems (IDSs), fraud detection, spam filters, biometrics and multimedia forensics. Measuring the security performance of these classifiers is an essential part for facilitating decision making, determining the viability of the product, or for comparing multiple classifiers. There are however relevant considerations for security related problems that are sometimes ignored by traditional evaluation schemes. In this paper we identify two pervasive problems in security-related applications. The first problem is the usually large class imbalance between normal events and attack events. This problem has been addressed by evaluating classifiers based on cost-sensitive metrics and with the introduction of Bayesian Receiver Operating Characteristic (B-ROC) curves. The second problem to consider is the fact that the classifier or learning rule will be deployed in an adversarial environment. This implies that good performance on average might not be a good performance measure, but rather we look for good performance under the worst type of adversarial attacks. In order to address this notion more precisely we provide a framework to model an adversary and define security notions based on evaluation metrics.

Introduction

The accepted paradigm for designing and evaluating machine learning algorithms is to induce a classifier from training data sets and then measure their classification performance in test data sets (usually performing some kind of cross-validation). The training data set consists on examples of one class (unsupervised learning) or both classes (supervised learning). The test data set usually consists of labeled examples with the known truth (labeled with the correct class). The classification performance is usually measured by its accuracy or in the case of cost sensitive classification or class imbalances, by the tradeoff between the false alarm rate and the detection rate (the ROC curve).

The computer security community has borrowed these evaluation methods when using statistical and machine learning tools for designing classifiers. There are however a large set of assumptions in this traditional paradigm that might not hold in practice for information security.

The first problem is that in practice, the large number of false alarms is one of the principal deterrents for the use of classifying techniques such as intrusion detection systems. The large number of false alarms is still a problem even when the evaluation methods have predicted what is typically considered small false alarm rates, such as 0.01 or 0.001. The problem is that in several security-related classification tasks, the number of normal events highly outnumbers the number of attack events. It is therefore important that designers specify clearly the *unit of analysis* and the expected likelihood of an attack, so that the evaluation can yield a better prediction and intuitive understanding of the real performance of the classifier.

The second problem is that traditional evaluation metrics are based on ideas mainly developed for non-security related fields and therefore, they do not take into account the role of an adversary. In practice, sophisticated attackers will try to react against any spam filter or intrusion detection technique implemented, trying to bypass or deteriorate the performance of the classifier. This problem has not been traditionally addressed by the statistical learning community, *after all, the Reuters-21578 data set never tried to evade your classifier*[1].

In the first part of this paper we summarize some of the metrics used in the intrusion detection community in order to deal with the class imbalance problem, and in the second part of the paper we present a set of guidelines and discuss examples of robust evaluation against adaptive attackers.

Evaluation Under Class Imbalances

The term *class imbalance* refers to the case when in a classification task, there are many more instances of some classes than others. The *problem* is that under this setting, classifiers in general perform poorly because they tend to concentrate on the large classes and disregard the ones with few examples.

Given the importance of the class imbalance problem in intrusion detection systems, several researches have proposed metrics that take into account the very few instances

[1] From http://taint.org

State of the system	Detector's report	
	No Alarm (A=0)	Alarm (A=1)
No Intrusion ($C=0$)	$L(0,0)$	$L(0,1)$
Intrusion ($C=1$)	$L(1,0)$	$L(1,1)$

Table 1: Loss function

of attacks. In this section we briefly summarize the metrics proposed for the evaluation of IDSs.

Before we present our formulation we need to introduce some notation and definitions. Assume that the input to the classifier is a feature-vector $\mathbf{x}$. Let C be an indicator random variable denoting whether $\mathbf{x}$ belongs to class zero: $C=0$ (the majority class) or class one: $C=1$ (the minority class). The output of the classifier is denoted by $A=1$ (or simply A) if the classifier assigns $\mathbf{x}$ to class one, and $A=0$ (or alternatively $\neg A$) if the classifier assigns $\mathbf{x}$ to class zero. With this notion we can define the *probability of false alarm* $P_{FA} \equiv \Pr[A=1|C=0]$ and the *probability of detection* $P_D \equiv \Pr[A=1|C=1]$. Finally, the class imbalance problem is quantified by the probability of a positive example $p=\Pr[C=1]$.

Expected Cost

The most traditional way of dealing with class imbalances under a single metric is to use Bayesian decision theory, an approach presented in (Gaffney & Ulvila 2001). In this method, a quantitative measure of the consequences of the output of the IDS to a given event are the costs shown in Table 1.

The *expected cost* (or Bayes risk) is defined as $\mathbf{E}[L(C,A)]$. The basic idea to deal with the class imbalance in this setting is to set the cost of missing an attack instance much higher than the cost of raising a false alarm.

The Intrusion Detection Capability

The main motivation for introducing the *intrusion detection capability* C_{ID} as an evaluation metric originates from the fact that the costs in Table 1 are chosen in a subjective way (Gu *et al.* 2006). Therefore the authors propose the use of the intrusion detection capability as an objective metric motivated by information theory:

$$C_{ID} = \frac{\mathbf{I}(C;A)}{\mathbf{H}(C)}$$

where $\mathbf{I}$ and $\mathbf{H}$ respectively denote the mutual information and the entropy. The $\mathbf{H}(C)$ term in the denominator is a normalizing factor so that the value of C_{ID} is always in the $[0,1]$ interval. C_{ID} can be interpreted as an instance of the expected cost problem with costs given by $L(i,j) = -\log\Pr[C=i|A=j]$. Note however that the costs in this case are not constant and depend on p, P_{FA} and P_D.

B-ROC curves

There are instances where the tradeoff of the objectives is a qualitative judgment best left to the user. In this case we need to consider the tradeoff between the parameters of interest. We therefore investigate which parameters are better suited to consider as trade-offs.

Of interest to the intrusion detection community, is that classifiers with ROC curves achieving traditionally "good" operating points such as $(P_{FA}=0.01, P_D=1)$ would still generate a huge amount of false alarms in realistic scenarios. This effect is due in part to the class imbalance problem, which is the cause of the base-rate fallacy (Axelsson 1999). In order to understand this problem, we now use two more metrics. The *positive predictive value* (or *precision*) $PPV \equiv \Pr[C=1|A=1]$, and the *negative predictive value* $NPV \equiv \Pr[C=0|A=0]$.

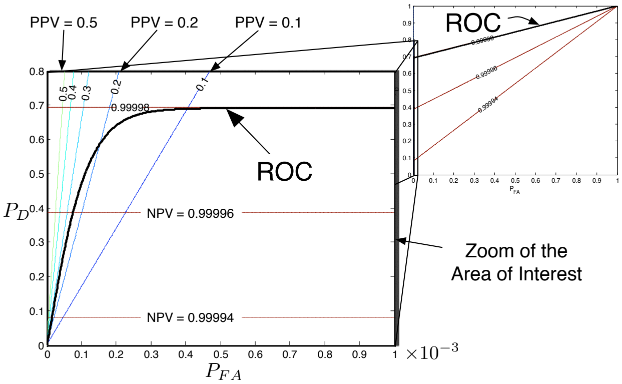

Figure 1: PPV and NPV isolines for the ROC of a typical IDS

In Figure 1 we can see the tradeoff of the four variables of interest: P_{FA} in the x-axis, P_D in the y-axis, PPV as the diagonal isolines, and NPV as the horizontal isolines. Notice that if we choose the optimal operating point based in P_{FA} and P_D, as in the typical ROC analysis, we might obtain misleading results because we do not know how to interpret intuitively very low false alarm rates, e.g. is $P_{FA}=10^{-3}$ much better than $P_{FA}=5\times10^{-3}$? The same reasoning applies to the study of PPV vs. NPV as we cannot interpret precisely small variations in NPV values, e.g. is $NPV=0.9998$ much better than $NPV=0.99975$? Therefore we conclude that the most relevant metrics to use for a tradeoff in the performance of a classifier in heavily imbalanced data sets are P_D and PPV, since they have an easily understandable range of interest.

However, even when you select as tradeoff parameters the PPV and P_D values, the isoline analysis shown in Figure 1 has still one deficiency, and it is the fact that there is no efficient way to account for the uncertainty of p. In order to solve this problem we introduce the B-ROC as a graph that shows how the two variables of interest: P_D and PPV are related under different severity of class imbalances. In order to follow the intuition of the ROC curves, instead of using PPV for the x-axis we prefer to use $1-PPV$. We use this quantity because it can be interpreted as the *Bayesian false alarm rate*: $B_{FA} \equiv \Pr[C=0|A=1]$. For example, for IDSs B_{FA} can be a measure of how likely it is, that the operators

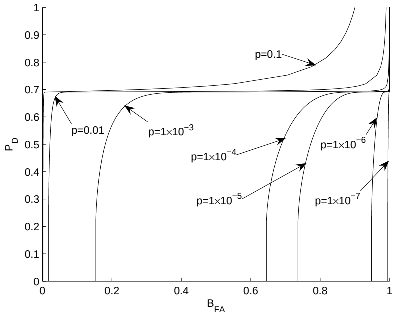

Figure 2: B-ROC for the ROC of Figure 1.

of the detection system will loose their time each time they respond to an alarm. Figure 2 shows the B-ROC for the ROC presented in Figure 1. More details and properties of B-ROC curves can be found in (Cárdenas & Baras 2006).

Towards Secure Classification

In this section we focus on designing a practical methodology for dealing with attackers. In particular we propose the use of a framework where each of these components is clearly defined:

Desired Properties Intuitive definition of the goal of the system.

Feasible Design Space The design space $\mathcal{S}$ for the classification algorithm.

Information Available to the Adversary Identify which pieces of information can be available to an attacker.

Capabilities of the Adversary Define a feasible class of attackers $\mathcal{F}$ based on the assumed capabilities.

Evaluation Metric The evaluation metric should be a reasonable measure how well the designed system meets our desired properties. We call a system *secure* if its metric outcome is satisfied for any feasible attacker.

Goal of the Adversary An attacker can use its capabilities and the information available in order to perform two main classes of attacks:

- *Evaluation attack.* The goal of the attacker is opposite to the goal defined by the evaluation metric. For example, if the goal of the classifier is to minimize $\mathbf{E}[L(C,A)]$, then the goal of the attacker is to maximize $E[L(C,A)]$.
- *Base system attack.* The goal of an attacker is not the opposite goal of the classifier. For example, even if the goal of the classifier is to minimize $E[L(C,A)]$, the goal of the attacker is still to minimize the probability of being detected.

Model Assumptions Identify clearly the assumptions made during the design and evaluation of the classifier. It is important to realize that when we borrow tools from other fields, they come with a set of assumptions that might not hold in an adversarial setting, because the first thing that an attacker will do is violate the set of assumptions that the classifier is relying on for proper operation. Therefore one of the most important ways to deal with an adversarial environment is to limit the number of assumptions made, and to evaluate the resiliency of the remaining assumptions to attacks.

We now present four examples to illustrate the applicability of the guidelines.

Example 1: Secret key Encryption

Cryptography is one of the best examples in which a precise framework has been developed in order to define properly what a secure system means, and how to model an adversary. Therefore, before we use the guidelines for the evaluation of classifiers, we describe a very simple example in cryptography. We believe this example clearly identifies the generality and use of the guidelines as a step towards achieving sound designs[2].

In secret key cryptography, Alice and Bob share a single key *sk*. Given a message *m* (called *plaintext*) Alice uses an encryption algorithm to produce unintelligible data *C* (called *ciphertext*): $C \leftarrow \mathcal{E}_{sk}(m)$. After receiving *C*, Bob then uses *sk* and a decryption algorithm to recover the secret message $m = \mathcal{D}_{sk}(C)$.

Desired Properties $\mathcal{E}$ and $\mathcal{D}$ should enable Alice and Bob to communicate secretly, that is, a feasible adversary should not get any information about *m* given *C* except with very small probability.

Feasible Design Space $\mathcal{E}$ and $\mathcal{D}$ have to be efficient probabilistic algorithms. They also need to satisfy correctness: for any *sk* and *m*, $\mathcal{D}_{sk}(\mathcal{E}_{sk}(m)) = m$.

Information Available to the Adversary It is assumed that an adversary knows the encryption and decryption algorithms. The only information not available to the adversary is the secret key *sk* shared between Alice and Bob.

Capabilities of the Adversary The class of feasible adversaries $\mathcal{F}$ is the set of algorithms running in a reasonable amount of time.

Evaluation Metric For any messages m_0 and m_1, given a ciphertext *C* which is known to be an encryption of either m_0 or m_1, no adversary $\mathcal{A} \in \mathcal{F}$ can guess correctly which message was encrypted with probability significantly better than $1/2$.

Goal of the Adversary Perform an evaluation attack. That is, design an algorithm $\mathcal{A} \in \mathcal{F}$ that can guess with probability significantly better than $1/2$ which message corresponds to the given ciphertext.

Model Assumptions The security of an encryption scheme usually relies in a set of cryptographic primitives, such as one way functions.

[2] We avoid the precise formal treatment of cryptography because our main objective here is to present the intuition behind the principles rather than the specific technical details.

Another interesting aspect of cryptography is the different notions of security when the adversary is modified. In the previous example it is sometimes reasonable to assume that the attacker will obtain valid plaintext and ciphertext pairs: $\{(m_0,C_0),(m_1,C_1),\ldots,(m_k,C_k)\}$. This new setting is modeled by giving the adversary more capabilities: the feasible set $\mathcal{F}$ will now consist of all efficient algorithms that have access to the ciphertexts of chosen plaintexts. An encryption algorithm is therefore secure against *chosen-ciphertext* attacks if even with this new capability, the adversary still cannot break the encryption scheme.

Example 2: Adversary with Control of the Base-Rate p

In this example we introduce probably one of the easiest formulations of an attacker against a classifier: we assume that the attacker cannot change its feature vectors **x**, but rather only its frequency of attacks: p.

We consider an intrusion detection system that monitors audit logs and has a false alarm rate of $\hat{P}_{FA}$ and a detection rate of $\hat{P}_D$. Assume now that the operator of the IDS has to decide whether to investigate audit logs or not based on the alarms reported by the IDS. The following table represents the possible decisions of the operator:

$$\begin{array}{ll} h_1(\neg A)=0 & h_1(A)=0 \\ h_2(\neg A)=1 & h_2(A)=0 \\ h_3(\neg A)=0 & h_3(A)=1 \\ h_4(\neg A)=1 & h_4(A)=1 \end{array}$$

For example h_1 represents the case when the operator does not check audit logs, and h_3 represents the case when the operator checks the audit log if and only if there is an alarm. Assume the operator decides on h_i with probability π_i.

Desired Properties Assume the operator wants to find a strategy that minimizes the probability of making errors. This is an example of the expected cost metric function with $L(0,0)=L(1,1)=0$ and $L(1,0)=L(0,1)=1$.

Feasible Design Space $\mathcal{S}=\{\pi_i\in[0,1]:\pi_1+\pi_2+\pi_3+\pi_4=1\}$.

Information Available to the Adversary We assume the adversary knows everything that we know and can make inferences about the situation the same way as we can. In game theory this adversaries are usually referred to as *intelligent*.

Capabilities of the Adversary The adversary has complete control over the base-rate p (its frequency of attacks). The feasible set is therefore $\mathcal{F}=[0,1]$.

Goal of the Adversary Evaluation attack.

Evaluation Metric

$$r^* = \min_{\pi_i\in\mathcal{S}} \max_{p\in\mathcal{F}} \mathbf{E}[L(C,A)]$$

Note the order in optimization of the evaluation metric. In this case we are assuming that the operator of the IDS makes the first decision, and that this information is then available to the attacker when selecting the optimal p. We call the strategy of the operator *secure* if the expected cost (probability of error) is never greater than r^* for any feasible adversary.

Model Assumptions We have assumed that the attacker will not be able to change $\hat{P}_{FA}$ and $\hat{P}_D$. This results from its assumed inability to directly modify the feature vector **x** (the security logs in this case).

The solution to this problem is easy to find once we identify it as a zero-sum game between the IDS operator and the attacker.

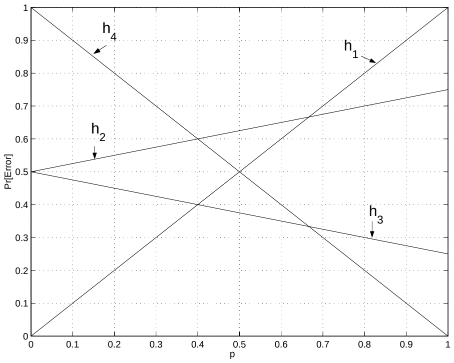

Figure 3: Probability of error for h_i vs. p

Assuming $\hat{P}_{FA}=0.5$ and $\hat{P}_D=0.75$, the strategy that achieves a Nash equilibrium is for the operator of the IDS to select $\pi_1^*=1/5$ and $\pi_3^*=4/5$, and for the attacker to select $p^*=2/5$ (see Figure 3). Therefore, after fixing the detection strategy, it does not matter if an attacker uses a p different from p^* because in the evaluation it is guaranteed that the probability of error will be no worse than $2/5$. So the evaluation estimating a probability of error of $2/5$ is secure (Cárdenas, Baras, & Seamon 2006).

It is important to note also that Figure 3 is a graph traditionally used in detection theory and game theory, and which has gained recent popularity in the machine learning community under the name of *cost curves* (Drummond & Holte 2001).

Example 3: Adversary with Partial Control Over the Observed Features

In the previous example the evaluation follows the traditional paradigm in which we assume the classifier is evaluated in data containing both normal and attack examples. We now turn to a more difficult problem: how to evaluate a classifier without attack data.

In this example the adversary is allowed to be more powerful by being able to modify the feature vector **x** observed in an attack. To illustrate an approach on how to formulate this problem we now turn our attention to watermarking (data hiding) schemes for multimedia data. In this scenario we assume we have a signal **s** (e.g., an image or an audio signal) to be marked. Assume that a classifier has learnt the signal distribution $f(\mathbf{s})$ via one class learning (recognition based learning).

Assume also there are two sources of content. Source 0 produces content without watermarks: $\mathbf{y} = \mathbf{s}$, and source 1 produces content with random watermarks $\mathbf{y} = \mathbf{s} + \mathbf{w}$, where $\mathbf{w}$ is assumed to have distribution $f_w(\mathbf{w})$. Furthermore assume that an attacker will modify $\mathbf{y}$ in order to produce the final observable $\mathbf{x}$ to the classifier. Note that the classifier does not have direct access to $\mathbf{s}$ or $\mathbf{x}$, it only has knowledge of their probability distributions (this is known as blind watermarking).

Desired Properties The classifier has to determine if a given $\mathbf{x}$ has a watermark or not.

Feasible Design Space The watermark should not distort the signal too much (the watermark should be invisible or inaudible for a human). Therefore the encoder of the watermark has to design $f_w(\mathbf{w})$ such that the average distortion is bounded by a given D_w. $\mathcal{S}$ is then the set of all pdf's f_w such that $\mathbf{E}[d(\mathbf{S}, \mathbf{S} + \mathbf{W})] \leq D_w$.

Information Available to the Adversary The only information not available to the adversary is the particular realizations of $\mathbf{S}$ and $\mathbf{W}$. So given a watermarked signal $\mathbf{y}$, the adversary does not know $\mathbf{s}$ or $\mathbf{w}$. Then again, this realizations are also not known to the classifier.

Capabilities of the Adversary As in the design space, an adversary also has a distortion constraint. Therefore $\mathcal{F}$ is the set of all conditional pdf's $f_{x|y}(\mathbf{x}|\mathbf{y})$ such that $\mathbf{E}[d(\mathbf{Y}, \mathbf{X})] \leq D_a$.

Goal of the Adversary Evaluation attack.

Evaluation Metric Minimize the probability of error for the worst type of feasible attacks. Again this is just the expected cost metric with $L(0,0) = L(1,1) = 0$ and $L(1,0) = L(0,1) = 1$.

$$r^* = \min_{f_w \in \mathcal{S}} \quad \max_{f_{x|y} \in \mathcal{F}} \quad \mathbf{E}[L(C,A)]$$

We call the strategy of the encoder *secure* if the probability of error is never greater than r^* for any feasible adversary.

Model Assumptions The robustness of the evaluation depends on how close $f(\mathbf{s})$ follows the real distribution of the source signal. If the real distribution of $\mathbf{s}$ changes or is not close enough to the learnt model $f(\mathbf{s})$, then the error in practice will be higher.

Finding optimal solutions for the above mentioned problem is sometimes intractable, and therefore some more simplifications are usually made. For example the attack distribution $f_{x|y}$ is most of the times assumed to be memoryless (i.e., $f_{x|y}(\mathbf{x}|\mathbf{y}) = \prod f(x_i|y_i)$). A survey of the analytical treatment of watermarking schemes can be found in (Moulin & Koetter 2005).

Example 4: Attacker With Complete Control of the Attack Distribution

An alternative view for finding least favorable attack distributions can be seen in the problem of detecting misbehavior in the MAC layer of wireless ad hoc networks (Radosavac, Baras, & Koutsopoulos 2005; Radosavac *et al.* 2006). In this case the feature vector $x_1, x_2, \ldots, x_n$ is a collection of back off times from a given node. We know the distribution of the normal instances (the specified back off protocol) and the problem is to find the optimal attack distribution that maximizes the number of times a selfish node accesses the MAC layer without being detected.

The pdf $f_0(x)$ of the normal behavior is assumed to be known or alternatively assumed that it can be learnt via one-class learning, however since the attacker can change its strategy, we cannot trust a machine learning technique to estimate $f_1(x)$ because the attacker can modify arbitrarily its strategy during the test period.

Desired Properties Detect misbehaving nodes as soon as possible at an acceptable false alarm rate.

Feasible Design Space $\mathcal{S}$ is defined to be any sequential test that satisfies a given false alarm and detection rates. A sequential test is an algorithm which with every new sample obtained x_i, either decides to classify based on $x_1, \ldots, x_i$ or waits for the next sample.

Information Available to the Adversary The adversary knows everything that we know and can make inferences about the situation the same way as we can.

Capabilities of the Adversary The adversary has control over its back-off distribution. The distribution however is assumed to be memoryless (the samples are i.i.d.). The feasible set of attacks is parameterized with parameter η (a measure of how aggressive the attacker is):

$$\mathcal{F}_\eta = \left\{ f_1 : \int_0^W \left(1 - \frac{x}{W}\right)^n f_1(x)\,dx \geq \eta \frac{1}{n+1} \right\}.$$

Goal of the Adversary Evaluation attack.

Evaluation Metric Obtain a saddle point for the amount of samples taken before reaching a decision:

$$\phi(h^*, f_1) \leq \phi(h^*, f_1^*) \leq \phi(h, f_1^*);\ \forall h \in \mathcal{S},\ \forall f_1 \in \mathcal{F}_\eta.$$

where $\phi(h, f)$ is the expected value of the number of samples collected before reaching a decision. $\phi(h^*, f_1^*)$ is *secure* if the expected time before reaching a decision is never greater for any feasible adversary.

Model Assumptions The attacker was constrained to be i.i.d. Furthermore, depending on the network, it might be difficult to model f_0 accurately: even though we know the back off specification for the MAC layer, the exponential back off mechanism and noisy observations can be a problem in practice.

On the Goal of the Adversary

So far all the examples we have considered assume only evaluation attacks, and all the evaluation attacks we presented can in fact be seen as zero sum games between the designer of the classifier h and the attacker.

Base system attacks on the other hand can be considered as nonzero sum games between the designer and the attacker, since the adversary will have a different objective function. These attacks are intuitively appealing in many classification scenarios. For example the primary goal of a

	Advantage	Disadvantage
Evaluation Attacks	More robust against modeling errors	Pessimistic evaluation: might be too restrictive
Base System Attacks	Can model more realistic attackers	Makes extra assumptions that might not hold in practice

Table 2: Goal of the adversary

spammer is to get the spam e-mails past spam filters, while the goal of the filter is to detect the spam messages and also maintain a low false alarm rate. Furthermore the utility of the spammer by getting an e-mail past the filter might be different to the cost the filter incurs by letting the message through. A formulation of base system attacks in spam filters can be found in (Dalvi *et al.* 2004).

By contrast, assuming that the attacker does not attack directly the evaluation of the system has some disadvantages. In particular we note that if the attacker deviates from the proposed goal (the adversary is not a rational player) or if the attacker has a different objective than the one it was assumed, the classifier can perform worse than what was determined by the evaluation.

However, the metric against evaluation attacks gives a lower bound on the performance of the classifier. If in real life, the attacker has another goal, or it deviates from the objective, the performance of the classifier will not be worse than the estimated value.

Table 2 summarizes the advantages and disadvantages of the different types of attacks considered.

Conclusions

The topics we have presented are just a small sample of a wide area of problems that need consideration. For example, obtaining optimal solutions to adversarial problems is often intractable. Therefore a compromise must be achieved sometimes between the accuracy of the model and the efficiency of solving the optimization problems.

It is also important to realize that the classifier is just a component of a larger system that has to be evaluated as a whole. For example, an IDS at the MAC layer of a communications networks should be evaluated with respect to the overall performance of the network. We can tolerate misbehavior as long as the network performance is not impacted.

A complimentary approach to the fully technical evaluation is that of using financial metrics. Several companies currently use metrics such as the return of investment, net present value and the internal rate of return for their security investments.

We should also point out that all our secure evaluations were done with evaluation metrics that return a single value. Therefore another important problem is how to define new meaningful performance tradeoffs and how to evaluate the security of classification with performance curves.

Finally, how to model accurately an adversary and its capabilities is a very important field. We for example did not considered the problem of how to train classifiers when the adversary can create errors in the labels of the training data set (Barreno *et al.* 2006).

Acknowledgements

This material is based upon work supported by the U.S. Army Research Office under Award No. DAAD19-01-1-0494 to the University of Maryland at College Park.

References

Axelsson, S. 1999. The base-rate fallacy and its implications for the difficulty of intrusion detection. In *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS '99),* 1–7.

Barreno, M.; Nelson, B.; Sears, R.; Joseph, A. D.; and Tygar, J. D. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security ASIACCS 06*, 16–25.

Cárdenas, A. A., and Baras, J. S. 2006. B-ROC curves for the assesment of classifiers over imbalanced data sets. In *Proceedings of the twenty-first National Conference on Artificial Intelligence, (AAAI 06) Nectar track.*

Cárdenas, A. A.; Baras, J. S.; and Seamon, K. 2006. A framework for the evaluation of intrusion detection systems. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*.

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *tenth ACM SIGKDD international conference on Knowledge discovery and data mining,* 99–108.

Drummond, C., and Holte, R. 2001. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 198–207.

Gaffney, J. E., and Ulvila, J. W. 2001. Evaluation of intrusion detectors: A decision theory approach. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, 50–61.

Gu, G.; Fogla, P.; Dagon, D.; Lee, W.; and Skoric, B. 2006. Measuring intrusion detection capability: An information-theoretic approach. In *Proceedings of ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS '06).*

Moulin, P., and Koetter, R. 2005. Data-hiding codes. *Proceedings of the IEEE* 93(12):2083–2126.

Radosavac, S.; Baras, J. S.; and Koutsopoulos, I. 2005. A framework for mac protocol misbehavior detection in wireless networks. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security,* 33–42.

Radosavac, S.; Cárdenas, A.; Baras, J. S.; and Moustakides, G. 2006. Detection of greedy individual and colluding mac layer attackers. Technical Report TR 2006-8, The Institute for Systems Research, University of Maryland.