www.ijiarec.com

ISSN: 2348-2079

MAY-2015

International Journal of Intellectual Advancements and Research in Engineering Computations

CONTROLLING PATTERN ATTACKS AGAINST ANOMALY BASED CLASSIFICATION

[1]S. Kalpana, [2]V. Saravanakumar.

ABSTRACT

Integrity violation, availability violation and privacy violation are the major intrusion caused in the network environment. An integrity violation, if it allows the adversary to access the service or resource protected by the classifier. An availability violation, if it denies legitimate users access to it. A privacy violation, if it allows the adversary to obtain confidential information from the classifier. Pattern classification techniques are extended with adversarial settings. Secure pattern classifiers are designed to control the performance degradation under potential attacks. Secure pattern classifier framework is build with model selection and training and testing data construction method. TR and TS construction algorithm is used to select data for pattern classifier. Secure pattern classifier is improved with attack control mechanism to handle the intruders in the testing process. Training patterns are secured with simulated attack patterns. Pattern update process is monitored and controlled in testing process. Classifier utility rate is improved in testing process.

INTRODUCTION

Machine learning techniques are rapidly emerging as a vital tool in a variety of networking and large-scale system applications because they can infer hidden patterns in large complicated datasets, adapt to new behaviors and provide statistical soundness to decision making processes. Application developers thus can employ learning to help solve so-called big-data problems and these include a number of security-related problems particularly focusing on identifying malicious or irregular behavior. In fact, learning approaches have already been used or proposed as solutions to a number of such security-sensitive tasks including spam, worm, intrusion and fraud detection. Unfortunately, in these domains, data is generally not only non-stationary but may also have an adversarial component and the exibility a forded by learning techniques can be exploited by an adversary to achieve his goals. For instance, in spam-detection, adversaries regularly adapt their approaches based on the popular spam detectors and generally a clever adversary will change his behavior either to evade or mislead learning.

Adversarial classification tasks like spam filtering, intrusion detection in computer networks and biometric identity verification have been typically faced as two-class classification problems, in which a classifier aims to discriminate between “malicious” and “legitimate” samples. Malicious samples are generated by an intelligent and adaptive adversary who can manipulate them to mislead the classifier [1]. For example, a well-known attack against spam filters consists in modifying a spam email by inserting words which are likely to appear in legitimate emails but not in spam and by obfuscating typical “spammy” words. These attacks are respectively referred to as good word insertion (GWI) and bad word obfuscation (BWO) [2]. For another instance, consider a biometric system that should verify the identity of users by analyzing their fingerprints. In this case an impostor may provide a “fake” fingerprint of a genuine user to the sensor, to

Author for correspondence:

[1]Final year ME CSE, Mahendra Institute of Technology, Mahendhrapuri, Tamilnadu, India.

[2]Assistant Professor/CSE Mahendra Institute of Technology, Mahendhrapuri, Tamilnadu, India.

gain access to the system as that genuine user. This is typically referred to as spoof attack in biometrics [3].

The presence of malicious adversaries causes a specific kind of non-stationary [4] and raises several open issues with respect to state-of-the-art design methods of pattern recognition systems. To date, a systematic and unifying treatment of adversarial classification problems is still lacking and this is attracting a growing interest from the pattern recognition and machine learning communities, how witnessed by a workshop held in the context of the NIPS 2007 conference and by a subsequent special issue of the Machine Learning Journal [5]. So far, works in the adversarial classification field have been focused on three main open problems:

1) identifying and categorising vulnerabilities of machine learning algorithms, which potentially expose them to adversarial attacks;

2) evaluating their performance under attack;

3) developing defense strategies to counteract attacks and secure classifiers.

Only a few attempts to develop general frameworks have been made, related to the first two issues above. Most of the works related to the third issue focused instead on specific applications, classifiers and attack scenarios.

RELATED WORK

A taxonomy of potential attacks against pattern classifiers was proposed in [8] and subsequently extended. We will exploit it in our framework, as part of the definition of attack scenarios. The taxonomy is based on two main features: the kind of influence of attacks on the classifier and the kind of security violation they cause. The influence can be either causative, if it undermines the learning algorithm to cause subsequent misclassifications; or exploratory, if it exploits knowledge of the trained classifier to cause misclassifications, without affecting the learning algorithm. Thus, causative attacks may influence both training and testing data and only training data, whereas exploratory attacks affect only testing data. The security violation can be an integrity violation, if it allows the adversary to access the service or resource protected by the classifier; an availability violation, if it denies legitimate users access to it; or a privacy violation, if it allows the adversary to obtain confidential information from the classifier. Integrity violations result in misclassifying malicious samples as legitimate, while availability violations can also cause legitimate samples to be misclassified as malicious. A third feature of the taxonomy is the specificity of an attack, that ranges from targeted to indiscriminate, depending on whether the attack focuses on a single or few specific samples, or on a wider set of samples.

Many authors implicitly performed security evaluation as a what-if analysis, based on empirical simulation methods; they mainly focused on a specific application, classifier and attack and devised ad hoc security evaluation procedures based on the exploitation of problem knowledge and heuristic techniques [12] [6]. Their goal was either to point out a previously unknown vulnerability, or to evaluate security against a known attack. In some cases, specific countermeasures were also proposed, according to a proactive/security-by-design approach. Attacks were simulated by manipulating train in g and testing samples according to application-specific criteria only, without reference to more general guidelines; consequently, such techniques cannot be directly exploited by a system designer in more general cases.

A few works proposed analytical methods to evaluate the security of learning algorithms or of some classes of decision functions, based on more general, application-independent criteria to model the adversary's behavior [10]. Some of these criteria will be exploited in our framework for empirical security evaluation; in the definition of the adversary model, as high-level guidelines for simulating attacks. We summarize here the three main concepts more or less explicitly emerged from previous work that will be exploited in our framework for security evaluation.

1) Arms race and security by design: since it is not possible to predict how many and which kinds of attacks a classifier will incur during operation, classifier security should be proactively evaluated

using a what-if analysis, by simulating potential attack scenarios.

2) Adversary modeling: effective simulation of attack scenarios requires a formal model of the adversary.

3) Data distribution under attack: the distribution of testing data may differ from that of training data, when the classifier is under attack.

PATTERN CLASSIFIER SECURITY

Pattern classification systems based on machine learning algorithms are commonly used in security-related applications like biometric authentication, network intrusion detection and spam filtering, to discriminate between a "legitimate" and a "malicious" pattern class. Contrary to traditional ones, these applications have an intrinsic adversarial nature since the input data can be purposely manipulated by an intelligent and adaptive adversary to undermine classifier operation. This often gives rise to an arms race between the adversary and the classifier designer. Well known examples of attacks against pattern classifiers are: submitting a fake biometric trait to a biometric authentication system, modifying network packets belonging to intrusive traffic to evade intrusion detection systems (IDSs); manipulating the content of spam emails to get them past spam filters. Adversarial scenarios can also occur in intelligent data analysis and information retrieval.

It is now acknowledged that, since pattern classification systems based on classical theory and design methods do not take into account adversarial settings, they exhibit vulnerabilities to several potential attacks, allowing adversaries to undermine their effectiveness [7]. A systematic and unified treatment of this issue is thus needed to allow the trusted adoption of pattern classifiers in adversarial environments, starting from the theoretical foundations up to novel design methods, extending the classical design cycle. In particular, three main open issues can be identified: (i) analyzing the vulnerabilities of classification algorithms and the corresponding attacks; (ii) developing novel methods to assess classifier security against these attacks, which is not possible using classical performance evaluation methods (iii) developing novel design methods to guarantee classifier security in adversarial environments. Although this emerging field is attracting growing interest [9], the above issues have only been sparsely addressed under different perspectives and to a limited extent. Most of the work has focused on application-specific issues related to spam filtering and network intrusion detection while only a few theoretical models of adversarial classification problems have been proposed in the machine learning literature; they do not yet provide practical guidelines and tools for designers of pattern recognition systems. Besides introducing these issues to the pattern recognition research community, in this work we address issues (i) and (ii) above by developing a framework for the empirical evaluation of classifier security at design phase that extends the model selection and performance evaluation steps of the classical design cycle.

We summarize previous work and point out three main ideas that emerge from it. We then formalize and generalize them in our framework. First, to pursue security in the context of an arms race it is not sufficient to react to observed attacks, but it is also necessary to proactively anticipate the adversary by predicting the most relevant, potential attacks through a what-if analysis; this allows one to develop suitable countermeasures before the attack actually occurs, according to the principle of security by design. Second, to provide practical guidelines for simulating realistic attack scenarios, we define a general model of the adversary, in terms of her goal, knowledge and capability, which encompass and generalizes models proposed in previous work. Third, since the presence of carefully targeted attacks may affect the distribution of training and testing data separately, we propose a model of the data distribution that can formally characterize this behavior and that allows us to take into account a large number of potential attacks. We give three concrete examples of applications of our framework in spam filtering, biometric authentication and network intrusion detection. We discuss how the classical design cycle of pattern classifiers should be revised to take security into account. Finally, we

summarize our contributions, the limitations of our framework and some open issues.

SECURITY SCHEME FOR DATA CLASSIFICATION

ADVERSARY MODELS

Although the definition of attack scenarios is ultimately an application-specific issue, it is possible to give general guidelines that can help the designer of a pattern recognition system. Here we propose to specify the attack scenario in terms of a conceptual model of the adversary that encompasses, unifies and extends different ideas from previous work. Our model is based on the assumption that the adversary acts rationally to attain a given goal, according to her knowledge of the classifier and her capability of manipulating data. This allows one to derive the corresponding optimal attack strategy.

ADVERSARY'S GOAL

It is formulated as the optimization of an objective function. We propose to define this function based on the desired security violation and on the attack specificity, according to the taxonomy [11]. For instance, the goal of an indiscriminate integrity violation may be to maximize the fraction of misclassified malicious samples; the goal of a targeted privacy violation may be to obtain some specific, confidential information from the classifier by exploiting the class labels assigned to some "query" samples, while minimizing the number of query samples that the adversary has to issue to violate privacy.

ADVERSARY'S KNOWLEDGE

Assumptions on the adversary's knowledge have only been qualitatively discussed in previous work, mainly depending on the application at hand. Here we propose a more systematic scheme for their definition, with respect to the knowledge of the single components of a pattern classifier: (k.i) the training data; (k.ii) the feature set; (k.iii) the learning algorithm and the kind of decision function; (k.iv) the classifier's decision function and its parameters; (k.v) the feedback available from the classifier, if any. It is worth noting that realistic and minimal assumptions about what can be kept fully secret from the adversary should be done.

ADVERSARY'S CAPABILITY

It refers to the control that the adversary has on training and testing data. We propose to define it in terms of: (c.i) the attack influence; (c.ii) whether and to what extent the attack affects the class priors; (c.iii) how many and which training and testing samples can be controlled by the adversary in each class; (c.iv) which features can be manipulated and to what extent, taking into account application-specific constraints.

ATTACK STRATEGY

One can finally define the optimal attack strategy, namely, how training and testing data should be quantitatively modified to optimize the objective function characterizing the adversary's goal. Such modifications are defined in terms of: (a.i) how the class priors are modified; (a.ii) what fraction of samples of each class is affected by the attack; and (a.iii) how features are manipulated by the attack. Once the attack scenario is defined in terms of the adversary model and the resulting attack strategy, our framework proceeds with the definition of the corresponding data distribution is used to construct training and testing sets for security evaluation.

TRAINING AND TESTING SET

Here we propose an algorithm to sample training (TR) and testing (TS) sets of any desired size from the distributions ptr(X|Y) and pts(X|Y). We assume that $k \geq 1$ different pairs of training and testing sets (D^{i}_{TR}, ,D^{i}_{TS}), i = (1, . . . , k) have been obtained from D using a classical resampling technique, like cross-validation or bootstrapping. Accordingly, their samples follow the distribution p (X|Y). We describe how to modify each of the sets D^{i}_{TR} to construct a training set Tri that follows the distribution ptr(X,Y). For the sake of simplicity, we will omit the superscript i. An identical procedure can be followed to construct a testing set T_{Si} from each of the D^{i}_{TS} .

Security evaluation is then carried out with the classical method, by averaging (if k > 1) the performance of the classifier trained on TRi and tested on TSi. If the attack does not affect the training samples, i.e., ptr(X, Y) = p (X|Y), TR is simply set equal to DTR. Otherwise, two alternatives are possible. (i) If ptr(X|Y, A) is analytically defined for each Y ∈ {L,M} and A{T,F}, then TR can be obtained by sampling the generative model of p(X,Y,A) of Eq. (3): first, a class label y is sampled from ptr(Y), then, a value a from ptr(A|Y = y) and finally, a feature vector x from ptr(X|Y = y,A = a). (ii) If ptr(X|Y = y,A = a) is not analytically defined for some y and a, but a set of its samples is available, denoted in the following as $D_{TR}^{y,a}$, it can be approximated as the empirical distribution of $D_{TR}^{y,a}$.

Accordingly, we can sample with replacement from $D_{TR}^{y,a}$. An identical procedure can be used to construct the testing set TS.

CATEGORIZATION PROCESS

We summarize here the steps that the designer of a pattern classifier should take to evaluate its security using our framework, for each attack scenario of interest. They extend the performance evaluation step of the classical design cycle, which is used as part of the model selection phase and to evaluate the final classifier to be deployed.

1) Attack scenario. The attack scenario should be defined at the conceptual level by making specific assumptions on the goal, knowledge (k.i-v) and capability of the adversary (c.i-iv) and defining the corresponding attack strategy (a.iii)

2) Data model. According to the hypothesized attack scenario, the designer should define the distributions p(Y), p(A|Y) and p(X|Y ,A), for Y ∈ {L,M}, A ∈ {F,T} and for training and testing data. If p(X|Y ,A) is not analytically defined for some Y = y and A = a, either for training or testing data, the corresponding set $D_{TR}^{y,F}\left(D_{TS}^{y,F}\right)$ must be constructed. The sets $D_{TR}^{y,F}\left(D_{TS}^{y,F}\right)$ are obtained from D_{TR} (D_{TS}). The sets $D_{TR}^{y,T} and D_{TS}^{y,T}$ can be generated, only if the attack involves sample manipulation, using an attack sample simulation technique according to the attack strategy (a.iii).

3) Construction of TR and TS. Given k ≥ 1 pairs $\left(D_{TR}^{i}, D_{TS}^{i}\right), i = 1,..., k$ obtained from classical resampling techniques like cross-validation or bootstrapping, the size of TR and TS must be run with the corresponding inputs to obtain TR^{i} and TS^{i}. If the attack does not affect the training (testing) data, TR^{i} (TS) is set to D_{TR}^{i} (D_{TS}^{i}).

4) Performance evaluation. The classifier performance under the simulated attack is evaluated using the constructed (TR^{i},TS^{i}) pairs, as in classical techniques.

PROBLEM DEFINITION

The security violation is divided into three types such as integrity violation, availability violation and privacy violation. An integrity violation, if it allows the adversary to access the service or resource protected by the classifier. An availability violation, if it denies legitimate users access to it. A privacy violation, if it allows the adversary to obtain confidential information from the classifier. Pattern classification techniques are extended with adversarial settings. Secure pattern classifiers are designed to control the performance degradation under potential attacks. Secure pattern classifier framework is build with model selection and training and testing data construction method. TR and TS construction algorithm is used to select data for pattern classifier. The following issues are identified from the pattern classifier security schemes.

- Misclassification rate is high
- Inaccurate pattern generation in training cycles
- Limited classifier security
- Detection latency is high

CONTROLLING PATTERN ATTACKS ON CLASSIFICATION

The proposed system is designed to enhance the secure pattern classifier with attack control mechanism. Training patterns are secured with simulated attack patterns. Pattern update process is monitored and controlled in testing process. Classifier utility rate is improved in testing process. The system is designed to classify the e-mails by analyzing the mail contents. Span and normal keywords are identified in the learning process. Learned patterns are protected with simulated patterns. The system is divided into five major modules. They are data preprocess, training process, pattern security, classification process and pattern update process.

The data preprocess module is designed to perform noise removal and feature selection process. Class patterns are identified in the learning process. Pattern security module is designed to protect learned patterns. Classification module is designed to categorize the e-mails. New patterns are updated in the pattern update module. The data preprocess is applied to parse email contents. Training set and testing set are separated with user data values. Noise elimination is performed in the email contents. Feature selection is carried out on the cleaned data values. Pattern learning is performed in the training process. The mail contents are selected with label information. Legitimate and spam mail contents are analyzed in the training process. Spam keywords and legitimate keywords are identified from the training process.

The pattern security process is initiated to protect the learned patterns. Security violation is controlled in the pattern security process. Simulated patterns are used to protect the patterns from the attackers. Confidential data access is protected from adversarial activities. The classification process is used to assign labels for emails. Testing set is used in the classification process. The mails are classified with learned patterns. Keywords are compared to identify the category of the emails. The testing process updates the learned patterns in the classification process. Additional keywords are added to the learned patterns. Pattern monitoring process is used to control the attacker activities. Updated patterns are used in the classification process.

The pattern classifier attack management mechanism is tested for the spam filtering process. Assume that a classifier has to discriminate between legitimate and spam emails on the basis of their textual content and that the bag-of-words feature representation has been chosen, with binary features denoting the occurrence of a given set of words. This kind of classifier has been considered by several authors and it is included in several real spam filters. The pattern classifier system is tested with two methods. They are Pattern Classifier with Security (PCS) and Enhanced Pattern Classifier with Security (EPCS) schemes. The system is tested with different e-mail counts.

CONCLUSION

Data classifiers are used to perform training and testing process is data categorization process. Pattern classifiers are used for anomaly based data classification process. Secure pattern classifiers are constructed to control attacks on classification process. Simulated patterns based attack controlling mechanism is used to improve the secure pattern classifier scheme. False positive and false negative errors are reduced in the classification process. Classification accuracy is improved. Computational complexity is reduced in the training and testing process. The system controls the pattern based attacks against classifiers.

REFERENCES

[1]. N. Dalvi, Mausam, Sanghai and D. Verma, “Adversarial classification,” in 10th ACM SIGKDD Int.’l Conf. Knowledge Discovery and Data Mining (KDD), Seattle, 2009.

[2]. A. Kolcz and C. H. Teo, “Feature weighting for improved classifier robustness,” in 6th Conf. Email and Anti-Spam (CEAS), CA, USA, 2009.

[3]. R. N. Rodrigues, Ling and V. Govindaraju, “Robustness of multimodal biometric fusion methods against spoof attacks,” Comput., 2009.

[4]. M. Barreno, B. Nelson, A. Joseph and J. Tygar, "The security of machine learning," Machine Learning, vol. 81, pp. 121–148, 2010.

[5]. P. Laskov and R. Lippmann, "Machine learning in adversarial environments," Machine Learning, vol. 81, pp. 115–119, 2010.

[6]. B. Biggio, Nelson and P. Laskov, "Poisoning Attacks against Support Vector Machines," Proc. 29th Int'l Conf. Machine Learning, 2012.

[7]. Huang and D. Tygar, "Adversarial Machine Learning," Proc. Fourth ACM Workshop Artificial Intelligence and Security, 2011.

[8]. M. Barreno, B. Nelson, A. Joseph and J. Tygar, "The Security of Machine Learning," Machine Learning, vol. 81, pp. 121-148, 2010.

[9]. Dagstuhl Perspectives Workshop Mach. Learning Methods for Computer Sec., http://www.dagstuhl.de/12371/, 2012.

[10]. M. Bruckner, C. Kanzow and T. Scheffer, "Static Prediction Games for Adversarial Learning Problems," J. Machine Learning Research, vol. 13, pp. 2617-2654, 2012.

[11]. Noman Mohammed and Mourad Debbabi, "Secure Two-Party Differentially Private Data Release for Vertically Partitioned Data", IEEE Transactions On Dependable And Secure Computing, January/February 2014

[12]. B. Biggio Roli and Didaci, "Poisoning Adaptive Biometric Systems," Proc. Joint IAPR Int'l Conf. Structural, Syntactic and Statistical Pattern Recognition, 2012.