International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 2, Issue 7,July 2015

Security Based Pattern Classifiers

[1]Mr. Zaid Alam Khan,[2]Mr. MD Azher,[3]Mr. Kante Surya Chandra Rao,[4]Ms. Neelu l
[1][2][3]UG students, Department of Computer Science & Engineering, RajaRajeswari College of Engineering, Bangalore-74, India
[4]Asst. Professor, Department of Computer Science & Engineering, RajaRajeswari College of Engineering, Bangalore-74, India
[1]zaid622@gmail.com,[2]mdazher.shaik786@gmail.com,[3]suryakante888@gmail.com,[4]neelu.lalband@gmail.com

Abstract- **Security is usually defined as opposing oneself from harmful attacks. Security is a part of everyone's life. People wants to be safe and secure all the time but one never knows when his/her system can be attacked by malicious intruders. However upgrading ones security at the highest level possible is a necessary task. Pattern classification systems are commonly used in adversarial applications, such as biometric authentication, network intrusion detection, and spam filtering. It is to be noted that in these three areas data can be purposely manipulated or modified by humans to undermine their operation. These scenarios are not considered by classical design methods. Pattern classification systems may exhibit vulnerabilities, and when exploited may severely affect performance. Extension of pattern classification theory and design methods to real time applications is thus a very relevant research direction which has not yet been pursued in a systematic way and proper way. This paper introduces one of the main open issues: establishing a security system as a real time application which can be used in several organisations such as hospitals, banking system, libraries etc. Reports shows that security evaluation can provide a more complete understanding of the classifier's behaviour and lead to better design choices.**

Keywords: **Pattern classification, adversarial classification, security evaluation, robustness evaluation**

I. INTRODUCTION

Pattern classification systems that are based on machine learning algorithms are widely used in security-related applications such as biometric authentication, network intrusion detection, and spam filtering, to differentiate between a "legal/legitimate" and a "malicious" pattern class ex: legitimate and spam mails. Contrary to existing approach, 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. Well known examples of attacks against pattern classifiers are: submission of a fake biometric trait to a biometric authentication system. This is popularly known as spoofing attacks; modifying or altering the network packets that belongs to intrusive traffic to escape intrusion detection systems (IDSs); modifying the file contents of the spam emails to getthem past the spam filters. This is achieved by misspelling common words that belongs under the category of spam to avoid their detection. A malicious web user may manipulate with the search engine ratings so as to artificially promote their webpages/websites.

Furthermore it is observed that the existing systems based on classical approach and design techniques exhibit vulnerabilities to different attacks posed by the intruders, which in turn affects the performance of the system resulting in the degrading the functionalities of the pattern classification systems. Thus, the system becomes less effective and more prone to attacks. A more systematic approach is needed to make the existing system more effective and trust worthy thereby providing a higher level of security, preventing the system from malicious attacks. Hence the user data is secured and safe in the system. There are two main open issues that can be identified: (i) analysing the existing classical algorithms of the previous work, and the related attacks on it; (ii) developing new methods to enhance the classifier security against these attacks, which is not possible using classical performance evaluation methods.

Besides introducing these concepts to the research community, the issues that are addressed are (i) and (ii) above by implementing the concepts in the real time applications.

II. RELATED WORK

This section deals with the background and previous work which lead to the making of the current system.

The attack against pattern classification systems was proposed in the paper, and further extended in paper. The classification is based on two key features: the kind of influence of attacks on the system, and the kind of security violation they cause in the system. The causative attacks can influence the training data as well as the testing data

respectively. The security violation can lead to integrity violation if it is able to access the resources protected by the user and a privacy violation takes place when it allows the adversary to access the resources or files that are confidential for the existing user. Integrity violations result in misclassified malicious samples as legitimate, while availability violations can also cause legitimate samples to be misclassified as malicious, however one feature of the taxonomy of the classification system is the specificity of an attack that ranges from targeted to indiscriminate, depending on whether the attack focuses on a single sample or few specific samples (e.g., a specific spam email misclassified as legitimate), or on a wider set of samples.

System designer should predict the adversary classified system by simulating a "proactive" arms race to (i) identify the most relevant threats and attacks on the system, and (ii) propose a proper countermeasures, before modifying the existing classification systems. Furthermore, this improves security as it requires the adversary to spend a greater effort i.e. to spend more time, put more skills and resources to find, modify and exploit vulnerabilities. Hence system security is guaranteed for a muchlonger time, with less frequent supervision or human intervention on the system.

The goal of security evaluation of the classification system is to address issue (i) above, i.e., to simulate a number of realistic attack scenarios that may be incurred during operation, and to assess the impact of the corresponding attacks on the targeted classifier to highlight the most critical vulnerabilities. Although security evaluation of the pattern classification system may also suggest specific countermeasures i.e. the design of secure classifiers.

Many authors implicitly performed security evaluation as a what-if analysis, based on empirical simulation methods; however, 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. 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 training 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.

BUILDING ON THE PREVIOUS WORK

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 modelling: 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.

Our main goal is to provide a quantitative and general-purpose basis for the application of the what-if analysis to classifier security evaluation, based on the definition of potential attack scenarios. To this end, we propose: (i) a model of the adversary, that allows us to define any attack scenario;(ii) a corresponding model of the data distribution; and (iii) a method for generating training and testing sets that are representative of the data distribution, and are used for empirical evaluation.

III. APPLICATION EXAMPLES

While previous work focused on a single application, we consider here three different applicationexamples: spam filtering, biometric authentication, and network intrusion detection. Our aim is to show how the designer of a pattern classifier can use our framework, and what kind of additional information he can obtain from security evaluation. We will show that a trade-off between classifier accuracy and security emerges sometimes, and that this information can be exploited for several purposes; e.g., to improve the model selection phase by considering both classification accuracy and security.

3.1 Spam Filtering

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.7

Attack scenario. Goal. The adversary aims at maximizing the percentage of spam emails misclassified as legitimate, which is an indiscriminate integrity violation.

The adversary in (i) is assumed to have perfect knowledge of the classifier, i.e.,: (ii) the feature set, (iii) the kind of decision function, and (iv) its parameters (the weight assigned to each feature, and the decision threshold). Assumptions on the knowledge of (v) the training data and (vi) feedback from the classifier are not relevant in this case, as they do not provide any additional information.

3.2 Biometric Authentication

Multimodal biometric systems for personal identity recognition have received great interest in the past few years. It has been shown that combining information coming from different biometric traits can overcome the limits and the weaknesses inherent in every individual biometric, resulting in a higher accuracy. Moreover, it is commonly believed that multimodal systems also improve security against spoofing attacks, which consist of claiming a false

identity and submitting at least one fake biometric trait to the system (e.g., a "gummy" fingerprint or a photograph of a user's face). The reason is that, to evade a multimodal system, one expects that the adversary should spoof all the corresponding biometric traits. In this application example, we show how the designer of a multimodal system can verify if this hypothesis holds, before deploying the system, by simulating spoofing attacks against each of the matchers. To this end, we partially exploit the analysis in.

The design phase includes the enrolment of authorized users (clients): reference templates of their biometric traits are stored into a database, together with the corresponding identities. During operation, each user provides the requested biometric traits to the sensors, and claims the identity of a client. Then, each matcher compares the submitted trait with the template of the claimed identity, and provides a real-valued matching score: the higher the score, the higher the similarity. We denote the score of the fingerprint and the face matcher respectively as xfing and xface. Finally, the matching scores are combined through a proper fusion rule to decidewhether the claimed identity is the user's identity (genuine user) or not (impostor).

1) Attack scenario. Goal. In this case, each malicious user (impostor) aims at being accepted as a legitimate (genuine) one. This corresponds to a targeted integrity violation, where the adversary's goal is to maximize the matching score.

Knowledge. As in [1], we assume that each impostor knows: (i) the identity of the targeted client; and (ii) the biometric traits used by the system. No knowledge of (iii) the decision function and (iv) its parameters is assumed, and (v) no feedback is available from the classifier.

3.3 Network Intrusion Detection

Intrusion detection systems analyze network traffic to prevent and detect malicious activities like intrusion attempts, port scans, and denial-of-service attacks.11 When suspected malicious traffic is detected, an alarm is raised by the IDS and subsequently handled by the system administrator. Two main kinds of IDSs exist: misuse detectors and anomaly-based ones. Misuse detectors match the analyzed network traffic against a database of signatures of known malicious activities (e.g., Snort).12 The main drawback is that they are not able to detect never-before-seen malicious activities, or even variants of known ones. To overcome this issue, anomaly-based detectors have been proposed. They build a statistical model of the normal traffic using machine learning techniques, usually one-class classifiers (e.g., PAYL), and raise an alarm when anomalous traffic is detected. Their training set is constructed, and periodically updated to follow the changes of normal traffic, by collecting unsupervised network traffic during operation, assuming that it is normal (it can be filtered by a misuse detector, and should be discarded if some system malfunctioning occurs during its collection). This kind of IDS is vulnerable to causative attacks, since an attacker may inject carefully designed malicious traffic during the collection of training samples to force the IDS to learn a wrong model of the normal traffic.

Attack scenario. Goal. This attack aims to cause an indiscriminate integrity violation by maximizing the fraction of malicious testing samples misclassified as legitimate.

Knowledge. The adversary is assumed to know: (ii) the feature set; and (iii) that a one-class classifier is used. No knowledge of (i) the training data and (iv) the classifiers' parameters is available to the adversary, as well as (v) any feedback from the classifier.

Capability. The attack consists of injecting malicious samples into the training set. Accordingly, we assume that: (i) the adversary can inject malicious samples into the training data, without manipulating testing data (causative attack); (ii) she can modify the class priors by injecting a maximum fraction pmax of malicious samples into the training data; (iii) all the injected malicious samples can be manipulated; and (iv) the adversary is able to completely control the feature values of the malicious attack samples. Repeat the security evaluation for pmax 2 ½0;0:5, since it is unrealistic that the adversary can control the majority of the training data.

IV. SYSTEM DESIGN

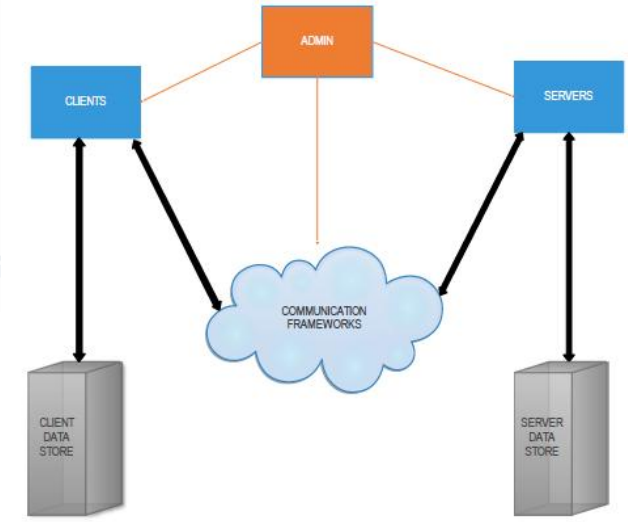

Figure 1: System Design

The above figure shows the system design of the system based classifiers where the interaction between each components are viewed.

There are six main components involved namely clients, client data store, servers, server data store, admin and communication frameworks.

i. The client's module involves the list of users which are registered into the system. The clients consists of the properties such as name, gender, email id, phone no., city, etc.

ii. Client's data store consists of the data set of each clients. The client's data store is used to retrieve and fetch information as and when required by the user or client.

iii. Servers are capable of accepting requests from the client and then

responding to the request made by the clients.

iv. Server data store is used to store the databases of the client's so as to provide a backup of the files and resources of the client as well as to provide online storage facility to the files.

v. Admin takes notice of both the client and server. The admin is able to see the user details where the admin can delete the user if the user does not exist in the system or when the user is involved in some illegal activities. The admin keeps the updated table of the biometric reports of the user i.e. the log in and log out status of the client and also whether the log in was a fail or success.

vi. The communication frameworks consists of the webpages and programming languages involved for the effective communication between the client and server.

V. SYSTEM IMPLEMENTATION AND MODULE DESCRIPTION

1. Attack scenario and model of the adversary
2. A model of the data distribution
3. Training and testing set generation
4. Performance evaluation

ATTACK SCENARIO AND MODEL OF THE ADVERSARY

Although the definition of attack scenarios is ultimately an application-specific issue, it is possible CLIENTSSERVERSCLIENTDATASTORECOMMUNICATION FRAMEWORKSSERVERDATASTOREADMIN 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 (integrity, availability, or privacy), and on the attack specificity (from targeted to indiscriminate), according to the taxonomy. 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 (e.g., the biometric trait of a given user enrolled in a biometric system) 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: (i) the training data; (ii) the feature set; (iii) the learning algorithm and the kind of decision function (e.g., a linear SVM); (iv) the classifier's decision function and its parameters (e.g., the feature weights of a linear classifier); (v) the feedback available from the classifier, if any (e.g., the class labels assigned to some "query" samples that the adversary issues to get feedback). 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: (i) the attack influence (either causative or exploratory), as defined; (ii) whether and to what extent the attack affects the class priors; (iii) how many and which training and testing samples can be controlled by the adversary in each class; (iv) which features can be manipulated, and to what extent, taking into account application-specific constraints (e.g., correlated features cannot be modified independently, and the functionality of malicious samples cannot be compromised).

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: (i) how the class priors are modified; (ii) what fraction of samples of each class is affected by the attack; and (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 that is used to construct training and testing sets for security evaluation.

VI. CONTRIBUTIONS, LIMITATIONS AND OPEN ISSUES

In this paper we focused on empirical security evaluation of pattern classifiers that have to be deployed in adversarial environments, and proposed how to revise the classical performance evaluation design step, which is not suitable for this purpose.

Our main contribution is a framework for empirical security evaluation that formalizes and generalizes ideas from previous work, and can be applied to different classifiers, learning algorithms, and classification tasks. It is grounded on a formal model of the adversary, and on a model of data

distribution that can represent all the attacks considered in previous work; provides a systematic method for the generation of training and testing sets that enables security evaluation; and can accommodate application-specific techniques for attack simulation. This is a clear advancement with respect to previous work, since without a general framework most of the proposed techniques (often tailored to a given classifier model, attack, and application) could not be directly applied to other problems.

An intrinsic limitation of our work is that security evaluation is carried out empirically, and it is thus data dependent; on the other hand, model-driven analyses require a full analytical model of the problem and of the adversary's behaviour that may be very difficult to develop for real-world applications. Another intrinsic limitation is due to fact that our method is not application-specific, and, therefore, provides only high-level guidelines for simulating attacks. Indeed, detailed guidelines require one to take into account application-specific constraints and adversary models. Our future work will be devoted to develop techniques for simulating attacks for different applications.

Although the design of secure classifiers is a distinct problem than security evaluation, our framework could be also exploited to this end. For instance, simulated attack samples can be included into the training data to improve security of discriminative classifiers (e.g., SVMs), while the proposed data model can be exploited to design more secure generative classifiers. We obtained encouraging preliminary results on this topic.

VII. CONCLUSION AND FUTURE ENHANCEMENT

In this paper we focused on empirical security evaluation of pattern classifiers that have to be deployed in adversarial environments, and proposed how to revise the classical performance evaluation design step, which is not suitable for this purpose. Our main contribution is a framework for empirical security evaluation that formalizes and generalizes ideas from previous work, and can be applied to different classifiers, learning algorithms, and classification tasks. It is grounded on a formal model of the adversary, and on a model of data distribution that can represent all the attacks considered in previous work; provides a systematicmethod for the generation of training and testing sets that enables security evaluation; and can accommodate application-specific techniques for attack simulation. This is a clear advancement with respect to previous work, since without a general framework most of the proposed techniques (often tailored to a given classifier model, attack, and application) could not be directly applied to other problems.

An intrinsic limitation of our work is that security evaluation is carried out empirically, and it is thus data dependent; on the other hand, model-driven analyses require a full analytical model of the problem and of the adversary's behavior that may be very difficult to develop for real-world applications. Another intrinsic limitation is due to fact that our method is not application-specific, and, therefore, provides only high-level guidelines for simulating attacks. Indeed, detailed guidelines require one to take into account application specific constraints and adversary models. Our future work will be devoted to develop techniques for simulating attacks for different applications.

Although the design of secure classifiers is a distinct problem than security evaluation, our framework could be also exploited to this end. For instance, simulated attack samples can be included into the training data to improve security of discriminative classifiers, while the proposed data model can be exploited to design more secure generative classifiers. We obtained encouraging preliminary results on this topic.

ACKNOWLEDGMENTS

The authors are grateful to Davide Ariu, Gavin Brown, Pavel Laskov, and Blaine Nelson for discussions and comments on an earlier version of this paper. This work was partly supported by a grant awarded to Battista Biggio by Regione Autonoma della Sardegna, PO Sardegna FSE 20072013, L.R. 7/2007 "Promotion of the scientific research and technological innovation in Sardinia", by the project CRP18293 funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009, and by the TABULA RASA project, funded within the seventh Framework Research Programme of the European Union.

REFERENCES

[1] R.N. Rodrigues, L.L. Ling, and V. Govindaraju, "Robustness of Multimodal Biometric Fusion Methods against Spoof Attacks," J. Visual Languages and Computing, vol. 20, no. 3, pp. 169-179, 2009.

[2] P. Johnson, B. Tan, and S. Schuckers, "Multimodal Fusion Vulnerability to Non-Zero Effort (Spoof) Imposters," Proc. IEEE Int'l Workshop Information Forensics and Security, pp. 1-5, 2010.

[3] P. Fogla, M. Sharif, R. Perdisci, O. Kolesnikov, and W. Lee, "Polymorphic Blending Attacks," Proc. 15th Conf. USENIX Security Symp., 2006.

[4] G.L. Wittel and S.F. Wu, "On Attacking Statistical Spam Filters," Proc. First Conf. Email and Anti-Spam, 2004.

[5] D. Lowd and C. Meek, "Good Word Attacks on Statistical Spam
Filters," Proc. Second Conf. Email and Anti-Spam, 2005.

[6] A. Kolcz and C.H. Teo, "Feature Weighting for Improved Classifier Robustness," Proc. Sixth Conf. Email and Anti-Spam, 2009.

[7] D.B. Skillicorn, "Adversarial Knowledge Discovery," IEEE Intelligent Systems, vol. 24, no. 6, Nov./Dec. 2009.

[8] D. Fetterly, "Adversarial Information Retrieval: The Manipulation of Web Content," ACM Computing Rev., 2007.

[9] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification. Wiley-Interscience Publication, 2000.
[10] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial Classification," Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 99-108, 2004.
[11] M. Barreno, B. Nelson, R. Sears, A.D. Joseph, and J.D. Tygar, "Can Machine Learning be Secure?" Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 16-25, 2006.
[12] A.A. Cardenas and J.S. Baras, "Evaluation of Classifiers: Practical Considerations for Security Applications," Proc. AAAI Workshop Evaluation Methods for Machine Learning, 2006.

[13] P. Laskov and R. Lippmann, "Machine Learning in Adversarial Environments," Machine Learning, vol. 81, pp. 115-119, 2010.
[14] L. Huang, A.D. Joseph, B. Nelson, B. Rubinstein, and J.D. Tygar, "Adversarial Machine Learning," Proc. Fourth ACM Workshop
Artificial Intelligence and Security, pp. 43-57, 2011.
[15] M. Barreno, B. Nelson, A. Joseph, and J. Tygar, "The Security of Machine Learning," Machine Learning, vol. 81, pp. 121-148, 2010.
[16] D. Lowd and C. Meek, "Adversarial Learning," Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 641647, 2005.
[17] P. Laskov and M. Kloft, "A Framework for Quantitative Security Analysis of Machine Learning," Proc. Second ACM Workshop Security and Artificial Intelligence, pp. 1-4, 2009.
[18] NIPS Workshop Machine Learning in Adversarial Environments for Computer Security, http://mls-nips07.first.fraunhofer.de/, 2007.
[19] Dagstuhl Perspectives Workshop Mach. Learning Methods for Computer Sec., http://www.dagstuhl.de/12371/, 2012.